ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Τροπική Γεωμετρία και Βελτιστοποίηση με Εφαρμογές στη Μηχανική Μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασίλειος Χαρισόπουλος

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής ΕΜΠ

Αθήνα, 1 Απριλίου 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Τροπική Γεωμετρία και Βελτιστοποίηση με Εφαρμογές στη Μηχανική Μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασίλειος Χαρισόπουλος

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Απριλίου 2017

.....	.....	.....
Πέτρος Μαραγκός Καθηγητής ΕΜΠ	Ανδρέας - Γεώργιος Σταφυλοπάτης Καθηγητής ΕΜΠ	Κωνσταντίνος Τζαφέστας Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, 1 Απριλίου 2017

.....
Βασίλειος Χαρισόπουλος
(Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών Ε.Μ.Π.)

Οι απόψεις που εκφράζονται σε αυτό το κείμενο είναι αποκλειστικά του συγγραφέα και δεν αντιπροσωπεύουν απαραίτητα την επίσημη θέση του Εθνικού Μετσόβιου Πολυτεχνείου.

Το περιεχόμενο της εργασίας διατίθεται υπό την άδεια Creative Commons Attribution-NonCommercial 4.0. Απαγορεύεται η χρήση του περιεχομένου για εμπορικούς σκοπούς.

Περίληψη

Την τελευταία δεκαετία, το επιστημονικό πεδίο της Τεχνητής Νοημοσύνης έχει σημαδευθεί από την ολοένα και εντονότερη χρήση του υπολογιστικού μοντέλου των Νευρωνικών Δικτύων. Τα Νευρωνικά Δίκτυα παραδοσιακά στηρίζονταν σε κυρίως γραμμικά μοντέλα, όπως η πολλαπλασιαστική-αθροιστική αρχιτεκτονική του γραμμικού Perceptron η οποία παραμένει το κυρίαρχο παράδειγμα νευρωνικών υπολογισμών. Ωστόσο, από την οπτική γωνία της Βιολογίας, η δραστηριότητα ενός νευρώνα θα μπορούσε εξίσου να στηρίζεται σε εγγενώς μη-γραμμικές και ανταγωνιστικές διαδικασίες και πράξεις. Η Μαθηματική Μορφολογία και η Minimax Άλγεβρα παρέχουν το κατάλληλο υπόβαθρο για τη μελέτη των Νευρωνικών Δικτύων που αποτελούνται από τέτοιες μη γραμμικές μονάδες.

Η αποτελεσματική χρήση των Νευρωνικών Δικτύων σχεδόν πάντα προϋποθέτει την επίλυση "δύσκολων" προβλημάτων βελτιστοποίησης. Για το σκοπό αυτό, στην παρούσα εργασία αρχικά εξετάζουμε μια πρόσφατα προτεινόμενη μέθοδο εξομάλυνσης μη-κυρτών προβλημάτων στην περίπτωση που οι υπολογισμοί που περιλαμβάνει δεν είναι δυνατόν να υπολογιστούν αναλυτικά. Δείχνουμε ότι μπορούμε να προσεγγίσουμε τη μέθοδο αυτή με τη χρήση δειγματοληπτικών μεθόδων και αποδεικνύουμε κάποιες σχετικές ιδιότητες, μεταξύ των οποίων ένα πιθανοτικό φράγμα στο απαιτούμενο πλήθος δειγμάτων όταν η συνάρτηση προς προσέγγιση ικανοποιεί ορισμένες συνήθεις συνθήκες. Ακόμα, μελετούμε πρόσφατες μεθόδους αποδοτικής βελτιστοποίησης για μια κλάση δομημένων μη-κυρτών προβλημάτων, τα οποία περιλαμβάνουν περιορισμούς και κόστη που μπορούν να γραφούν ως συνδυασμός κυρτών συναρτήσεων, γνωστά ως DC προγράμματα.

Έπειτα, προχωρούμε στη μελέτη ενός μη-γραμμικού μοντέλου νευρώνα, το οποίο ονομάζεται Τροπικό ή Μορφολογικό Perceptron. Αρχικά, επιχειρείται η διερεύνηση βασικών γεωμετρικών ιδιοτήτων του υπό το πρίσμα της δυαδικής ταξινόμησης. Σε επόμενο βήμα, προτείνεται ένας αλγόριθμος ο οποίος συνδέει την εκπαίδευση του Τροπικού Perceptron με την επίλυση ενός κατάλληλου στιγμιοτύπου της προαναφερθείσας κλάσης μη-κυρτών προβλημάτων. Ο αλγόριθμος εκμεταλλεύεται μια τεχνική χαρακτηρισμού "ασυνήθιστων" δεδομένων για την κατάλληλη στάθμιση της συμβολής τους στην εκπαίδευση, σε μια προσπάθεια να επιτύχουμε ευρωστία. Ακόμη, σχολιάζεται η καταλληλότητα της επιλογής της προαναφερθείσας κλάσης προβλημάτων και οι αναλλοίωτες της μεθόδου.

Σε όλη σχεδόν την έκταση της εργασίας, αναδεικνύεται η συσχέτιση μεταξύ των μορφολογικών υπολογιστικών μοντέλων και του αναπτυσσόμενου πεδίου της Τροπικής Γεωμετρίας, η οποία μας προσφέρει διαίσθηση αλλά και αναλυτικά εργαλεία για την μελέτη βασικών χαρακτηριστικών των μοντέλων αυτών. Καταφέρνουμε να συσχετίσουμε το πλήθος των γραμμικών περιοχών μιας δημοφιλούς δομικής μονάδας βαθιών νευρωνικών δικτύων με ένα καλά μελετημένο γεωμετρικό αντικείμενο, το Πολύτοπο Newton ενός τροπικού πολυωνύμου. Η διερεύνηση αυτή μας επιτρέπει ταυτόχρονα να διατυπώσουμε μια διαδικασία για την απομάκρυνση των περιττών όρων από τροπικά πολυώνυμα πολλών μεταβλητών.

Τέλος, παρουσιάζουμε τα αποτελέσματα αριθμητικών πειραμάτων, στα οποία σχολιάζεται η καταλληλότητα και η συμπεριφορά των Μορφολογικών Νευρωνικών Δικτύων για προβλήματα μηχανικής μάθησης έπειτα από εκπαίδευση σε διάφορα σύνολα δεδομένων που έχουν χρησιμοποιηθεί ως benchmarks στη βιβλιογραφία.

Λέξεις-κλειδιά: Μαθηματική Μορφολογία · Τροπική Γεωμετρία · Βελτιστοποίηση · Νευρωνικά Δίκτυα · Μηχανική Μάθηση

Abstract

Within the last decade, the scientific discipline of Artificial Intelligence has been marked by the steep rise of the use of Neural Networks. These have traditionally relied on mostly linear models, such as the multiply-accumulate architecture of the Linear Perceptron which remains the dominant model of neuronal computation. However, from a biological standpoint, a neuron's activity might as well rely on inherently nonlinear and competitive operations. Mathematical Morphology and Minimax Algebra provide the necessary framework consisting of such nonlinear units.

Effective usage of Neural Networks almost always requires solving "difficult" optimization problems. For that reason, the current thesis initially examines a recently proposed method for smoothing of nonconvex problems, for the case where the computations it involves do not allow for an analytical form. We show that this method can be approximated by sampling methods and prove some pertinent results, among which a concentration bound on the number of required samples when the approximated function satisfies some common properties. Additionally, we study recently proposed efficient optimization methods for a class of structured nonconvex problems which include costs and inequalities that can be written as a combination of convex functions, also known as DC programs.

We then move on to the study of a nonlinear neuron, which is called a Tropical (or Morphological) Perceptron. Initially, we attempt to explore its fundamental geometric properties in the context of binary classification. As a next step, we propose an algorithm which related the Tropical Perceptron's training to the solution of an appropriate instance of the aforementioned class of DC programs. The algorithm also utilizes a simple form of outlier ablation in order to weigh their contribution to training, in an effort towards robustness. Additionally, we comment on the fitness of the chosen class of problems and the method's invariants.

Throughout almost all of the course of the thesis, the relationship between morphological computational models and the developing field of Tropical Geometry is highlighted. Tropical Geometry offers us intuition as well as analytical tools to study fundamental properties of those models. We manage to relate the number of linear regions of a popular choice for deep neural networks with a well-studied geometric object, namely the Newton Polytope of a tropical polynomial. Our exploration at the same time enables us to state a procedure for removing redundant terms from tropical polynomials of multiple variables.

Finally, we present the results of numerical experiments, in which we also comment on the aptness and the general behaviour of Morphological Neural Networks with regards to problems in machine learning, after training them on several datasets used as benchmarks in recent literature.

Keywords: Mathematical Morphology · Tropical Geometry · Optimization · Neural Networks · Machine Learning

Ευχαριστίες

Η διπλωματική αυτή κλείνει τον κύκλο των προπτυχιακών μου σπουδών και ταυτόχρονα είναι η πρώτη μου εκτεταμένη ερευνητική απόπειρα. Θέλω να ευχαριστήσω ιδιαίτερα -μεταξύ άλλων και για την υπομονή του- τον κύριο Πέτρο Μαραγκό, ο οποίος δεν αντιμετώπισε την επίβλεψη της διπλωματικής μου διεκπεραιωτικά, αλλά αντίθετα παρείχε αξιοσημείωτη ελευθερία κινήσεων και επιλογών, μαζί με συνεχή ενθάρρυνση και χρήσιμες συμβουλές. Με τον τρόπο αυτό έκανε τη συνεργασία μας ευχάριστη και παραγωγική και ταυτόχρονα επέτρεψε τη μετουσίωσή της σε μια πρωτότυπη ερευνητική συμβολή. Ακόμη, η στήριξή του για τα επόμενα βήματα της ακαδημαϊκής μου πορείας υπήρξε ανεκτίμητη.

Θέλω επίσης να ευχαριστήσω διάφορους ανθρώπους που συνέβαλαν τα μέγιστα ώστε τα σχεδόν 7 χρόνια που πέρασα στο Πολυτεχνείο να είναι τα καλύτερα που έχω περάσει μέχρι στιγμής. Το Δημήτρη, φίλο και συνοδοιπόρο όλα τα φοιτητικά μου χρόνια και σταθερή πρώτη επιλογή συνεργάτη για projects. Τη Βασιλική, η οποία συνεχίζει να με στηρίζει και να με ανέχεται από το πρώτο έτος, κόντρα σε όλες τις επιστημονικές αντενδείξεις. Τη Χλόη, της οποίας κάθε προσπάθεια να με κάνει άνθρωπο έχει αποτύχει παταγωδώς. Τον Κωστή, για τα παραγωγικά ξενύχτια και την υπαρξιακή αγωνία, κατά κανόνα μπροστά από ένα terminal. Την Ελένη, που μου έμαθε τι σημαίνει production και ταυτόχρονα "προσπάθησε" να με κάνει σοβαρό επαγγελματία. Ακόμη, το Βασίλη Κ., μεταξύ άλλων για τα χρήσιμα pointers του στο τελευταίο στάδιο της διπλωματικής.

Τέλος, ένα ευχαριστώ σε όσους έβαλαν (από νωρίς!) το λιθαράκι τους ώστε να μην περάσω όλη μου τη φοιτητική ζωή στα αμφιθέατρα, είτε για να βρεθούμε μαζί σε κάποια συνέλευση, είτε απλά για να μοιραστούμε έναν καφέ στα τραπεζάκια.

Βασίλης Χαρισόπουλος

Αθήνα, 1 Απριλίου 2017

Περιεχόμενα

1 Εισαγωγή **7**
- 1.1 Στόχοι και περιεχόμενα της Εργασίας . . . 8
  - 1.1.1 Κίνητρα και στόχοι . . . 8
  - 1.1.2 Δομή και συνεισφορές . . . 9
- 1.2 Συμβολισμός . . . 10
- 1.3 $(\max, +)$ άλγεβρα . . . 11
- 1.4 Στοιχεία Πολυεδρικής Γεωμετρίας . . . 14
- 1.5 Κυρτές Συναρτήσεις & Προβλήματα Κυρτής Βελτιστοποίησης . . . 16
  - 1.5.1 Βασικές έννοιες κυρτής ανάλυσης . . . 16
  - 1.5.2 Disciplined Convex Programming . . . 18
  - 1.5.3 DC Programming . . . 19

2 Νευρωνικά Δίκτυα **23**
- 2.1 Το Perceptron του Rosenblatt . . . 24
  - 2.1.1 Συναρτήσεις ενεργοποίησης . . . 24
- 2.2 Πολυεπίπεδα νευρωνικά δίκτυα . . . 26
  - 2.2.1 Συναρτήσεις Κόστους . . . 27

3 Μη-Κυρτή Βελτιστοποίηση **31**
- 3.1 Βελτιστοποίηση με Εξομάλυνση μέσω Ισοτροπικής Διάχυσης . . . 32
  - 3.1.1 Εξομάλυνση με μεθόδους Monte Carlo . . . 34
  - 3.1.2 Επιλογή δειγμάτων . . . 38
  - 3.1.3 Αριθμητική Προσέγγιση της Ισοτροπικής Διάχυσης . . . 46
- 3.2 Η Convex-Concave διαδικασία . . . 50
  - 3.2.1 Convex-Concave Procedure χωρίς περιορισμούς . . . 51
  - 3.2.2 Convex-Concave Procedure με DC περιορισμούς . . . 52
  - 3.2.3 Penalty CCP . . . 53

4 Το Τροπικό Perceptron **55**
- 4.1 Το τροπικό perceptron . . . 56
  - 4.1.1 Το τροπικό perceptron ως μορφολογικό φίλτρο . . . 57
  - 4.1.2 Η γεωμετρία του τροπικού perceptron . . . 58
  - 4.1.3 Convex-Concave Programming για τον προσδιορισμό των w_i . . . 62

4.1.4 Ευρετικές μέθοδοι για περιπτώσεις μη-διαχωρισιμότητας 64
4.2 Πολυωνυμικός Αλγόριθμος Ανάθεσης Βαρών 68
4.3 Μη-γραμμικότητα: η συνάρτηση ReLU 72
4.4 Τροπικές Μηχανές Διανυσμάτων Υποστήριξης 75
4.4.1 Τροπικές καμπύλες 75

5 Μορφολογικά Νευρωνικά δίκτυα **81**
5.1 Μορφολογικά νευρωνικά δίκτυα 82
5.1.1 Αρχιτεκτονική 82
5.1.2 Σχέση με νευρωνικά δίκτυα τύπου Maxout 83
5.1.3 Γραμμικές περιοχές 86
5.1.4 Πολύτοπα Newton και δίκτυα maxout 89
5.1.5 Απλοποίηση τροπικών πολυωνύμων 97
5.2 Εκπαίδευση 101
5.2.1 Gradient Descent 101
5.2.2 Convex-Concave Programming 102
5.2.3 Πειραματισμός σε Datasets 105

6 Επίλογος **113**
6.1 Ερευνητική Συμβολή 114
6.2 Μελλοντικές επεκτάσεις 115

Κατάλογος σχημάτων

1.1 Ισομορφισμοί μεταξύ τροπικών ημιδακτυλίων . . . 13
1.2 $\mathrm{Conv}(\{u_1, u_2, u_3, u_4\}) = \{u_1, u_2, u_3\}$ . . . 15
1.3 Η $f(x) = \mathrm{relu}(x)$ και 2 subgradients της . . . 17
1.4 Κανόνες παραγωγής της $h(\mathbf{x}) = ||2 \cdot \mathrm{square}(\mathbf{x}) + 3||_1$ . . . 20

2.1 Linear perceptron . . . 25
2.2 Συνήθεις επιλογές συναρτήσεων ενεργοποίησης . . . 26
2.3 Πολυεπίπεδο Perceptron με ένα κρυφό επίπεδο . . . 27

3.1 Valley function, $V(\boldsymbol{x}; 0.65, 0.2) * k_\sigma$ για διάφορες τιμές του σ . . . 34
3.2 Valley function, $V(\boldsymbol{x}; 0.65, 0.2) * k_\sigma$ για $\boldsymbol{x} \in \mathbb{R}$ . . . 35
3.3 Προσεγγίσεις της $g(\mathbf{x}) = [f * k_\sigma](\mathbf{x})$ για διαφορετικές πολιτικές δειγματοληψίας . . . 40
3.4 Προσέγγιση $g_\sigma(x), \sigma = 0.1, \epsilon = 0.1, \delta = 0.001$ . . . 48
3.5 Προσέγγιση $g_\sigma(x), \sigma = 0.5, \epsilon = 0.1, \delta = 0.001$ . . . 49
3.6 Προσέγγιση $g_\sigma(x), \sigma = 1, \epsilon = 0.1, \delta = 0.001$ . . . 49

4.1 Τροπικώς γραμμικά διαχωρίσιμα πρότυπα, $(w_1, w_2) = (-0.1, -0.3)$ . . . 60
4.2 Διαχωριστική επιφάνεια, $\boldsymbol{w} = (-0.20394, -0.4951)$ . . . 63
4.3 Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 10\%$ . . . 65
4.4 Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 15\%$ . . . 66
4.5 Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 20\%$ . . . 67
4.6 Ακρίβεια μεθόδου με και χωρίς ομαλοποίηση για συνθετικά δεδομένα . . . 68
4.7 Ακρίβεια μεθόδου με και χωρίς ομαλοποίηση για συνθετικά δεδομένα, περιστραμμένα με τυχαίο τρόπο . . . 69
4.8 $\zeta(x) = \log(1 + e^x), \zeta_h(x) = \lim_{h \to 0^+} h\zeta(x/h)$ . . . 73
4.9 Τροπική καμπύλη της Εξ. (4.27) . . . 76

5.1 Max-min tropical perceptron . . . 82
5.2 $f(x) = 3.7 \vee (-0.7x + 1) \vee (0.3x + 4) \vee (1.2x + 1)$ . . . 85
5.3 Max-min TP με $k = 2$ όρους . . . 90

5.4 $V(p) = \{-2, 1\}$ . . . 91
5.5 Newt $[\max(0, x + y, 3x, 2x + 2y, 3y)]$ . . . 92
5.6 $g(x, y)$ . . . 95
5.7 Πολύγωνα Newton για τους όρους της $g(x, y)$ . . . 96
5.8 Στάδια εφαρμογής του Αλγορίθμου 7 για το πολυώνυμο $p(x) = \max\{2, 2x+1, -2x+0.6, 0.5x - 1\}$ . . . 99
5.9 Το πολυώνυμο $p(x)$ και οι παράγοντές του . . . 100
5.10 Ripley's synthetic dataset . . . 105
5.11 `Ripleys` classification results . . . 106
5.12 Best result on accuracy . . . 107
5.13 DCCP output, `Ripleys` dataset . . . 107
5.14 WDBC dataset, two choices of feature plots . . . 108
5.15 `WDBC` classification results . . . 108
5.16 Παραδείγματα εικόνων ψηφίων από το `MNIST` dataset . . . 109
5.17 Αρχιτεκτονική δικτύου για το σύνολο χειρόγραφων ψηφίων `MNIST` . . . 110
5.18 Απεικόνιση των παραμέτρων του Dilation layer . . . 110
5.19 Απεικόνιση των L_i που αντιστοιχούν στις μέγιστες ενεργοποιήσεις . . . 110
5.20 Απεικόνιση βαρών για τα φίλτρα $L_i, i \in 1 \dots n_1$ της Εικόνας 5.17 . . . 111
5.21 Erosion Layers . . . 112

Κατάλογος πινάκων

3.1 Συναρτήσεις με τις εξομαλυμένες εκδοχές τους 47
3.2 Πλήθος δειγμάτων, $L = 1$, $\delta = 0.001$ 48

4.1 Σύγκριση μεθόδων εκπαίδευσης, 2 layers, `MNIST` dataset 80

5.1 Συγκριτικά αποτελέσματα των 2 δικτύων, $\lambda = 1$ 111
5.2 Συγκριτικά αποτελέσματα των 2 δικτύων, $\lambda = 0$ 112

Κεφάλαιο 1

Εισαγωγή

1.1 Στόχοι και περιεχόμενα της Εργασίας

1.1.1 Κίνητρα και στόχοι

Είναι σχεδόν σίγουρο πως ο 21ος αιώνας θα αποτυπωθεί, μεταξύ άλλων, ως ο αιώνας της μεγάλης έκρηξης της Τεχνητής Νοημοσύνης. Ένα συνεχώς αυξανόμενο κομμάτι των καθημερινών δραστηριοτήτων του ανθρώπου εκμεταλλεύεται ή στηρίζεται σε κάποια μορφή υπολογιστικής νοημοσύνης, η οποία τροφοδοτεί και ανατροφοδοτείται από την επιθυμία της κοινωνίας να αυτοματοποιεί ή να επιδιώκει να βελτιστοποιήσει διάφορες διαδικασίες, οι οποίες συνήθως περιλαμβάνουν στοιχεία απόφασης. Ένα χαρακτηριστικό παράδειγμα, το οποίο εμφανίζεται και στο κυρίως κομμάτι αυτής της εργασίας, είναι η αναγνώριση χειρόγραφων ψηφίων με σκοπό την ψηφιοποίησή τους με την ελάχιστη δυνατή ανάμειξη του ανθρώπινου παράγοντα.

Ωστόσο, παρά το γεγονός ότι η Τεχνητή Νοημοσύνη έχει εδραιωθεί ως πεδίο έρευνας μετά την δεκαετία του 1960 περίπου, οι εξελίξεις των τελευταίων ετών είναι αυτές που επιτρέπουν την πραγματοποίηση πολλών εκ των προαναφερθεισών διαδικασιών που αναφέραμε με ικανοποιητική ακρίβεια. Ένας από τους λόγους που πυροδότησαν αυτό το ποιοτικό άλμα είναι η "εκλαΐκευση" των Νευρωνικών Δικτύων. Πλέον, πίσω από σχεδόν κάθε υπηρεσία που λειτουργεί σε μεγάλη κλίμακα, λειτουργούν ένα ή περισσότερα νευρωνικά δίκτυα: από τις (αμφιλεγόμενες) διαφημιστικές προτάσεις της Google και τις προτεινόμενες αγορές στην Amazon, μέχρι αυτοματοποιημένη αναγνώριση χαρακτηριστικών ή αντικειμένων σε εικόνες και ομιλία. Η χρήση αρχιτεκτονικών πολλών επιπέδων (ενίοτε εκατοντάδων) σε αυτά τα δίκτυα οδήγησε στη γέννηση του ιδιώματος του Deep Learning (Βαθιάς Μηχανικής Μάθησης), η εξέλιξη του οποίου είναι τέτοια ώστε εκπαιδευμένα μοντέλα να έχουν καλύτερη επίδοση σε διάφορα tasks από αυτήν του ανθρώπου.

Η μεταστροφή του υπολογιστικού παραδείγματος που κυριαρχούσε στα νευρωνικά δίκτυα μέχρι πριν λίγα χρόνια, η οποία συνόδεψε την εδραίωση του Deep Learning, είναι ιδιαίτερα σημαντική ως σημείο αναφοράς αυτής της εργασίας. Τα κλασσικά μοντέλα νευρωνικών δικτύων στηρίζονταν κατά βάση σε μια γραμμική συνάρτηση ενεργοποίησης, χρησιμοποιώντας προαιρετικά μη-γραμμικότητες συναρτήσει των αναγκών κάθε εφαρμογής. Το πιο δημοφιλές μοντέλο, γνωστό ως Perceptron του Rosenblatt, στηρίχθηκε σε νευροφυσιολογικές παρατηρήσεις και ως εκ τούτου δε θεωρείται ασυνεπές με τη λειτουργία των νευρώνων ενός ζωντανού οργανισμού. Παρ'όλ' αυτά, βιολογικές ενδείξεις αναφέρουν πως η λειτουργία αυτή δεν είναι απαραίτητα κυρίως γραμμική, αλλά κατά περίπτωση περιλαμβάνει και εγγενώς μη-γραμμικές ή ανταγωνιστικές λειτουργίες. Τα πολυεπίπεδα (βαθιά) νευρωνικά δίκτυα είναι ένα βήμα προς αυτή τη θεώρηση, αφού παρεμβάλλουν μη-γραμμικούς υπολογισμούς ανάμεσα σε γραμμικές ενεργοποιήσεις.

Ωστόσο, μια προσέγγιση η οποία δεν έχει λάβει την προσοχή που της αναλογεί με βάση τις τελευταίες εξελίξεις, είναι αυτή των **μορφολογικών νευρωνικών δικτύων**. Τα δίκτυα αυτά αποτελούν γενικεύσεις ή μεταφορές των κλασσικών μοντέλων σε ένα υπολογιστικό παράδειγμα το οποίο δίνει έμφαση κυρίως σε μη-γραμμικούς υπολογισμούς. Όσον αφορά το αντικείμενο της μηχανικής μάθησης, η συνεισφορά τόσο θεωρητικών (όπως η περιγραφή της γεωμετρίας ή της πολυπλοκότητας ενός μοντέλου) όσο και πρακτικών (όπως η κατασκευή αποδοτικών αλγορίθμων εκμάθησης) αποτελεσμάτων στον τομέα των μορφολογικών νευρωνικών δικτύων είναι αρκετά σημαντική και ταυτόχρονα ιδιαίτερα πρόσφορη για σύχρονη έρευνα.

Ένα τρέχον αναπτυσσόμενο πεδίο των μαθηματικών φαίνεται να κάνει τη μελέτη αυτών των

νευρωνικών δικτύων με συστηματικό τρόπο εφικτή: το πεδίο της **τροπικής γεωμετρίας** (γνωστής και ως $(\max, +)$ άλγεβρας). Η γεωμετρία αυτή προκύπτει με φυσικό τρόπο μελετώντας μη-γραμμικούς χώρους και αλγεβρικά συστήματα και έχει βοηθήσει την επιστημονική κοινότητα να διατυπώσει κομψά αποτελέσματα σχετικά με προβλήματα συνδυαστικής και αλγορίθμων πάνω σε διακριτά αντικείμενα, την ανακάλυψη / επιβεβαίωση κλασσικών θεωρημάτων μέσω μιας διαφορετικής οδού, αλλά και διάφορα ερωτήματα σχετικά με βασικές ιδιότητες στατιστικών μοντέλων ή μοντέλων μάθησης. Το δρόμο αυτό επιχειρεί να ακολουθήσει και η παρούσα εργασία, η οποία στοχεύει ταυτόχρονα να ενοποιήσει με αυτό τον τρόπο κομμάτια της θεωρίας που διέπει τα μορφολογικά μοντέλα μάθησης και εν γένει να συνδέσει τις 2 ερευνητικές περιοχές. Η κατανόηση και θεωρητική περιγραφή ορισμένων από αυτά τα μοντέλα μπορεί ταυτόχρονα να μας δώσει περαιτέρω διορατικότητα ή απλά αναλυτικά εργαλεία για την ανάλυση των μοντέλων του Deep Learning, την καταλληλότητα και την απόδοση των οποίων αποτυγχάνουμε σε μεγάλο βαθμό να ερμηνεύσουμε.

Τέλος, η εργασία αυτή αγγίζει παράλληλα και το κομμάτι της μαθηματικής βελτιστοποίησης, η οποία συνοδεύει κάθε πρόβλημα μηχανικής μάθησης αλλά ταυτόχρονα αποτελεί και ένα αυθύπαρκτο ενδιαφέρον. Μας απασχολεί κυρίως η οικογένεια των συναρτήσεων που καλείται συνήθως να βελτιστοποιήσει ένα νευρωνικό δίκτυο προκειμένου να "εκπαιδευθεί": συναρτήσεις που είναι ιδιαίτερα δύσκολο να βελτιστοποιηθούν ακόμη και τοπικά, των οποίων η δομή συχνά δεν είναι εκμεταλλεύσιμη (ή δεν υπάρχει καν). Πέρα από την κατασκευή αποδοτικών αλγορίθμων για πρακτικά ζητήματα, είναι χρήσιμο να αντιμετωπίζουμε το πρόβλημα της μαθηματικής βελτιστοποίησης αφαιρετικά: μεταξύ άλλων, η προσέγγιση αυτή μας επιτρέπει αφενός να συνδέουμε φαινομενικά άσχετα προβλήματα μεταξύ τους, αφετέρου να αναπτύξουμε νέες, ακριβείς ή προσεγγιστικές, μεθόδους για μια μεγάλη οικογένεια δυσεπίλυτων μαθηματικών προβλημάτων.

1.1.2 Δομή και συνεισφορές

Περιγράφουμε σύντομα τη δομή της εργασίας, μαζί με τις συνεισφορές που περιέχει η κάθε ενότητα. Το υπόλοιπο του Κεφαλαίου 1 αποτελεί μια εισαγωγή στα πιο θεωρητικά κομμάτια που αγγίζει η εργασία, όπως την $(\max, +)$ άλγεβρα και τη θεωρία κυρτής βελτιστοποίησης, ενώ περιγράφεται εν συντομία ο συμβολισμός που ακολουθείται. Στο Κεφάλαιο 2, παρουσιάζεται το κλασσικό μοντέλο perceptron και νευρωνικού δικτύου, μαζί με τις συναρτήσεις κόστους που συνήθως χρησιμοποιούνται για την εκπαίδευσή τους.

Το Κεφάλαιο 3 καταπιάνεται με τη βελτιστοποίηση μη-κυρτών συναρτήσεων σε δομημένα και μη προβλήματα. Στο κεφάλαιο αυτό, αρχικά εξετάζουμε έναν μετα-αλγόριθμο ελαχιστοποίησης με τη χρήση μιας μεθόδου *ομοτοπίας*, την οποία προσεγγίζουμε αριθμητικά για περιπτώσεις όπου αναλυτικοί υπολογισμοί δεν είναι εφικτοί. Καταφέρνουμε να εξάγουμε ένα πιθανοτικό φράγμα του απαιτούμενου πλήθος δειγμάτων για ένα διάστημα εμπιστοσύνης της προσέγγισης, το οποίο στηρίζεται στην ανισότητα του Hoeffding μιας οικογένειας συναρτήσεων οι οποίες είναι φραγμένες ή Lipschitz-συνεχείς. Το υπόλοιπο του κεφαλαίου παρουσιάζει μια πρόσφατη μέθοδο για ελαχιστοποίηση συναρτήσεων που μπορούν να γραφούν ως διαφορά κυρτών συναρτήσεων και εμφανίζονται στην εκπαίδευση τροπικών (μορφολογικών) perceptrons.

Στο Κεφάλαιο 4, παρουσιάζεται το μοντέλο του τροπικού perceptron, ενώ γίνεται μια διερεύνηση των γεωμετρικών του χαρακτηριστικών με τη βοήθεια της Τροπικής Γεωμετρίας. Με βάση τις θεωρητικές μας παρατηρήσεις, διατυπώνουμε έναν πολυωνυμικό αλγόριθμο απόφασης για το αν ένα

σύνολο δεδομένων μπορεί να διαχωριστεί πλήρως από ένα τροπικό perceptron, καθώς και έναν αλγόριθμο που ανάγει το πρόβλημα εκπαίδευσης στην κλάση δομημένων προβλημάτων που αναφέρθηκε στο δεύτερο κομμάτι του Κεφαλαίου 3, του οποίου σχολιάζονται η ακρίβεια και η ευρωστία. Ακόμη, παρουσιάζεται μια γενίκευση ενός πρόσφατα προτεινόμενου τροπικού μοντέλου, χωρίς ωστόσο να γίνεται εκτενής διερεύνηση.

Τέλος, το Κεφάλαιο 5 αρχικά αντιμετωπίζει τα μορφολογικά νευρωνικά δίκτυα από μια πιο πρακτική σκοπιά, ως κομμάτια πιο σύνθετων αρχιτεκτονικών. Ταυτόχρονα, παρατηρείται η καταλληλότητα της περιγραφής μιας δημοφιλούς δομικής μονάδας βαθιών νευρωνικών δικτύων, ως πολυωνύμου στην τροπική γεωμετρία. Η παρατήρηση αυτή μας επιτρέπει να διατυπώσουμε νέα φράγματα για το πλήθος των "απλών" περιοχών που περιγράφει μια τέτοια μονάδα, το οποίο είναι χρήσιμο αποτέλεσμα για την ανάλυση της πολυπλοκότητας σύγχρονων νευρωνικών δικτύων. Γενικεύουμε τα φράγματα αυτά για 2 ελαφρώς πιο σύνθετα μοντέλα, ενώ ταυτόχρονα τα αναλυτικά εργαλεία που χρησιμοποιήσαμε στην προηγούμενη διαδικασία μας επιτρέπουν να διατυπώσουμε και έναν αλγόριθμο θεωρητικού ενδιαφέροντος, ο οποίος απομακρύνει περιττούς όρους από τροπικά πολυώνυμα. Στο τέλος του Κεφαλαίου, παρουσιάζονται ορισμένα αποτελέσματα από αριθμητικά πειράματα που σκοπό έχουν να αναδείξουν την απόδοση του αλγορίθμου εκπαίδευσης που προτείνεται στο Κεφάλαιο 4 σε σχέση με άλλες μεθόδους καθώς και την καταλληλότητα και συμπεριφορά των τροπικών perceptrons ως κομμάτια πολυεπίπεδων αρχιτεκτονικών. Σε ένα κλασσικό dataset, παρατηρείται η τάση των μορφολογικών ταξινομητών να επιβάλλουν αραιές αναπαραστάσεις στον χώρο των δεδομένων πάνω στον οποίο επιδρούν.

Το Κεφάλαιο 6 συνοψίζει και πάλι την ερευνητική συμβολή της εργασίας, ενώ προτείνονται ορισμένες μελλοντικές ερευνητικές κατευθύνσεις που μπορούν να ακολουθηθούν ως συνέχεια της Διπλωματικής με βάση τα νέα ερωτήματα που έχουν προκύψει.

1.2 Συμβολισμός

Πριν την εισαγωγή, είναι σκόπιμο να αναφερθούμε σύντομα στον συμβολισμό που χρησιμοποιείται στην παρούσα εργασία, ώστε ο αναγνώστης να μπορεί να παρακολουθήσει τις μαθηματικές σχέσεις και τη ροή του κειμένου χωρίς δυσκολία.

Αναφερόμαστε με το σύμβολο $\mathbb{R}$ στην ευθεία των πραγματικών αριθμών, $(-\infty, \infty)$, με το σύμβολο $\mathbb{N}$ στο σύνολο των φυσικών αριθμών $\mathbb{N} = 1, 2, \ldots$ και με το σύμβολο $\mathbb{Z}$ στο σύνολο των ακεραίων, $\mathbb{Z} = \ldots, -2, -1, 0, 1, 2, \ldots$. Εκτός από εξαιρέσεις που αναφέρονται ρητά ή είναι προφανείς από τα συμφραζόμενα, συμβολίζουμε:

- με μικρά γράμματα συνηθισμένης έντασης (χωρίς bold) βαθμωτές ποσότητες, π.χ. $x \in \mathbb{R}$.
- με μικρά γράμματα σε bold γραμματοσειρά ποσότητες ή μεταβλητές που είναι διανύσματα ή μονοδιάστατοι πίνακες, π.χ. $\boldsymbol{x} \in \mathbb{Z}^n$.
- με κεφαλαία γράμματα σε bold γραμματοσειρά ποσότητες και μεταβλητές που είναι πίνακες 2 ή παραπάνω διαστάσεων, π.χ. $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

Υποθέτουμε ότι όλα τα διανύσματα είναι διανύσματα-στήλες, εκτός αν αναφέρεται ρητά κάτι διαφορετικό. Αναφερόμαστε στον ανάστροφο πίνακα του $\boldsymbol{A}$ με το σύμβολο $\boldsymbol{A}^T$. Ακόμη, χρησιμοποιούμε

δείκτες που ξεκινούν από 1 για να αναφερθούμε στα στοιχεία ενός διανύσματος ή ενός πίνακα, δηλαδή αν $\boldsymbol{a} \in \mathbb{R}^3$, $\boldsymbol{A} \in \mathbb{R}^{3\times 2}$, γράφουμε

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$
$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{pmatrix}$$

Ακόμη, έαν η f είναι συνάρτηση χρησιμοποιούμε το συμβολισμό $\mathbf{dom} f$ για να αναφερθούμε στο πεδίο ορισμού της, και το σύμβολο ∇f για να αναφερθούμε στην παράγωγο (gradient) της. Για παράδειγμα, αν $f : \mathbb{R}^2 \to \mathbb{R}$, ισχύουν:

$$\mathbf{dom} f = \mathbb{R}^2$$
$$\nabla f(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial f}{\partial x_1}(\mathbf{x}) \\ \\ \dfrac{\partial f}{\partial x_2}(\mathbf{x}) \end{pmatrix}$$

Τέλος, υπενθυμίζουμε τον ορισμό των ℓ_p νορμών, οι οποίες αναφέρονται συχνά στο κείμενο της εργασίας. Γενικά, μια νόρμα ορισμένη πάνω σε ένα διανυσματικό χώρο $\mathcal{V}$, ο οποίος δρα σε ένα σώμα $\mathbb{K}$, είναι μια συνάρτηση $p : \mathcal{V} \to \mathbb{R}$ η οποία ικανοποιεί τις εξής ιδιότητες: $\forall a \in \mathbb{K}$ και $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{V}$, ισχύουν

$$p(a\boldsymbol{v}) = |a| p(\boldsymbol{v})$$
$$p(\boldsymbol{u} + \boldsymbol{v}) \leq p(\boldsymbol{u}) + p(\boldsymbol{v})$$
$$p(\boldsymbol{v}) = 0 \Rightarrow \boldsymbol{v} = 0$$

Για τις ℓ_p νόρμες του διανυσματικού χώρου $\mathbb{R}^n$ με $\mathbf{x} \in \mathbb{R}^n$, ισχύει

$$||\mathbf{x}||_p \triangleq \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \tag{1.1}$$

Μια περίπτωση που χρήζει ιδιαίτερης προσοχής είναι όταν $p \to \infty$, οπότε προκύπτει

$$||\mathbf{x}||_\infty = \max\{|x_1|, |x_2|, \dots |x_n|\} \tag{1.2}$$

1.3 $(\max, +)$ άλγεβρα

Η $(\max, +)$ άλγεβρα είναι ένας ημιδακτύλιος ορισμένος στο σύνολο $\mathbb{R}_{\max} \triangleq \mathbb{R} \cup \{-\infty\}$, εφοδιασμένος με τις ακόλουθες διμελείς πράξεις[1]:

[1] Η επιστημονική κοινότητα στον τομέα της τροπικής γεωμετρίας συνήθως χρησιμοποιεί τα σύμβολα $\oplus, \odot$ αντίστοιχα, ωστόσο εδώ ακολουθούμε τον πλεγματοθεωρητικό συμβολισμό που βρίσκεται πιο κοντά στην κοινότητα της μαθηματικής μορφολογίας, όπως ακολουθείται π.χ. στο [41]

- $x \vee y = \max\{x, y\}$ ως την πράξη της γενικευμένης "πρόσθεσης"
- $x + y$ ως την πράξη του γενικευμένου "πολλαπλασιασμού"

Αντίστοιχα, ορίζουμε ένα διάνυσμα $\mathbf{x} \in \mathbb{R}^n_{\max}$ ως

$$\mathbf{x} = [x_1, x_2, \dots x_n]^T$$

με $x_i \in \mathbb{R}_{\max}, \forall i \in N = [1 \dots n]$ και έναν πίνακα $A \in \mathbb{R}^{m \times n}_{\max}$ με τον ίδιο τρόπο που ορίζεται στην κλασσική άλγεβρα. Οι πράξεις πινάκων στη $(\max, +)$ άλγεβρα ορίζονται ως εξής:

$$(\boldsymbol{A} \vee \boldsymbol{B})_{ij} = \max(a_{ij}, b_{ij}) \text{ για κατάλληλους πίνακες } \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}_{\max} \tag{1.3}$$

$$(\boldsymbol{A} \boxplus \boldsymbol{B})_{ij} = \bigvee_{q=1}^{k} a_{iq} + b_{qj} \text{ για } \boldsymbol{A} \in \mathbb{R}^{m \times k}_{\max}, \boldsymbol{B} \in \mathbb{R}^{k \times n}_{\max} \tag{1.4}$$

Παρουσιάζουμε μερικά παραδείγματα στη συνέχεια:

- "Εσωτερικό γινόμενο" διανυσμάτων:

$$[1, -4, 0, 1] \boxplus [0, -9, 4, 6]^T = (1 + 0) \vee (-4 - 9) \vee (0 + 4) \vee (1 + 6) = 7$$

- Πολλαπλασιασμός τετραγωνικών πινάκων:

$$\begin{pmatrix} 5 & 9 \\ -7 & 6 \end{pmatrix} \boxplus \begin{pmatrix} -1 & 0 \\ -2 & 4 \end{pmatrix} = \begin{pmatrix} (5-1) \vee (9-2) & (5-0) \vee (9+4) \\ (-7-1) \vee (6-2) & (-7-0) \vee (6+4) \end{pmatrix}$$

$$= \begin{pmatrix} 7 & 13 \\ 4 & 10 \end{pmatrix}$$

Από τα παραπάνω παραδείγματα γίνεται σαφές ότι το 0 αποτελεί το ουδέτερο στοιχείο ως προς τη διμελή πράξη $\boxplus$, ενώ κάθε στοιχείο $a \in \mathbb{R}_{\max} \setminus \{-\infty\}$ διαθέτει πολλαπλασιαστικό αντίστροφο $a^{-1} = -a$. Όσον αφορά την πράξη $\vee$, συμβολίζουμε $\varepsilon = -\infty$ το "ουδέτερο" στοιχείο, αφού ισχύει $\max(-\infty, a) = a, \forall a \in \mathbb{R}_{\max}$. Ο "μοναδιαίος" πίνακας, δηλαδή ο πίνακας $\boldsymbol{I}$ που ικανοποιεί την ιδιότητα

$$\boldsymbol{I} \boxplus \boldsymbol{A} = \boldsymbol{A} \boxplus \boldsymbol{I} = \boldsymbol{A} \tag{1.5}$$

στη $(\max, +)$ άλγεβρα είναι ο πίνακας με 0 στη διαγώνιο και ε σε όλες τις υπόλοιπες θέσεις, δηλαδή

$$\boldsymbol{I}_{n \times n} = \begin{pmatrix} 0 & \varepsilon & \dots & \varepsilon \\ \varepsilon & 0 & \dots & \varepsilon \\ \vdots & & & \vdots \\ \varepsilon & \varepsilon & \dots & 0 \end{pmatrix} \tag{1.6}$$

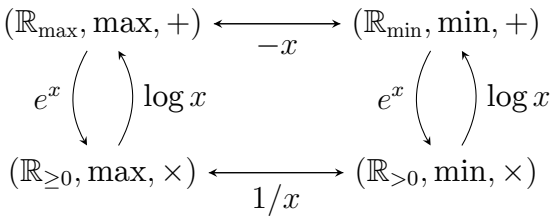
Η $(\max, +)$ άλγεβρα είναι ένα παράδειγμα ταυτοδύναμου (**idempotent**) ημιδακτυλίου [24]: ιδιοδύναμος δακτύλιος αποκαλείται κάθε ημιδακτύλιος $(R, \diamond, \star)$ στον οποίο η διμελής πράξη της "πρόθεσης", συμβολικά $\diamond$, είναι ιδιοδύναμη:

$$a \diamond a = a, \quad \forall a \in R \tag{1.7}$$

Όμοια με την $(\max, +)$ άλγεβρα ορίζεται η $(\min, +)$ άλγεβρα, άλλη μια περίπτωση ιδιοδύναμου ημιδακτυλίου, στην οποία η διμελής πράξη $\vee$ αντικαθιστάται από την $\wedge$ (min), ενώ το σύνολο στο οποίο είναι ορισμένη είναι πλέον το $\mathbb{R}_{\min} \triangleq \mathbb{R} \cup \{+\infty\}$. Αντίστοιχα με πριν, οι πράξεις μεταξύ πινάκων ορίζονται ως:

$$\begin{aligned} (\boldsymbol{A} \wedge \boldsymbol{B})_{ij} &= \min(a_{ij}, b_{ij}) \\ (\boldsymbol{A} \boxplus' \boldsymbol{B})_{ij} &= \bigwedge_{q=1}^{k} a_{iq} + b_{qj} \end{aligned} \tag{1.8}$$

Όπως φαίνεται στην Εξίσωση 1.8, χρησιμοποιούμε το σύμβολο $\boxplus'$ ώστε να αναφερόμαστε στον πολλαπλασισμό πινάκων στη $(\min, +)$ άλγεβρα. Ο ημιδακτύλιος $(\mathbb{R}_{\max}, \max, +)$ είναι *ισομορφικός* με τον ημιδακτύλιο $(\mathbb{R}_{\min}, \min, +)$, αφού χρησιμοποιώντας τον ισομορφισμό $\phi(x) = -x$, με τη σύμβαση ότι $\phi(\pm\infty) = \mp\infty$, μεταβαίνουμε από τη μια περίπτωση στην άλλη.[2] Στο Σχήμα 1.1 φαίνονται οι φυσικοί ισομορφισμοί μεταξύ των διαφόρων ημιδακτυλίων της τροπικής άλγεβρας, όπου $\mathbb{R}_{>0} \triangleq (0, \infty]$, $\mathbb{R}_{\geq 0} \triangleq [0, \infty)$. Λόγω του γεγονότος ότι η δομή $(\mathbb{R}_{\max}, \max)$ είναι ένα μονοειδές



Σχήμα 1.1: Ισομορφισμοί μεταξύ τροπικών ημιδακτυλίων

(monoid) χωρίς αντίστροφο για κάθε στοιχείο, διάφορες ιδιότητες που έχουμε "συνηθίσει" να ισχύουν στη γραμμική άλγεβρα δεν μεταφέρονται αυτόματα στη $(\max, +)$ άλγεβρα. Για παράδειγμα, οι τετραγωνικοί πίνακες $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ διαθέτουν αντίστροφο αν και μόνο αν είναι διαγώνιοι. Ορισμένοι ερευνητές όπως ο Izhakian [32] προσπαθούν να επιλύσουν τέτοια προβλήματα μεταβαίνοντας σε μια αλγεβρική δομή που "συμμετροποιεί" τον $(\max, +)$-ημιδακτύλιο, χρησιμοποιώντας στοιχεία που είναι διατεταγμένα ζεύγη (a, b) στοιχείων του $\mathbb{R}_{\max}$. Παρ'όλ' αυτά, κάποια από τα προβλήματα που μας ενδιαφέρουν σε αυτή την εργασία μπορούν να αντιμετωπιστούν χωρίς να ξεφύγουμε από τη βασική δομή $(\mathbb{R}_{\max}, \max, +)$. Ένα από αυτά είναι η επίλυση ενός συστήματος γραμμικών εξισώσεων, στη μορφή

$$\boldsymbol{A} \boxplus \mathbf{x} = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}_{\max}^{m \times n}, \mathbf{x} \in \mathbb{R}_{\max}^{n}, \boldsymbol{b} \in \mathbb{R}_{\max}^{m} \tag{1.9}$$

Η συνθήκη για την ύπαρξη λύσης στην Εξ. (1.9), καθώς και η λύση $\overline{\mathbf{x}}$ που την ικανοποιεί, εφόσον υπάρχει, έχει δοθεί ήδη από τον Cuninghame-Green, και περιγράφεται στην Πρόταση 1.

[2] Η ερευνητική κοινότητα της τροπικής γεωμετρίας συχνά αναφέρεται σε έναν εκ των $(\mathbb{R}_{\max}, \max, +), (\mathbb{R}_{\min}, \min, +)$ με τον όρο *τροπικός ημιδακτύλιος*. Ο όρος αυτός εισήχθη παιγνιωδώς από τον Dominique Perrin προς τιμήν του Βραζιλιάνου μαθηματικού Imre Simon. Ένα άλλο παράδειγμα είναι ο ημιδακτύλιος $(\mathbb{R}_{+}, \max, \times)$, γνωστός και ως υποτροπικός (subtropical) ημιδακτύλιος.

Πρόταση 1 ([13]). Έστω $\boldsymbol{A} \in \mathbb{R}_{\max}^{m\times n}, \mathbf{x} \in \mathbb{R}_{\max}^{n}, b \in \mathbb{R}_{\max}^{m}$. Τότε η εξίσωση

$$\boldsymbol{A} \boxplus \mathbf{x} = b$$

έχει λύση μόνο αν το $\overline{\mathbf{x}}$, που ορίζεται ως

$$\overline{\mathbf{x}} = \boldsymbol{A}^{\sharp} \boxplus' b, \quad \boldsymbol{A}^{\sharp} \triangleq -\boldsymbol{A}^{T} \tag{1.10}$$

αποτελεί λύση του. Ακόμη, το $\overline{\mathbf{x}}$ είναι η μεγαλύτερη λύση της ανισότητας

$$\boldsymbol{A} \boxplus \mathbf{x} \leq b$$

Ο πίνακας $\boldsymbol{A}^{\sharp}$ συχνά αναφέρεται στη βιβλιογραφία ως ο *αντίστροφος Cuninghame-Green* (Cuninghame-Green inverse).

1.4 Στοιχεία Πολυεδρικής Γεωμετρίας

Σε επόμενα κεφάλαια, θα χρειαστούμε ορισμένες βασικές έννοιες από την περιοχή της πολυεδρικής γεωμετρίας. Βασικό σύγγραμμα-αναφορά για το σκοπό αυτό θα αποτελέσει το βιβλίο του G.M. Ziegler [70]. Αρχικά αναφέρουμε κάποιους χρήσιμους ορισμούς:

Ορισμός 1 (Κυρτό (convex) σύνολο). Έστω σύνολο $S \subset \mathcal{V}$, όπου $\mathcal{V}$ ένας διανυσματικός χώρος (ή διατεταγμένο πεδίο γενικότερα). Το σύνολο S αποκαλείται *κυρτό* αν:

$$x, y \in S \Rightarrow tx + (1-t)y \in S, \ \forall t \in [0,1] \tag{1.11}$$

Με βάση τον παραπάνω ορισμό, μπορούμε να ορίσουμε τον *κυρτό συνδυασμό* ενός συνόλου σημείων ως

$$C(x_1, \ldots, x_k) = \sum_{i=1}^{k} x_i\lambda_i, \lambda_i \geq 0, \sum_i \lambda_i = 1$$

Για παράδειγμα, στο χώρο $\mathbb{R}$, ένα κυρτό σύνολο είναι ένα ευθύγραμμο τμήμα που ορίζεται από τα σημεία εκκίνησης και τερματισμού. Το ευθύγραμμο τμήμα $[x_1, x_2]$ όπου $x_1 \leq x_2$ μπορεί να περιγραφεί παραμετρικά ως

$$[x_1, x_2] = x_1 + t(x_2 - x_1) = (1-t)x_1 + tx_2, \ t \in [0,1] \tag{1.12}$$

Στον $\mathbb{R}^2$, το απλούστερο κυρτό σύνολο πλήρους διάστασης $d = 2$ είναι το τρίγωνο με κορυφές τα σημεία x_1, x_2, x_3, αφού για κάθε σημείο x εσωτερικά του τριγώνου ισχύει

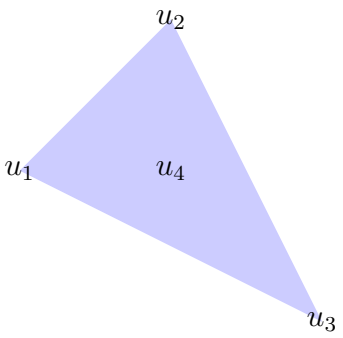
$$x = t_1x_1 + t_2x_2 + t_3x_3, \quad t_1 + t_2 + t_3 = 1, t_i \geq 0$$

Γενικότερα, στις d διαστάσεις, το απλούστερο κυρτό σύνολο πλήρους διάστασης που μπορούμε να ορίσουμε, όπως υποδεικνύεται από το θεώρημα Καραθεοδωρή, αποτελείται από τον κυρτό συνδυασμό $d+1$ σημείων και ονομάζεται $d-$simplex. Ο κυρτός συνδυασμός ενός συνόλου σημείων μας δίνει το ακόλουθο εξαιρετικά σημαντικό σύνολο:

Ορισμός 2 (Convex Hull)**.** Έστω $S = \{\mathbf{x}_1, \dots \mathbf{x}_k \in \mathbb{R}^d\}$ ένα σύνολο σημείων στις d διαστάσεις. Το κυρτό κάλυμμα (convex hull) του S ορίζεται ως το σύνολο

$$\text{conv}(S) = \left\{ \sum_{i=1}^{K} \lambda_i \mathbf{x}_i, \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\} \tag{1.13}$$

Σε κάθε περίπτωση (π.χ. ακόμα και αν το σύνολο S δεν δύναται να οριστεί ως ο κυρτός συνδυασμός πεπερασμένου αριθμού σημείων), το convex hull αποτελεί το ελάχιστο κυρτό σύνολο που περιέχει το S. Στο Σχήμα 1.2 φαίνεται το Convex Hull ενός συνόλου σημείων στον $\mathbb{R}^2$, το οποίο είναι 2-simplex.



Σχήμα 1.2: $\text{Conv}(\{u_1, u_2, u_3, u_4\}) = \{u_1, u_2, u_3\}$

Ορισμός 3 ($\mathcal{V}$-Polytope)**.** Έστω S ένα πεπερασμένο σύνολο σημείων $\mathbf{x} \in \mathbb{R}^d$. Το κυρτό κάλυμμα (convex hull) του συνόλου S λέγεται $\mathcal{V}$-πολύτοπο.

Με άλλα λόγια, κάθε $\mathcal{V}$-πολύτοπο είναι convex hull ενός πεπερασμένου συνόλου σημείων. Υπάρχει ένας ακόμα τύπος πολυτόπου, το λεγόμενο $\mathcal{H}$-πολύτοπο, το οποίο ορίζεται μέσω αφινικών ανισοτήτων, καθεμία από τις οποίες ορίζει έναν ημιχώρο.

Ορισμός 4 ($\mathcal{H}$-Polytope)**.** Ένα $\mathcal{H}$-πολύτοπο είναι η τομή ενός πεπερασμένου πλήθους n κλειστών ημιχώρων (halfspaces) της μορφής $\boldsymbol{a}^T \cdot \mathbf{x} \leq b$, όπου $\boldsymbol{a}, \mathbf{x} \in \mathbb{R}^d, b \in \mathbb{R}$ το οποίο είναι φραγμένο.

Τελικά, αποκαλούμε *πολύτοπο* (polytope) ένα σύνολο σημείων στο $\mathbb{R}^d$ το οποίο μπορεί να αναπαρασταθεί είτε σαν $\mathcal{V}$- είτε σαν $\mathcal{H}$- πολύτοπο. Θα χρειαστούμε ακόμη την έννοια της *όψης* ενός πολυτόπου, ειδικά στο κεφάλαιο όπου θα χρησιμοποιήσουμε τις ιδιοτήτες ενός πολυτόπου Newton. Επομένως, ορίζουμε:

Ορισμός 5 (Όψη (face))**.** Έστω $P \subset \mathbb{R}^d$ ένα κυρτό πολύτοπο στις d διαστάσεις. Μια γραμμική ανίσωση τύπου $\boldsymbol{c}^T\mathbf{x} \leq c_0$ ισχύει στο P αν ικανοποιείται $\forall \mathbf{x} \in P$. Μια *όψη* του P είναι κάθε σύνολο

$$F = P \cap \{\mathbf{x} \in \mathbb{R}^d : \boldsymbol{c}^T\mathbf{x} = c_0\}$$

όταν η ανίσωση $\boldsymbol{c}^T\mathbf{x} \leq c_0$ ισχύει στο P.

Για τη διάσταση dim μιας όψης F_P ισχύει ότι $\dim(F_P) \leq \dim(P)$ και την χαρακτηρίζουμε ως:

- *κορυφή*, εάν $\dim(F_P) = 0$ (αποτελεί δηλαδή ένα μεμονωμένο σημείο)
- *ακμή*, εάν $\dim(F_P) = 1$
- *πλευρά* (**facet**), έαν $\dim(F_P) = \dim(P) - 1$. Οι πλευρές είναι οι μεγαλύτερες όψεις που αποτελούν γνήσια υποσύνολα του αρχικού πολυτόπου.

Σε επόμενη ενότητα, θα μας χρειαστεί η έννοια του αθροίσματος Minkowski 2 πολυτόπων, η οποία ουσιαστικά αναφέρεται στο διανυσματικό άθροισμα μεταξύ των σημείων των πολυτόπων:

Ορισμός 6 (Άθροισμα Minkowski)**.** Έστω $P, Q \in \mathbb{R}^n$ πολύτοπα. Τότε το άθροισμα Minkowski των P, Q, συμβολιζόμενο ως $P \oplus Q$, ορίζεται ως

$$P \oplus Q \triangleq \{\boldsymbol{p} + \boldsymbol{q} : \boldsymbol{p} \in P, \boldsymbol{q} \in Q\} \tag{1.14}$$

1.5 Κυρτές Συναρτήσεις & Προβλήματα Κυρτής Βελτιστοποίησης

Το μεγαλύτερο κομμάτι αυτής της ενότητας προέρχεται από τα βιβλίο των Boyd & Vanderberghe [8] και των Rockafellar & Wetts [55]. Ξεκινάμε δίνοντας ορισμένους απαραίτητους ορισμούς για κυρτές συναρτήσεις.

1.5.1 Βασικές έννοιες κυρτής ανάλυσης

Μια συνάρτηση $f : \mathbb{R}^n \to \mathbb{R}$ λέγεται **κυρτή** (convex) εάν το πεδίο ορισμού $\mathbf{dom}f$ της είναι κυρτό και ισχύει

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}), \text{ για κάθε } 0 \leq \theta \leq 1 \tag{1.15}$$

Εάν η ανισότητα ισχύει χωρίς ισότητα για $0 < \theta < 1$, η συνάρτηση f λέγεται *ισχυρά κυρτή* (strongly convex). Η f λέγεται κοίλη (concave) εάν η $-f$ είναι κυρτή. Ο ορισμός της Εξ. 1.15 είναι ο πιο γενικός ορισμός της κυρτής συνάρτησης. Μια παραγωγίσιμη συνάρτηση f είναι κυρτή εάν ισχύει

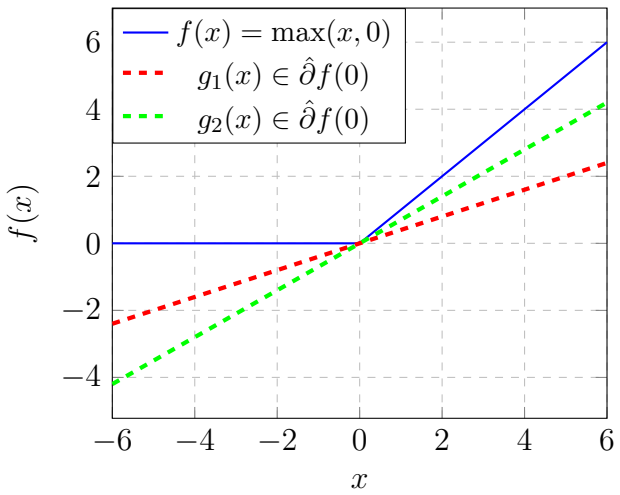
$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f^T(\mathbf{x})(y - x) \tag{1.16}$$

Η γεωμετρική εποπτεία της ιδιότητας (1.15) που ικανοποιεί κάθε convex συνάρτηση είναι πάρα πολύ σημαντική για τη χρησιμότητα των κυρτών συναρτήσεων στη βελτιστοποίηση. Η ιδιότητα (1.15) ουσιαστικά μας λέει ότι ο γράφος της συνάρτησης f ανάμεσα στα σημεία $\mathbf{x}, \mathbf{y}$ βρίσκεται πάντα κάτω από τη χορδή (ή επιφάνεια, ή υπερεπιφάνεια, ανάλογα με τη διάσταση του πεδίου ορισμού) που συνδέει τα σημεία $(\mathbf{x}, f(\mathbf{x})), (\mathbf{y}, f(\mathbf{y}))$.
Αντίστοιχα, η ιδιότητα (1.16) υποδεικνύει ότι η εφαπτομένη (ή το εφαπτόμενο επίπεδο) της f σε οποιοδήποτε σημείο $\mathbf{x}$ βρίσκεται κάτω από το γράφο της f σε ολόκληρο το $\mathbf{dom} f$. Επομένως, το ανάπτυγμα Taylor 1ης τάξης μιας κυρτής συνάρτησης f σε οποιοδήποτε σημείο της στο οποίο είναι παραγωγίσιμη, αποτελεί μια υποεκτιμήτριά της. Η ιδιότητα αυτή χρησιμοποιείται εκτενώς σε μεθόδους προσέγγισης προβλημάτων βελτιστοποίησης, όπως για παράδειγμα στο Convex-Concave Programming που θα παρουσιαστεί σε επόμενη ενότητα.

Δεν είναι ασυνήθιστο να καλούμαστε να χειριστούμε κυρτές συναρτήσεις οι οποίες είναι παραγωγίσιμες παντού με εξαίρεση μεμονωμένα σημεία (δηλαδή παντού εκτός από ένα σύνολο μέτρου 0), στα οποία ωστόσο είναι συνεχείς. Σε αυτή την περίπτωση, χρειαζόμαστε την έννοια της υποπαραγώγου (**subgradient**) [55]. Για τους σκοπούς αυτής της εργασίας, αρκεί η περιγραφή της παραγώγου που δίνεται ως Πρόταση στο βιβλίο των Rockafellar & Wetts.

Ορισμός 7 (Prop. 8.5, [55])**.** Έστω $f : \mathbb{R}^n \to \mathbb{R}$ και $\boldsymbol{v} \in \mathbb{R}^n$. Θα λέμε ότι το διάνυσμα $\boldsymbol{v}$ είναι υποπαράγωγος (subgradient) της f στο σημείο $\mathbf{x}$ εάν και μόνο εάν μπορεί να οριστεί συνάρτηση $h \leq f$ σε μια περιοχή του $\mathbf{x}$ ώστε $h(\mathbf{x}) = f(\mathbf{x})$ παραγωγίσιμη στο $\mathbf{x}$ με $\nabla h(\mathbf{x}) = \boldsymbol{v}$. Για λόγους ευκολίας, μπορούμε να θεωρήσουμε ότι $h(\mathbf{y}) < f(\mathbf{y}), \forall\ \mathbf{y} \neq \mathbf{x}$.
Συμβολίζουμε $\boldsymbol{v} \in \hat{\partial} f(\mathbf{x})$. Όταν η f είναι παραγωγίσιμη στο $\mathbf{x}$, ισχύει $\hat{\partial} f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$.

Η έννοια του subgradient χρησιμοποιείται συχνά στην κυρτή ανάλυση και το πεδίο του nonsmooth optimization, αφού αποτελεί μια κομψή γενίκευση της παραγώγου για σημεία όπου οι κυρτές συναρτήσεις δεν είναι παραγωγίσιμες. Στην Εικόνα 1.3 φαίνονται 2 subgradients της $\mathrm{relu}(x)$ στο σημείο



Σχήμα 1.3: Η $f(x) = \mathrm{relu}(x)$ και 2 subgradients της

0, όπου αυτή δεν είναι παραγωγίσιμη. Το σύνολο των subgradients της είναι

$$\hat{\partial} f(x) = \begin{cases} [0,1] \ (= \text{conv}\{0,1\}) & x = 0 \\ \{\nabla f(x)\} = \{0\} & x < 0 \\ \{\nabla f(x)\} = \{1\} & x > 0 \end{cases}$$

Ένα **πρόβλημα κυρτής βελτιστοποίησης** (convex optimization problem, ή convex program), είναι ένα πρόβλημα το οποίο περιλαμβάνει την βελτιστοποίηση μιας αντικειμενικής συνάρτησης (objective function) η οποία είναι κυρτή, πάνω σε ένα σύνολο εφικτών λύσεων το οποίο είναι κυρτό. Η πρωτοτυπική μορφή ενός convex program δίνεται με βάση τον Ορισμό 8 [8].

Ορισμός 8. Έστω $f_i : \mathbb{R}^n \to \mathbb{R}$ μια συλλογή κυρτών και $h_j : \mathbb{R}^n \to \mathbb{R}$ μια συλλογή αφφινικών συναρτήσεων. Τότε, ονομάζουμε πρόβλημα κυρτής βελτιστοποίησης κάθε πρόβλημα διατυπωμένο στην ακόλουθη μορφή:

$$\begin{aligned} &\text{Minimize } f_0(\mathbf{x}) \\ &\text{s.t. } \begin{cases} f_i(\mathbf{x}) \leq 0 & i = 1, \dots m \\ h_j(\mathbf{x}) = 0 & j = 1, \dots k \end{cases} \end{aligned} \tag{1.17}$$

Το σύνολο εφικτών λύσεων (feasible set) του προβλήματος (1.17) είναι το σύνολο

$$\mathbf{dom}_{\mathcal{F}} = \left\{ \mathbf{x} \in \left(\bigcap_i \mathbf{dom}(f_i) \right) \cap \left(\bigcap_j \mathbf{dom}(h_j) \right) : \begin{array}{ll} f_i(\mathbf{x}) \leq 0 & i \in 1, \dots m \\ h_j(\mathbf{x}) = 0 & j \in 1, \dots k \end{array} \right\} \tag{1.18}$$

Προφανώς, στο πρόβλημα (1.17) μπορεί να ισχύει $\mathbf{dom}_{\mathcal{F}} = \emptyset$, επομένως να μην έχει λύση. Αντίστοιχα, εάν το σύνολο εφικτών λύσεων είναι άπειρο, ενδέχεται η ελάχιστη τιμή της $f_0(\mathbf{x})$ να είναι $-\infty$, όπως για παράδειγμα στην ελαχιστοποίηση μιας αφφινικής συνάρτησης χωρίς περιορισμούς.

1.5.2 Disciplined Convex Programming

Στην ενότητα αυτή, περιγράφουμε μια τεχνική η οποία χρησιμοποιείται στη διατύπωση προβλημάτων κυρτού προγραμματισμού και "συνοδεύει" γενικούς solvers κυρτών προβλημάτων όπως τα CVX/CVXPY. Η τεχνική αυτή λέγεται Disciplined Convex Programming (DCP), περιγράφεται εκτενώς στο [27] και στηρίζεται στην bottom-up κατασκευή κυρτών συναρτήσεων μέσα από στοιχειώδεις συναρτήσεις ή εκφράσεις των οποίων η καμπυλότητα, η μονοτονία, το πρόσημο κλπ. είναι γνωστά[3] Οι συναρτήσεις αυτές αναφέρονται ως *άτομα* (atoms). Περιγράφουμε ορισμένους κανόνες από το [8] σχετικά με γνωστές κυρτές συναρτήσεις, τις οποίες χρησιμοποιούμε και σε επόμενη ενότητα:

- Αν $f_i = \{f_1, \dots, f_k\}$ μια συλλογή κυρτών συναρτήσεων, τότε η

$$g(\mathbf{x}) = \max\{f_1(\mathbf{x}), \dots f_k(\mathbf{x})\}$$

[3] Στη διεύθυνση http://dcp.stanford.edu, οι συγγραφείς παρέχουν μια αναλυτική περιγραφή του DCP ruleset, ενώ διατίθεται και ένας διαδραστικός αναλυτής συναρτήσεων στο http://dcp.stanford.edu/analyzer.

είναι κι αυτή κυρτή ως προς $\mathbf{x}$.

- Αν $\boldsymbol{X} \in \mathbb{R}^{m\times n}, p \geq 1$, τότε η

$$g(\boldsymbol{X}) = ||\boldsymbol{X}||_p = \left(\sum_{ij} |X_{ij}|^p\right)^{\frac{1}{p}}$$

  δηλαδή η p-νόρμα του πίνακα, είναι κυρτή ως προς $\boldsymbol{X}$. Το ίδιο ισχύει για διάνυσμα $\boldsymbol{x} \in \mathbb{R}^n$. Η συγκεκριμένη περίπτωση προκύπτει εύκολα από τις ιδιότητες της ομογενούς απεικόνισης και της τριγωνικής ανισότητας που ικανοποιούν οι νόρμες, οι οποίες αποτελούν υπογραμμικές (sublinear) συναρτήσεις.

- Η συνάρτηση $g(x) = x^p, p = 2, 4, \ldots, 2k, k \in \mathbb{N}$ είναι κυρτή.

Με βάση τους παραπάνω κανόνες, δείχνουμε την κατασκευή της έκφρασης

$$h(\mathbf{x}) = ||2 \cdot \text{square}(\mathbf{x}) + 3||_1, \quad \mathbf{x} \in \mathbb{R}^n$$

η οποία, στην περίπτωση λ.χ. του χώρου $\mathbb{R}^2$, μπορεί να γραφεί ως

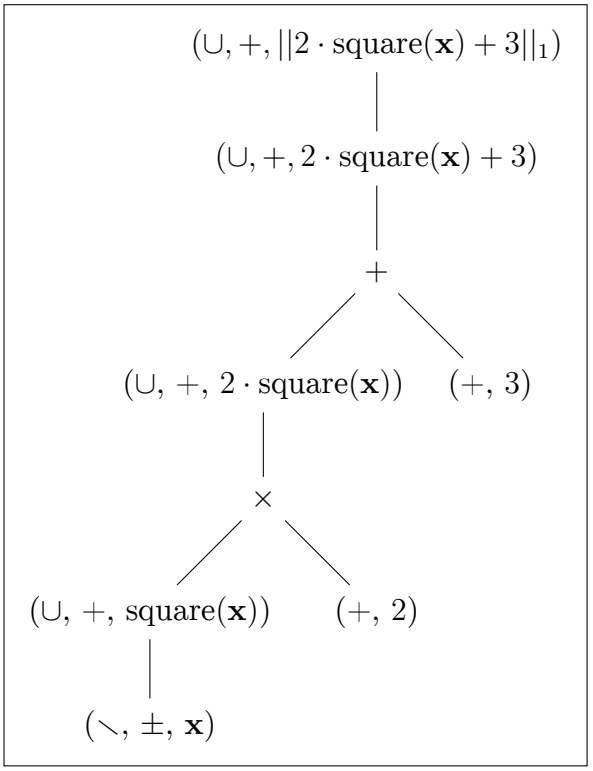
$$h(x, y) = |2x + 3| + |2y + 3|$$

όπου η έκφραση $\text{square}(\mathbf{x})$ συμβολίζει το διάνυσμα που προκύπτει από την ύψωση κάθε στοιχείου του $\mathbf{x}$ στο τετράγωνο. Η κατασκευή φαίνεται στην Εικόνα 1.4. Στην κατασκευή χρησιμοποιούμε τα σύμβολα: $\cup$ για κυρτότητα, $\searrow$ για αφινικότητα, $+$ για θετικό και $\pm$ για άγνωστο πρόσημο. Για παράδειγμα, η μετάβαση $\mathbf{x} \to \text{square}(\mathbf{x})$ χρησιμοποιεί τη γνωστή κυρτότητα του x^p, p άρτιο για να κατασκευάσει την υποέκφραση $\text{square}(\mathbf{x})$, η οποία είναι ένα διάνυσμα με το i-στό στοιχείο του να είναι μια κυρτή, θετική έκφραση x_i^2.

Έτσι, η τεχνική DCP παρέχει ένα αρκετά εκφραστικό σύνολο κανόνων βάσει του οποίου μπορούμε να διατυπώσουμε ένα μεγάλο υποσύνολο των κυρτών προβλημάτων - ουσιαστικά όσα προβλήματα έχουν περιορισμούς και αντικειμενική συνάρτηση που μπορούν να γραφούν ως εκφράσεις με τα atoms που παρέχει η εκάστοτε υλοποίηση του DCP. Ταυτόχρονα, παρέχει μια κατασκευαστική μεθοδολογία για την διατύπωση διαφόρων προβλημάτων βελτιστοποίησης η οποία μπορεί εύκολα να επεκταθεί και για μη-κυρτά, δομημένα προβλήματα, όπως η κατηγορία που περιγράφεται στην Ενότητα 1.5.3.

1.5.3 DC Programming

Τα αρχικά "DC" στο DC programming προέρχονται από τη φράση "Difference of Convex", αφού η κατηγορία προβλημάτων βελτιστοποίησης που περιγράφουν επάγεται από συναρτήσεις κόστους και περιορισμούς που μπορούν να γραφούν ως διαφορές convex συναρτήσεων. Η βελτιστοποίηση DC συναρτήσεων μας ενδιαφέρει ιδιαίτερα, αφού υπεισέρχεται με φυσικό τρόπο στην εκπαίδευση νευρώνων που αποτελούν το αντίστοιχο του γραμμικού perceptron στη $(\max, +)$-άλγεβρα. Διατυπώνουμε εδώ την πρωτοτυπική μορφή ενός DC προβλήματος βελτιστοποίησης.



Σχήμα 1.4: Κανόνες παραγωγής της $h(\mathbf{x}) = ||2 \cdot \text{square}(\mathbf{x}) + 3||_1$

Ορισμός 9 (DC Programming)**.** Έστω $f_i, g_i : \mathbb{R}^n \to \mathbb{R}$ συλλογές κυρτών συναρτήσεων, $h_i : \mathbb{R}^n \to \mathbb{R}$ συλλογή αφφινικών συναρτήσεων. Τότε, το ακόλουθο πρόγραμμα βελτιστοποίησης:

$$\begin{aligned} &\text{Minimize } f_0(\mathbf{x}) - g_0(\mathbf{x}) \\ &\text{s.t. } \begin{cases} f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0 & i = 1, \dots m \\ h_j(\mathbf{x}) = 0 & i = 1, \dots k \end{cases} \end{aligned} \tag{1.19}$$

αποκαλείται πρόβλημα DC (difference of convex) βελτιστοποίησης. Κατ' αντιστοιχία με τον ορισμό 8, το σύνολο εφικτών λύσεών του είναι το σύνολο

$$\begin{aligned} \mathbf{dom}_{\mathcal{F}} = \Bigg\{ \mathbf{x} \in \left(\bigcap_i \mathbf{dom}(f_i) \cap \mathbf{dom}(g_i) \right) \cap \left(\bigcap_j \mathbf{dom}(h_j) \right) : \\ f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0, i \in 1, \dots m, h_j(\mathbf{x}) = 0, j \in 1, \ \dots k \} \end{aligned} \tag{1.20}$$

Είναι εύκολο να παρατηρήσει κανείς ότι ο Ορισμός 9 επιτρέπει τη συμπερίληψη περιορισμών

ισότητας μεταξύ κυρτών συναρτήσεων, αφού μπορούμε να γράφουμε

$$f_i(\mathbf{x}) = g_i(\mathbf{x}) \Longleftrightarrow f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0,\ g_i(\mathbf{x}) - f_i(\mathbf{x}) \leq 0 \tag{1.21}$$

που εμπίπτει στον ορισμό εάν f_i, g_i κυρτές.

Τα DC προγράμματα στη γενική περίπτωση είναι προβλήματα με μη-κυρτές συναρτήσεις κόστους και με μη-κυρτό σύνολο εφικτών λύσεων, τα οποία είναι δύσκολο να βελτιστοποιηθούν συνολικά (globally). Οι περισσότερες μέθοδοι για εύρεση ολικών ελαχίστων μη-κυρτών προβλημάτων στη βιβλιογραφία στηρίζονται κατά βάση σε branch-and-bound αλγορίθμους και η πολυπλοκότητά τους αυξάνεται εκθετικά συναρτήσει του πλήθους μεταβλητών και περιορισμών. Στην πραγματικότητα, ο μη-κυρτός προγραμματισμός αποδεικνύεται τόσο δύσκολος στην πράξη ώστε ακόμα και οι θεωρητικές εγγυήσεις για σύγκλιση σε τοπικά ελάχιστα να μην θεωρούνται δεδομένες. Για παράδειγμα, είναι γνωστό ότι η επαναληπτική μέθοδοδος Newton, η οποία χρησιμοποιείται συχνά σε κυρτά προβλήματα καθώς επιδεικνύει ιδιαίτερα γρήγορη σύγκλιση, συνήθως αποτυγχάνει να εντοπίσει τοπικά ελάχιστα όταν η συνάρτηση κόστους είναι μη-κυρτή, αφού "έλκεται" από σελοειδή σημεία (saddle points), όπου η παράγωγος της συνάρτησης είναι μηδενική χωρίς το σημείο να αποτελεί τοπικό ακρότατο. Ιδιαίτερα στην υψηλοδιάστατη μη-κυρτή βελτιστοποίηση, όπως για παράδειγμα στην εκπαίδευση νευρωνικών δικτύων, τα σελοειδή σημεία συχνά συγκεντρώνονται σε μεγάλες περιοχές που επιβραδύνουν ιδιαίτερα την πρόοδο της επαναληπτικής διαδικασίας, όπως σχολιάζουν οι Pascanu et. al [50]. Η ίδια ομάδα σε επόμενη δημοσίευση [15] προτείνει μια τροποποιημένη μέθοδο πρώτης τάξης για την αποφυγή αυτού του προβλήματος.

Τα DC προβλήματα αποτελούν μια από τις καλύτερα μελετημένες κατηγορίες μη-κυρτών προβλημάτων, για την οποία διαθέτουμε αλγορίθμους που λειτουργούν αποδοτικά αλλά και θεωρητικές εγγυήσεις για τη σύγκλιση αυτών σε τοπικά ελάχιστα. Σε επόμενο κεφάλαιο θα αναφερθούμε αναλυτικά σε μια από αυτές τις μεθόδους η οποία επιλύει μια "χαλάρωση" του αρχικού προβλήματος. Ακόμη, θα εξετάσουμε μια μέθοδο που εφαρμόζεται και σε γενικότερα μη-κυρτά προβλήματα και στηρίζεται στην εξομάλυνση της συνάρτησης κόστους με έναν ισοτροπικό Gaussian πυρήνα και τη σταδιακή εκλέπτυνση αυτής της εξομάλυνσης. Έτσι, η προκύπτουσα συνάρτηση κόστους μετασχηματίζεται σταδιακά από μια (ασυμπτωτικά) κυρτή συνάρτηση στην αρχική συνάρτηση κόστους, δίνοντάς μας μια επιθυμητή ακολουθία τοπικών ελαχίστων. Περιγράφουμε την ιδιότητα της ασυμπτωτικής κυρτότητας που μας ενδιαφέρει στην ορισμό 10.

Ορισμός 10 (Ασυμπτωτική Κυρτότητα)**.** Έστω $f : \mathcal{X} \to \mathbb{R}$ μια μετρήσιμη συνάρτηση, και έστω k_σ ο ισοτροπικός Gaussian πυρήνας με τυπική απόκλιση σ, δηλαδή:

$$k_\sigma(\mathbf{x}) \triangleq \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{||\mathbf{x}||^2}{2\sigma^2}} \tag{1.22}$$

Ορίζουμε με $[f * k_\sigma](\mathbf{x})$ το συνελικτικό ολοκλήρωμα

$$g_\sigma(\mathbf{x}) = \int_{\mathcal{X}} f(\mathbf{x} - \boldsymbol{v}) k_\sigma(\boldsymbol{v}) d\boldsymbol{v} \tag{1.23}$$

Περιοριζόμαστε στις περιπτώσεις όπου η $g(\mathbf{x})$ υπάρχει. Τότε, η f θα αποκαλείται **ασυμπτωτικά κυρτή** με παράμετρο σ_a, εάν υπάρχει $\sigma_a < \infty$ τέτοιο ώστε:

$$g_\sigma(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda g_\sigma(\mathbf{x}) + (1-\lambda) g_\sigma(\mathbf{y}), \ \forall \sigma \geq \sigma_a \tag{1.24}$$

Κεφάλαιο 2

Νευρωνικά Δίκτυα

Με τον όρο (Τεχνητά) Νευρωνικά Δίκτυα αναφερόμαστε σε μια ευρεία γκάμα υπολογιστικών μοντέλων, τα οποία στηρίζονται σε μια ή περισσότερες υπολογιστικές μονάδες, αποκαλούμενες ως *νευρώνες*. Στην περίπτωση που το υπολογιστικό μοντέλο αποτελείται από ένα μοναδικό νευρώνα, είθισται να λέγεται *perceptron*, ενώ στην περίπτωση που αποτελείται από πολλαπλούς νευρώνες, η βιβλιογραφία περιλαμβάνει τους όρους Νευρωνικό Δίκτυο, Perceptron πολλών επιπέδων (MLP - MultiLayer Perceptron), κλπ.

2.1 Το Perceptron του Rosenblatt

Η ορολογία perceptron στη βιβλιογραφία έχει συνδεθεί κυρίως με το μοντέλο περιγράφεται από τον Rosenblatt [56], κατά το οποίο ένας νευρώνας είναι μια υπολογιστική μονάδα η οποία δέχεται ως είσοδο ένα διάνυσμα εισόδου $\mathbf{x} \in \mathbb{R}^n$ και παράγει μια έξοδο $y \in \mathbb{R}$ και αποτελείται από:

- Ένα σύνολο συνάψεων μεταξύ του διανύσματος εισόδου $\mathbf{x}$ και του "σώματος" του νευρώνα, καθεμία από τις οποίες χαρακτηρίζεται από ένα βάρος σύνδεσης $w_i, i \in N$.
- Μια συνάρτηση $f(\cdot)$, η οποία είναι ένας αθροιστής που υπολογίζει την παράσταση

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + b \tag{2.1}$$

  η οποία είναι η διέγερση του νευρώνα από το διάνυσμα $\mathbf{x}$. Η παράμετρος b ονομάζεται πόλωση και στη γενική περίπτωση ισχύει $b \neq 0$.

- Μια *συνάρτηση ενεργοποίησης* $\phi(\cdot)$ η οποία δέχεται ως είσοδο τη διέγερση του νευρώνα και έχει τη μορφή

$$\phi(v) = \begin{cases} 1 & v \geq \theta \\ 0 & v < \theta \end{cases} \tag{2.2}$$

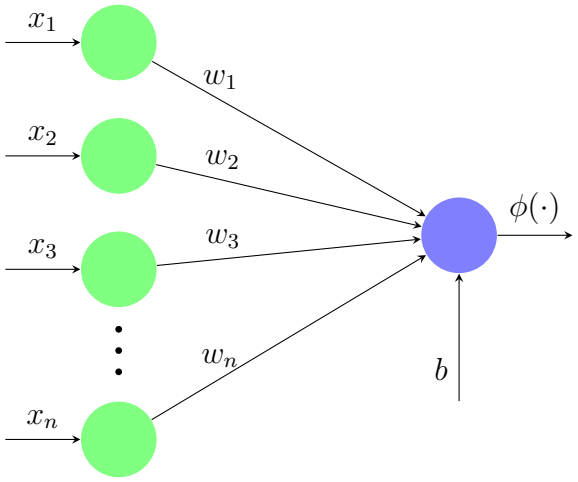
Η εργασία του Rosenblatt βασίστηκε στη προγενέστερη δημοσιευμένη εργασία των Pitts-McCulloch [43], η οποία παρείχε και το κίνητρο για την επιλογή ενός τελεστή κατωφλίωσης για τη συνάρτηση ενεργοποίησης. Τα διανύσματα εισόδου $\mathbf{x}$ συχνά αναφέρονται και ως *πρότυπα*. Σχηματικά, ένας νευρώνας με τη μορφή που περιγράφηκε παραπάνω φαίνεται στην εικόνα 2.1.

2.1.1 Συναρτήσεις ενεργοποίησης

Προφανώς, η συνάρτηση κατωφλίου δεν αποτελεί τη μοναδική επιλογή ως συνάρτηση ενεργοποίησης. Ανάλογα με τον τύπο της εφαρμογής, τη μορφή των προτύπων εισόδου $\mathbf{x}_k$ κλπ. μπορούν να επιλεγούν διάφορετικές συναρτήσεις ενεργοποίησης (βλ. για παράδειγμα [29]). Κάποια γνωστά παραδείγματα ακολουθούν:

- Η συνάρτηση προσήμου

$$\phi(v) = \text{sign}(v) = \begin{cases} 1 & v > 0 \\ 0 & v = 0 \\ -1 & v < 0 \end{cases}$$



Σχήμα 2.1: Linear perceptron

η οποία χρησιμοποιείται ευρέως στην ταξινόμηση προτύπων που έχουν αντληθεί από 2 ξεχωριστές κλάσεις (binary classification).

- Η *σιγμοειδής* συνάρτηση, η οποία δίνεται από τον τύπο

$$\phi(v) = \frac{1}{1 + e^{-v/T}}$$

και είναι επηρεασμένη από την πιθανοκρατική θεώρηση, σύμφωνα με την οποία η κατάσταση ενός νευρώνα είναι

$$s = \begin{cases} 1 & \text{με πιθανότητα } \phi(v) \\ -1 & \text{με πιθανότητα } 1 - \phi(v) \end{cases}$$

Η παράμετρος T είναι μια *ψευδοθερμοκρασία* επηρεασμένη από τη θερμοδυναμική η οποία ελέγχει ουσιαστικά το πόσο "στοχαστική" είναι η συμπεριφορά του νευρώνα, με $T \to 0$ να οδηγεί στο αιτιοκρατικό μοντέλο McCulloch-Pitts.

- Η συνάρτηση υπερβολικής εφαπτομένης με παραμέτρους a, b, δηλαδή

$$\phi(v) = a \tanh(bv)$$

η οποία ουσιαστικά είναι μια σιγμοειδής συνάρτηση με $\phi : \mathbb{R} \to [-1, 1]$, δηλαδή με εικόνα το $[-1, 1]$.

- Η γραμμική συνάρτηση

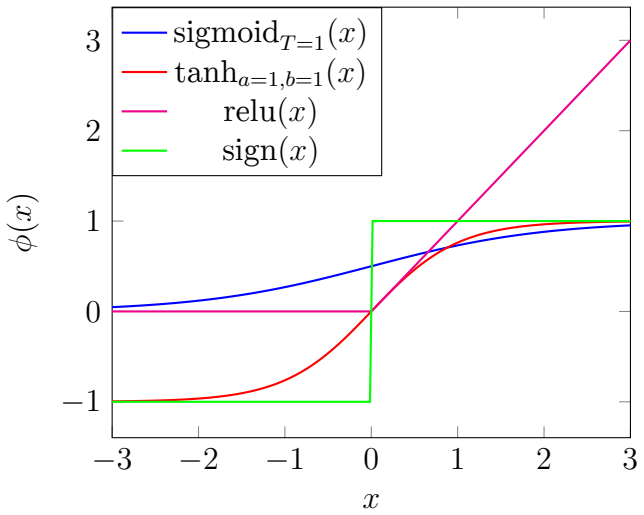
$$\phi(v) = v$$

η οποία ωστόσο σπάνια χρησιμοποιείται στην πράξη.

- Η συνάρτηση ReLU (Rectifier Linear Unit) [47], η οποία ορίζεται ως

$$\phi(v) = \max\{0, v\}$$

και έχει επικρατήσει τον τελευταίο καιρό ως συνάρτηση ενεργοποίησης σε βαθιές αρχιτεκτονικές νευρωνικών δικτύων.



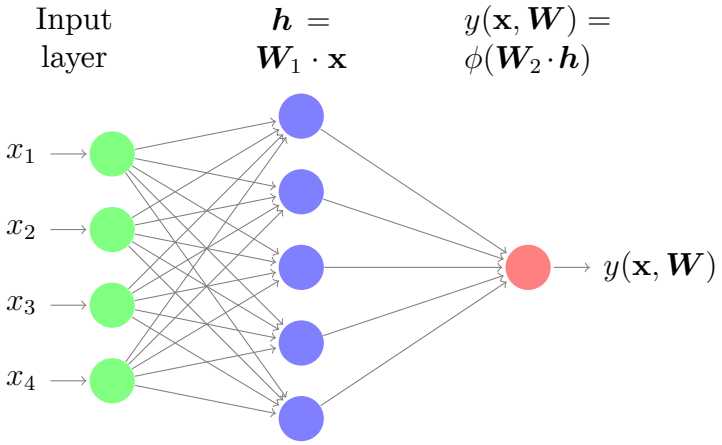
Σχήμα 2.2: Συνήθεις επιλογές συναρτήσεων ενεργοποίησης

Παρατηρείται δηλαδή ότι το βασικό μοντέλο ενός perceptron αποτελείται από ένα γραμμικό συνδυασμό του διανύσματος εισόδου, τον οποίο ακολουθεί μια μη-γραμμικότητα. Κάποιες από τις μη-γραμμικές συναρτήσεις ενεργοποίησης που αναφέρθηκαν απεικονίζονται στο Σχήμα 2.2. Το μη-γραμμικό υπολογιστικό παράδειγμα παρατηρείται έντονα στη βιολογία, δίνοντας το έναυσμα για το μοντέλο που θα παρουσιάσουμε στην επόμενη ενότητα.

2.2 Πολυεπίπεδα νευρωνικά δίκτυα

Στη σημερινή εποχή, η λέξη που πιθανότατα έχει συνδεθεί περισσότερο από κάθε άλλη με την έκφραση "νευρωνικό δίκτυο" είναι η λέξη *εκπαίδευση*. Η δυνατότητα των νευρωνικών δικτύων να προσαρμόζουν τις παραμέτρους τους ώστε να αναγνωρίζουν ή να αναπαριστούν συγκεκριμένα σύνολα δεδομένων γίνεται εφικτή μέσω επαναληπτικών διαδικασιών που ελαχιστοποιούν κάποια συνάρτηση κόστους, η οποία συνήθως εκφράζει πόσο "απέχει" η αναπαράσταση που υπολογίζει το νευρωνικό δίκτυο από την επιθυμητή. Μέχρι τα μέσα της δεκαετίας του 1980, η ιδέα της "εκπαίδευσης" ενός νευρωνικού δικτύου αντιμετωπιζόταν ως πρωτόγνωρη ή και ανέφικτη. Οι Rumelhart, Hinton & Williams [57] πρότειναν το 1986 τη χρήση ενός αλγορίθμου που έμεινε γνωστός ως Backpropagation. Το όνομά του είναι απόρροια του τρόπου που λειτουργεί, υπολογίζοντας την έξοδο του δικτύου για ένα ή περισσότερα δεδομένα εκπαίδευσης και διαδίδοντας "προς τα πίσω", ξεκινώντας από το επίπεδο εξόδου,

το σφάλμα του δικτύου. Χωρίς την ύπαρξη μιας τέτοιας επαναληπτικής διαδικασίας, η κατασκευή πολυεπίπεδων νευρωνικών δικτύων και η προσαρμογή των παραμέτρων θα ήταν πρακτικά αδύνατη. Στην Εικόνα 2.3 φαίνεται ένα πολυεπίπεδο νευρωνικό δίκτυο με πίνακα παραμέτρων $\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_1^T & \boldsymbol{W}_2^T \end{pmatrix}$.



Σχήμα 2.3: Πολυεπίπεδο Perceptron με ένα κρυφό επίπεδο

Ουσιαστικά, εάν $\boldsymbol{W}$ είναι ο πίνακας των παραμέτρων κάθε επιπέδου του δικτύου και $J(\boldsymbol{w})$ μια συνάρτηση που αντιπροσωπεύει το "κόστος" (σχεδόν πάντα μια συνάρτηση που εκτιμά το σφάλμα) του, ο αλγόριθμος backpropagation υπολογίζει την ποσότητα

$$\frac{\partial J}{\partial w_{ij}}, \forall w_{ij} \in \boldsymbol{W}$$

Ο υπολογισμός της ποσότητας είναι εύκολος για τις παραμέτρους του επιπέδου εξόδου, ενώ ο υπολογισμός για τα προηγούμενα επίπεδα γίνεται σταδιακά, χρησιμοποιώντας τον κανόνα της αλυσίδας για τις παραγώγους κάθε επιπέδου. Στην Ενότητα 5.2.1 δίνεται ένα παράδειγμα εφαρμογής του αλγορίθμου backpropagation για την εύρεση των κανόνων ανανέωσης των παραμέτρων ενός μορφολογικού νευρωνικού δικτύου.

2.2.1 Συναρτήσεις Κόστους

Στο σημείο αυτό, μπορούμε να αναφερθούμε σε ορισμένες συχνά χρησιμοποιούμενες συναρτήσεις κόστους. Θεωρούμε σε κάθε περίπτωση ότι $\hat{\boldsymbol{y}}$ είναι ένα n-διάστατο διάνυσμα προβλέψεων/εξόδων και $\boldsymbol{y}$ είναι ένα διάνυσμα ίσης διάστασης που περιέχει τις παρατηρούμενες τιμές/πραγματικές εξόδους.

- Μεσοτετραγωνικό σφάλμα (Mean Squared Error - MSE): Ορίζεται ως

$$\text{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \triangleq \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{2.3}$$

Είναι γνωστό πως το μεσοτετραγωνικό σφάλμα είναι ιδιαίτερα ευαίσθητο σε **outliers**, δηλαδή σε παρατηρήσεις που διαφέρουν (ως προς την τιμή που τους αποδίδει κάποια συνάρτηση κόστους) ριζικά από τον υπόλοιπο πληθυσμό με τον οποίο μοιράζονται κάποιο χαρακτηριστικό. Λόγω αυτής της παθολογικής συμπεριφοράς, συνήθως το MSE συνδυάζεται με κάποια προεπεξεργασία που στοχεύει στην απομάκρυνση των outliers ή το "μαρκάρισμά" τους ώστε να μην λαμβάνονται υπ'όψιν στον ίδιο βαθμό κατά την προσαρμογή του προβλέπτη/ταξινομητή.

- Μέσο απόλυτο σφάλμα (Mean Absolute Error - MAE): ορίζεται ως

$$\mathrm{MAE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \triangleq \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{2.4}$$

Σε σύγκριση με το MSE, η έκφραση (2.4) παρουσιάζει πιο εύρωστη συμπεριφορά απέναντι σε outliers, χωρίς ωστόσο τα προβλήματα να εκλείπουν. Μπορούμε να συνδυάσουμε τις 2 συμπεριφορές των MSE/MAE σε μια υβριδική συνάρτηση κόστους η οποία παρουσιάζεται στη συνέχεια.

- Σφάλμα του Huber [31] (Huber Loss): συνδυάζει τη συμπεριφορά των μεσοτετραγωνικού και του μέσου απόλυτου σφάλματος, αφού για ένα ζεύγος ορίζεται ως

$$\mathrm{Huber}_\delta(\hat{y}_i, y) \triangleq \begin{cases} \frac{1}{2}(\hat{y}_i - y_i)^2 & |\hat{y}_i - y_i| \leq \delta \\ \delta|\hat{y}_i - y_i| - \frac{1}{2}\delta^2 & \text{αλλιώς} \end{cases} \tag{2.5}$$

Η παράμετρος δ ελέγχει το "κατώφλι" στο οποίο το σφάλμα παύει να είναι τετραγωνικό, δρώντας ως παράμετρος ευρωστίας. Συνεπώς, για ένα σύνολο παρατηρήσεων, μπορούμε να γενικεύσουμε το κόστος ως

$$\mathrm{Huber}_\delta(\hat{\boldsymbol{y}}, \boldsymbol{y}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathrm{Huber}_\delta(\hat{y}_i, y_i) \tag{2.6}$$

Η συνάρτηση αυτή χρησιμοποιείται αρκετά συχνά ως εύρωστη εναλλακτική του μεσοτετραγωνικού σφάλματος όταν μπορεί να γίνει η παραδοχή πως εάν η εκτίμηση διαφέρει από την παρατήρηση y_i πάνω από ένα κατώφλι δ, τότε το δείγμα y_i συμπεριφέρεται ως outlier. Ενίοτε το κόστος Huber αναφέρεται και ως soft-threshold least squares στη βιβλιογραφία, αντανακλώντας ακριβώς αυτήν την παρατήρηση.

Οι παραπάνω συναρτήσεις κόστους φαίνονται κατάλληλες για την περίπτωση όπου χρησιμοποιούμε ένα νευρωνικό δίκτυο προκειμένου να ταξινομήσουμε πρότυπα σε 2 διαφορετικές κατηγορίες/κλάσεις $\mathcal{C}_1, \mathcal{C}_0$, αφού (συμβολίζοντας με $\hat{y}(\mathbf{x})$ την απόκριση του δικτύου στο πρότυπο εισόδου $\mathbf{x}$) αρκεί να κάνουμε την παραδοχή ότι

$$\mathbf{x} \in \begin{cases} \mathcal{C}_1 & \hat{y}(\mathbf{x}) \geq 0 \\ \mathcal{C}_0 & \text{αλλιώς} \end{cases} \tag{2.7}$$

Ακόμη, μπορούν να χρησιμοποιηθούν εύκολα σε περίπτωση που χρησιμοποιούμε το νευρωνικό μας για regression (παλινδρόμηση), όπου ζητούμενο δεν είναι να ταξινομήσουμε τα πρότυπα που μας

δίνονται αλλά να προσαρμόσουμε τις παραμέτρους του δικτύου ώστε να "μιμείτα" μια δοθείσα κατανομή δεδομένων όσο καλύτερα γίνεται.

Τι συμβαίνει στην περίπτωση που θέλουμε να ταξινομήσουμε πρότυπα εισόδου σε παραπάνω από μια κατηγορίες $\mathcal{C}_i, i = 1, \dots n$; Πλέον υπάρχουν 2 διαφορετικές προσεγγίσεις:

1. Να θεωρήσουμε τα μοναδιαία διανύσματα

$$\boldsymbol{y}_i^n \triangleq \begin{pmatrix} y_1 = 0 \\ y_2 = 0 \\ \vdots \\ y_i = 1 \\ \vdots \\ y_n = 0 \end{pmatrix} \tag{2.8}$$

και να εκπαιδεύσουμε ένα νευρωνικό δίκτυο με n εξόδους στο σύνολο δεδομένων $\left(\mathbf{x}^k, \boldsymbol{y}_{i_k}^n\right)_k$, όπου

$$\mathbf{x}^k \in \mathcal{C}_a \Rightarrow i_k = a$$

Η μέθοδος αυτή αναφέρεται συχνά ως one-hot encoding. Μετά την επιλογή αυτή το νευρωνικό δίκτυο μπορεί να εκπαιδευθεί χρησιμοποιώντας οποιαδήποτε από τις προαναφερθείσες συναρτήσεις σφάλματος.

2. Να καταφύγουμε στη χρήση συναρτήσεων κόστους που είναι σχεδιασμένες για την περίπτωση ταξινόμησης σε πολλαπλές κατηγορίες. Η πιο διαδεδομένη εξ' αυτών είναι η αρνητική λογαριθμική πιθανοφάνεια, συνδυασμένη με τη συνάρτηση εξόδου softmax [4]. Για ένα νευρωνικό δίκτυο του οποίου το επίπεδο εξόδου έχει M νευρώνες (άρα M εξόδους), συμβολίζοντας την έξοδο του m-οστού νευρώνα με z_m, η softmax συνάρτηση εξόδου ορίζεται ως

$$\sigma(\boldsymbol{z})_m \triangleq \frac{e^{z_m}}{\sum_{m=1}^{M} e^{z_m}} \tag{2.9}$$

Αυτό που επιτυγχάνει ουσιαστικά η συνάρτηση softmax είναι να απεικονίσει ένα M-διάστατο διάνυσμα $\boldsymbol{z} \in \mathbb{R}^M$ στο εύρος $[0, 1]$, με $\sum_{m=1}^{M} \sigma(\boldsymbol{z})_m = 1$. Στο παράδειγμα της ταξινόμησης προτύπων σε πολλές κλάσεις, συνδυάζουμε την softmax με εκπαίδευση πάνω στο σύνολο $(\mathbf{x}^k, y_k)_k$, με $y_k \in 1, \dots M$. Μετά την εφαρμογή softmax στην απόκριση στο πρότυπο $\mathbf{x}^k$, επιλέγεται

$$a = \arg\max_m \sigma(\boldsymbol{z})_m \Rightarrow \mathbf{x}^k \in \mathcal{C}_a$$

Η βασική διαφορά μεταξύ της χρήσης softmax loss / one-hot encoding είναι ο τρόπος που οι επιθυμητές έξοδοι παρουσιάζονται κατά τη διάρκεια της εκπαίδευσης (στη μεν περίπτωση ως βαθμωτές ποσότητες (scalars), στη δε ως διανύσματα).

Στην παρούσα εργασία, το μοντέλο πολυεπίπεδου νευρωνικού δικτύου που θα μας απασχολήσει είναι κατά βάση το πολυεπίπεδο Perceptron, όπως αυτό που φαίνεται στην Εικόνα 2.3. Ωστόσο στη

βιβλιογραφία υπάρχουν πολλά διαφορετικά μοντέλα, όπως για παράδειγμα οι μηχανές Boltzmann (Restricted Boltzmann Machines - RBMs), τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs) κλπ.

Κεφάλαιο 3

Μη-Κυρτή Βελτιστοποίηση

Ο τομέας της μηχανικής μάθησης που καταπιάνεται με την εκπαίδευση νευρωνικών δικτύων και εν γένει την ελαχιστοποίηση συναρτήσεων που εμφανίζονται σε σχετικά προβλήματα δεν περιορίζεται στην κυρτή ανάλυση και βελτιστοποίηση, αλλά πολύ συχνά έρχεται αντιμέτωπη με προβλήματα που εμφανίζουν έντονα μη-κυρτή δομή. Για παράδειγμα, δεν είναι ασυνήθιστο να καλούμαστε να ελαχιστοποιήσουμε μια συνάρτηση πάνω σε μη-κυρτά σύνολα. Με αυτό το δεδομένο και ορμώμενοι από τη διάθεσή μας να ασχοληθούμε με μη-κυρτά μοντέλα στις επόμενες ενότητες, θα επιχειρήσουμε μια διερεύνηση μεθόδων βελτιστοποίησης που ξεφεύγουν από την κλασσική "εργαλειοθήκη" του κυρτού προγραμματισμού. Η βελτιστοποίηση μη-κυρτών συναρτήσεων με εγγυήσεις βελτιστότητας αποτελεί το "ιερό δισκοπότηρο" της μοντέρνας θεωρίας βελτιστοποίησης.

3.1 Βελτιστοποίηση με Εξομάλυνση μέσω Ισοτροπικής Διάχυσης

Θα επικεντρώσουμε την προσοχή μας σε μια μέθοδο η οποία είναι εμπνευσμένη από μεθόδους της Όρασης Υπολογιστών, η οποία χρησιμοποιεί συνέλιξη με έναν ισοτροπικό γκαουσσιανό πυρήνα, ο οποίος αρχικά έχει αρκετά μεγάλη τυπική απόκλιση σ η οποία ελαττώνεται σε κάθε βήμα βελτιστοποίησης. Ένας ισοτροπικός gaussian πυρήνας στις d διαστάσεις με τυπική απόκλιση σ ορίζεται ως

$$k_\sigma : \mathbb{R}^d \to \mathbb{R}, \quad k_\sigma(\mathbf{x}) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^d} \cdot e^{-\frac{||\mathbf{x}||^2}{2\sigma^2}} \tag{3.1}$$

Πρόσφατα, οι Mobahi & Fisher [45] πρότειναν την εξής μέθοδο για βελτιστοποίηση μιας συνάρτησης κόστους $J(\boldsymbol{w})$, η οποία μπορεί να είναι μη-κυρτή ως προς τις παραμέτρους $\boldsymbol{w}$:

Algorithm 1 Optimization By Continuation

Input: $J : \mathcal{W} \to \mathbb{R}$, Sequence $\sigma_0 > \sigma_1 > \cdots > \sigma_n$

$\boldsymbol{w}_0 \leftarrow$ global minimizer of $g(\boldsymbol{w}; \sigma_0)$
for $k = 1$ to n **do**
  $\boldsymbol{w}_k \leftarrow$ local minimizer of $g(\boldsymbol{w}; \sigma_k)$, initialized at $\boldsymbol{w}_{k-1}$
end for
return $\boldsymbol{w}_n$

Στον Αλγόριθμο 1, η συνάρτηση $g(\boldsymbol{w}; \sigma_k)$ ορίζεται ως η συνέλιξη της συνάρτησης κόστους με τον ισοτροπικό gaussian πυρήνα με τυπική απόκλιση σ_k, δηλαδή

$$g(\boldsymbol{w}; \sigma_k) \triangleq [J * k_\sigma](\boldsymbol{w}) = \int_{\mathcal{W}} J(\boldsymbol{w} - \boldsymbol{v}) k_\sigma(\boldsymbol{v}) d\boldsymbol{v} \tag{3.2}$$

όπου $\mathcal{W}$ συμβολίζει το πεδίο ορισμού των παραμέτρων $\boldsymbol{w}$. Ουσιαστικά, ο αλγόριθμος ξεκινάει από μια αρκετά εξομαλυμένη μορφή της συνάρτησης κόστους και σε κάθε βήμα βρίσκει ένα τοπικό ελάχιστο, ξεκινώντας την αναζήτηση από το τοπικό ελάχιστο του προηγούμενου βήματος. Πρόσφατα, ο αλγόριθμος αυτός χρησιμοποιήθηκε για την ταχύτερη εκπαίδευση ενός αναδρομικού νευρωνικού δικτύου (RNN) [44]. Η χρησιμότητά του έγκειται σε 2 πολύ σημαντικές ιδιότητες:

- Η συνέλιξη μιας συνάρτησης με έναν ισοτροπικό Gaussian πυρήνα αρκούντως μεγάλης τυπικής απόκλισης ($\sigma \gg 1$) μπορεί να οδηγήσει σε μια κυρτή συνάρτηση, υπό συγκεκριμένες συνθήκες (ασυμπτωτική κυρτότητα). Εν γένει, οι συγγραφείς στο [45] έχουν δείξει πως ο ισοτροπικός Gaussian πυρήνας προκύπτει ως η καλύτερη αφφινική προσέγγιση της μερικής διαφορικής εξίσωσης Vese [65], η οποία μετατρέπει σταδιακά μια (μη-κυρτή) συνάρτηση f, η οποία έχει τεθεί ως αρχική συνθήκη, στην κυρτή περιβάλλουσά της (convex envelope). Το αποτέλεσμα αυτό ενισχύει τα επιχειρήματα για την καταλληλότητα της χρήσης αυτής της μεθόδου για την επίλυση μη-κυρτών προβλημάτων βελτιστοποίησης.

- Καθώς $\sigma \to 0$, η συνάρτηση $g(\boldsymbol{w}; \sigma)$ προσεγγίζει την αρχική συνάρτηση $J(\boldsymbol{w})$, ενώ η ακολουθία των ελαχίστων $\{\boldsymbol{w}\}_k$ έχει σαν πρώτο όρο το ολικό ελάχιστο της $g(\boldsymbol{w}; \sigma_0)$. Διαισθητικά, αν αυτό το ολικό ελάχιστο βρίσκεται σε μια περιοχή "αρκετά κοντά" στο ολικό ελάχιστο της συνάρτησης J, ο αλγόριθμος μπορεί να μας οδηγήσει σε αυτό, χωρίς ωστόσο η διαίσθηση αυτή να αποτελεί εγγύηση βελτιστότητας στη γενική περίπτωση.

Σε ένα πρόσφατο paper, οι Hazan et. al [30] δίνουν το παράδειγμα μιας συνάρτησης 2 διαστάσεων η οποία διαθέτει 2 ελάχιστα, εκ των οποίων το ένα είναι τοπικό, και αποτελεί στιγμιότυπο συνάρτησης στην οποία οι συνήθεις μέθοδοι βελτιστοποίησης μπορούν να εγκλωβιστούν στο τοπικό ελάχιστο, αναλόγως το σημείο αρχικοποίησης. Η συνάρτηση δίνεται από τον ακόλουθο τύπο:

$$V(\boldsymbol{x}; a, \lambda) = 0.5||\boldsymbol{x}||_2^2 - ae^{-\frac{|\boldsymbol{x}_1 - 1|^2}{2\lambda^2}} \tag{3.3}$$

Για $a > 0.5$, η συνάρτηση διαθέτει ολικό ελάχιστο στο σημείο $\boldsymbol{x} = (1, 0)$, ενώ για $a < 0.5$, στο σημείο $(0, 0)$.

Στο Σχήμα 3.1, βλέπουμε ότι ξεκινώντας από $\sigma = 4$, η εξέλιξη του αλγορίθμου θα "οδηγούσε" μια μέθοδο βελτιστοποίησης στο ολικό ελάχιστο, αφού οι εξομαλυμένες εκδοχές της συνάρτησης αρχίζουν να "μαζεύονται" προς αυτό πριν εμφανιστεί η λεπτομέρεια του τοπικού ελαχίστου στο σημείο $(0, 0)$. Στο Σχήμα 3.2, αυτό φαίνεται πιο καθαρά εξετάζοντας τη μονοδιάστατη εκδοχή της $V(\boldsymbol{x}; a, \lambda)$ για διάφορα σ. Ωστόσο, στη γενική περίπτωση, έχουμε να απαντήσουμε 2 ενδιαφέροντα ερωτήματα:

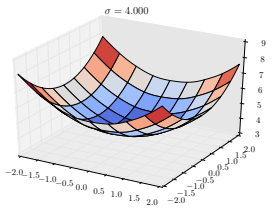
1. Υπό ποιες συνθήκες γνωρίζουμε ότι μια συνάρτηση θα είναι ασυμπτωτικά κυρτή, και, εάν θα είναι, υπό ποιους περιορισμούς η ακολουθία ελαχίστων θα μας οδηγήσει στο ολικό ελάχιστο;

2. Τι συμβαίνει όταν δεν μπορούμε να υπολογίσουμε την έκφραση (3.2) σε κλειστή μορφή;

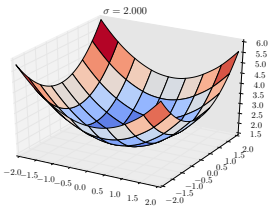
Για το πρώτο ερώτημα, οι Mobahi & Fisher αναφέρουν μια ικανή συνθήκη στο [45], η οποία εγγυάται ότι εάν

$$-\infty < \int_{\mathcal{W}} f(\boldsymbol{w}) dw < 0 \tag{3.4}$$

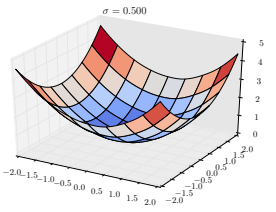
τότε η $g(\mathbf{x}; \sigma)$ θα είναι ασυμπτωτικά κυρτή για αρκετά μεγάλο σ. Αναβάλλουμε προς το παρόν τη συζήτηση περί ασυμπτωτικής κυρτότητας και παρουσιάζουμε μια μέθοδο προσέγγισης της $g(\mathbf{x}; \sigma)$ μέσω δειγματοληψίας Monte Carlo.



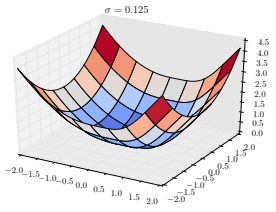
(α′) $g(\boldsymbol{x}; \sigma = 4)$



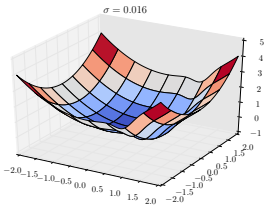
(β′) $g(\boldsymbol{x}; \sigma = 2)$



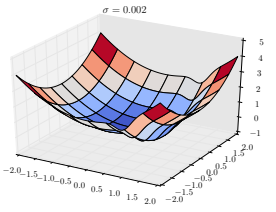
(γ′) $g(\boldsymbol{x}; \sigma = 0.5)$



(δ′) $g(\boldsymbol{x}; \sigma = 0.125)$



(ε′) $g(\boldsymbol{x}; \sigma = 0.016)$



(στ′) $g(\boldsymbol{x}; \sigma = 0.002)$

Σχήμα 3.1: Valley function, $V(\boldsymbol{x}; 0.65, 0.2) * k_\sigma$ για διάφορες τιμές του σ

3.1.1 Εξομάλυνση με μεθόδους Monte Carlo

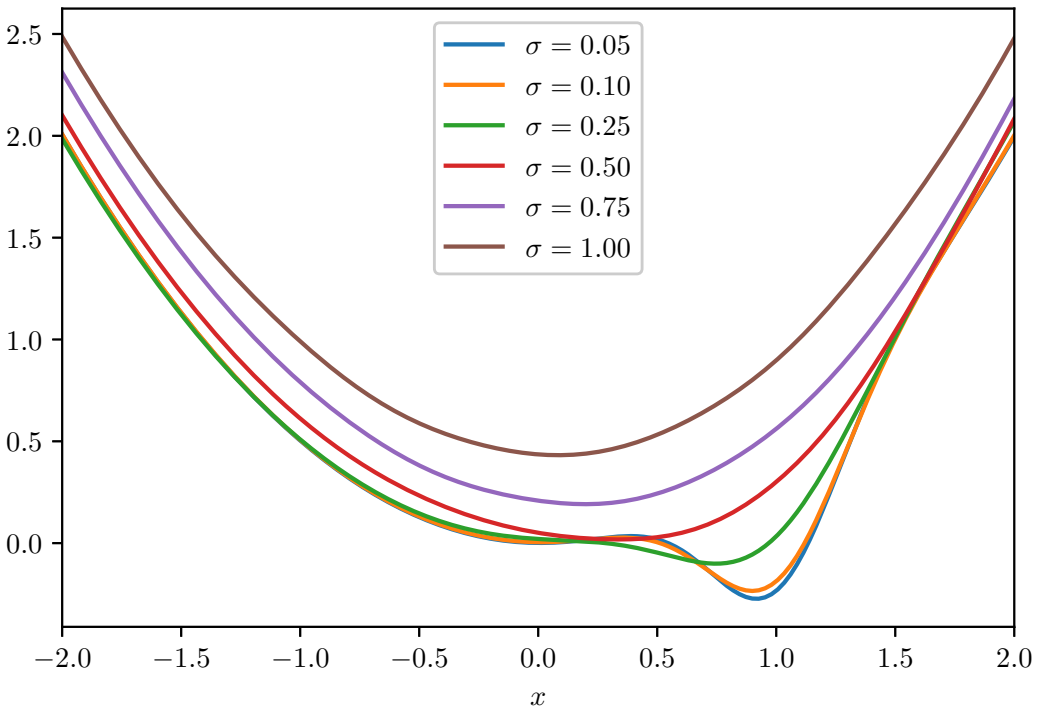
Υποθέτουμε ότι ενδιαφερόμαστε να βελτιστοποιήσουμε μια συνάρτηση $J(\boldsymbol{w}) : \mathcal{W} \to \mathbb{R}$ χρησιμοποιώντας τον Αλγόριθμο 1. Σε όλη τη διάρκεια της ενότητας, υποθέτουμε ότι δε γνωρίζουμε τον ακριβή τύπο της συνάρτησης $J(\boldsymbol{w})$, αλλά έχουμε στη διάθεσή μας ένα "μαντείο" αποτίμησης (**value oracle**), το οποίο με είσοδο $\boldsymbol{w}_k$ μας επιστρέφει την ακριβή τιμή της συνάρτηση $J(\boldsymbol{w}_k)$. Ακόμη, υποθέτουμε ότι δεν έχουμε στη διάθεσή μας πληροφορίες για την παράγωγο ∇J.

Η προσέγγισή μας είναι η εξής: για να υπολογίσουμε την έκφραση $[J * k_\sigma](\boldsymbol{w})$, εκτελούμε δειγματοληψία Monte Carlo ώστε να υπολογίσουμε αριθμητικά το ολοκλήρωμα της Εξ. (3.2). Καθώς έχουμε συνέλιξη με ένα γκαουσσιανό πυρήνα, ο υπολογισμός μέσω Monte Carlo έγκειται στον υπολογισμό της παράστασης

$$\hat{g}(\boldsymbol{w}) \triangleq \frac{1}{n} \sum_{i=1}^{n} J(\boldsymbol{w} - \boldsymbol{v}_i), \quad \boldsymbol{v}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \tag{3.5}$$

Αν υποθέσουμε ότι μπορούμε με αρκετά δείγματα να προσεγγίσουμε την $g(\boldsymbol{w}; \sigma_k)$ με ικανοποιητική ακρίβεια, μπορούμε να δείξουμε το εξής:

Πρόταση 2. Έστω ότι $J : \mathcal{W} \to \mathbb{R}$ συνεχής και ολοκληρώσιμη. Τότε, μέσω δειγματοληψίας MC, μπορούμε να προσεγγίσουμε την παράγωγο ∇g χωρίς πληροφορίες για την παράγωγο ∇J.



Σχήμα 3.2: Valley function, $V(\boldsymbol{x}; 0.65, 0.2) * k_\sigma$ για $\boldsymbol{x} \in \mathbb{R}$

Απόδειξη. Εάν f, h συνεχείς, γνωρίζουμε ότι ισχύει

$$\frac{\partial(f * h)}{\partial x_i} = h * \frac{\partial f}{\partial x_i} \tag{3.6}$$

ενώ ταυτόχρονα βάσει της αντιμεταθετικότητας της συνέλιξης έχουμε

$$\frac{\partial(f * h)}{\partial x_i} = f * \frac{\partial h}{\partial x_i} \tag{3.7}$$

Στην περίπτωσή μας, ισχύει $f = J$, $h = k_\sigma$. Για την περίπτωση της ισοτροπικής γκαουσσιανής συνάρτησης, γνωρίζουμε ότι

$$\nabla k_\sigma(\mathbf{x}) = -\frac{\mathbf{x}}{\sigma^2} k_\sigma(\mathbf{x}) \tag{3.8}$$

Επομένως, ισχύει ότι

$$\begin{aligned} \nabla g &= \nabla \left[\int_{\mathcal{W}} J(\boldsymbol{v}) k_\sigma(\boldsymbol{w} - \boldsymbol{v}) d\boldsymbol{v} \right] \\ &= \int_{\mathcal{W}} J(\boldsymbol{v}) \nabla k_\sigma(\boldsymbol{w} - \boldsymbol{v}) d\boldsymbol{v} \end{aligned}$$

$$= -\frac{1}{\sigma^2}\int_{\mathcal{W}} J(\boldsymbol{v})k_\sigma(\boldsymbol{w}-\boldsymbol{v})\cdot(\boldsymbol{w}-\boldsymbol{v})d\boldsymbol{v} \tag{3.9}$$

Παρατηρώντας ότι η Εξ. (3.9) αποτελεί μια συνέλιξη, μπορούμε να την ξαναγράψουμε ως

$$\nabla g = -\frac{1}{\sigma^2}\int_{\mathcal{W}} J(\boldsymbol{w}-\boldsymbol{v})k_\sigma(\boldsymbol{v})\cdot\boldsymbol{v}d\boldsymbol{v} \tag{3.10}$$

Επομένως με βάση τη μορφή της προσέγγισης (3.5), μπορεί να ξαναγραφεί ως

$$\hat{\nabla} g(\boldsymbol{w}) = -\frac{1}{\sigma^2 n}\sum_{i=1}^{n} J(\boldsymbol{w}-\boldsymbol{v}_i)\cdot\boldsymbol{v}_i, \quad \boldsymbol{v}_i \sim \mathcal{N}(0,\sigma^2\boldsymbol{I}) \tag{3.11}$$

□

Επομένως, εάν "εμπιστευόμαστε" την εκτίμηση Monte Carlo που μπορούμε να πάρουμε με δειγματοληψία, μπορούμε να χρησιμοποιήσουμε τον Αλγόριθμο 1 για να βελτιστοποιήσουμε συναρτήσεις ακόμη και με τον περιορισμό του value oracle.

Στατιστικά μεγέθη της προσέγγισης Monte Carlo

Κατά την προσέγγιση μιας ποσότητας με αριθμητικές μεθόδους όπως η μέθοδος Monte Carlo, είναι χρήσιμο να γνωρίζουμε τα στατιστικά μεγέθη της προσέγγισής μας. Για να μην υπάρξει σύγχυση ως προς τα σύμβολα που χρησιμοποιούμε, σε αυτή την ενότητα θα συμβολίσουμε την παράμετρο κλίμακας της ισοτροπικής γκαουσσιανής με λ, δηλαδή

$$k_\lambda(\mathbf{x}) = \frac{1}{(\sqrt{2\pi}\lambda)^d} e^{-\frac{||\mathbf{x}||^2}{2\lambda^2}} \tag{3.12}$$

Η εκτίμηση (3.5) μπορεί να ιδωθεί ως μια περίπτωση importance sampling, με το ζητούμενο ολοκλήρωμα (3.2) να γράφεται ως

$$\mu(\mathbf{x}) = \mathbb{E}_{k_\lambda}\left(f(\boldsymbol{v})\right) = \mathbb{E}_{k_\lambda}\left(J(\mathbf{x}-\boldsymbol{v})\right) \tag{3.13}$$

όπου προφανώς η τυχαία μεταβλητή είναι το διάνυσμα $\boldsymbol{v}$. Συμβολίζοντας

$$\hat{\mu}^{(\mathbf{x})}_{k_\lambda}(f) \triangleq \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{v}), \quad (\boldsymbol{v}\sim k_\lambda)$$

και με βάση το [48, Theorem 9.1], συνάγουμε ότι:

Πρόταση 3. Έστω οι ποσότητες $\mu(\mathbf{x}), \hat{\mu}_{k_\lambda}$ όπως έχουν οριστεί παραπάνω. Τότε ισχύει:

$$\mathbb{E}[\hat{\mu}^{(\mathbf{x})}_{k_\sigma}] = \mu(\mathbf{x}) \tag{3.14}$$

$$\mathrm{Var}(\hat{\mu}^{(\mathbf{x})}_{k_\lambda}) = \frac{\sigma^2_{k_\lambda}}{n} \tag{3.15}$$

όπου

$$\sigma^2_{k_\lambda} = \int_{\mathcal{V}} f(\boldsymbol{v})^2 k_\lambda(\boldsymbol{v})d\boldsymbol{v} - \mu^2(\mathbf{x}) \tag{3.16}$$

Για την περίπτωση του υπολογισμού της παραγώγου, έχουμε

$$\boldsymbol{\mu}(\mathbf{x}) = \nabla(k_\lambda * f)(\mathbf{x}) = \int_{\mathcal{V}} -\frac{1}{\lambda^2} k_\lambda(\boldsymbol{v}) f(\boldsymbol{v}) \boldsymbol{v} d\boldsymbol{v} = \mathbb{E}_{k_\lambda}\left[-\frac{1}{\lambda^2} f(\boldsymbol{v})\boldsymbol{v}\right] \tag{3.17}$$

$$\hat{\boldsymbol{\mu}}_{k_\lambda}^{(\mathbf{x})} = -\frac{1}{\lambda^2 n}\sum_{i=1}^{n} f(\boldsymbol{v}_i)\cdot \boldsymbol{v}_i \tag{3.18}$$

Για την ποσότητα $\mathbb{E}[\hat{\boldsymbol{\mu}}_{k_\lambda}^{(\mathbf{x})}]$ ισχύει

$$\mathbb{E}\left[\hat{\boldsymbol{\mu}}_{k_\lambda}^{(\mathbf{x})}\right] = \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\frac{-1}{\lambda^2} f(\boldsymbol{v}_i)\boldsymbol{v}_i\right] \tag{3.19}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[-\frac{1}{\lambda^2} f(\boldsymbol{v}_i)\boldsymbol{v}_i\right] \tag{3.20}$$

$$= n\frac{1}{n}\mathbb{E}\left[-\frac{1}{\lambda^2} f(\boldsymbol{v})\boldsymbol{v}\right] = \boldsymbol{\mu}(\mathbf{x}) \tag{3.21}$$

ενώ για τη διασπορά ισχύει

$$\mathrm{Var}(\hat{\boldsymbol{\mu}}) = \mathbb{E}\left[(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\right] \tag{3.22}$$

$$= \mathbb{E}\left[\hat{\boldsymbol{\mu}}\cdot\hat{\boldsymbol{\mu}}^T\right] - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{3.23}$$

Γράφοντας αναλυτικά το γινόμενο έχουμε

$$\mathbb{E}[\hat{\boldsymbol{\mu}}\cdot\hat{\boldsymbol{\mu}}^T] = \mathbb{E}\left[\frac{1}{\lambda^4 n^2}\left(\sum_{i=1}^{n} f(\boldsymbol{v}_i)\boldsymbol{v}_i\right)\left(\sum_{j=1}^{n} f(\boldsymbol{v}_j)\boldsymbol{v}_j\right)^T\right] \tag{3.24}$$

$$= \frac{1}{\lambda^4 n^2}\mathbb{E}\left[\sum_{i,j}^{n} f(\boldsymbol{v}_i) f(\boldsymbol{v}_j)\boldsymbol{v}_i\boldsymbol{v}_j^T\right] \tag{3.25}$$

Στο σημείο αυτό καλούμαστε να πάρουμε 2 περιπτώσεις για τους διάφορους παράγοντες του αθροίσματος:

1. $i \neq j$: σε αυτή την περίπτωση, οι $\boldsymbol{v}_i, \boldsymbol{v}_j$ είναι ανεξάρτητες, ισόνομες μεταβλητές (από υπόθεση) και άρα ισχύει

$$\mathbb{E}\left[f(\boldsymbol{v}_i) f(\boldsymbol{v}_j)\boldsymbol{v}_i\boldsymbol{v}_j^T\right] = \mathbb{E}\left[f(\boldsymbol{v}_i)\boldsymbol{v}_i\right]\cdot\mathbb{E}\left[f(\boldsymbol{v}_j)\boldsymbol{v}_j\right]^T \tag{3.26}$$

$$= \lambda^4(\boldsymbol{\mu}\cdot\boldsymbol{\mu}^T) \tag{3.27}$$

2. $i = j$: στην περίπτωση αυτή δεν έχουμε καταφέρει να καταλήξουμε σε περαιτέρω απλοποίηση, άρα συμβολίζουμε με

$$\alpha_{f,k_\lambda} = \int_{\mathcal{V}} f(\boldsymbol{v}_i)^2 \boldsymbol{v}_i\cdot\boldsymbol{v}_i^T\cdot k_\lambda(\boldsymbol{v}_i) d\boldsymbol{v}_i$$

Το εξωτερικό γινόμενο στην (3.25) έχει συνολικά n^2 όρους εκ των οποίων οι n εμπίπτουν στην περίπτωση $i = j$ και $n^2 - n$ στην περίπτωση $i \neq j$. Επομένως τη γράφουμε ως

$$\mathbb{E}[\hat{\boldsymbol{\mu}} \cdot \hat{\boldsymbol{\mu}}^T] = \frac{n^2 - n}{\lambda^4 n^2} \lambda^4 (\boldsymbol{\mu} \cdot \boldsymbol{\mu}^T) + \frac{n}{\lambda^4 n^2} \lambda^4 \alpha_{f,k_\lambda}(f\boldsymbol{v}) \tag{3.28}$$

$$= \left(1 - \frac{1}{n}\right) \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T + \frac{\alpha_{f,k_\lambda}}{n} \tag{3.29}$$

Επομένως η Εξ. (3.23) προκύπτει ως

$$\mathrm{Var}(\hat{\boldsymbol{\mu}}) = \frac{\sigma_{k_\lambda}^2}{n} \tag{3.30}$$

$$\sigma_{k_\lambda}^2 \triangleq \alpha_{f,k_\lambda} - \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T \tag{3.31}$$

δηλαδή αντίστοιχα με την προσέγγιση στην περίπτωση της f.

3.1.2 Επιλογή δειγμάτων

Ένα ζήτημα που προκύπτει κατά τη δειγματοληψία είναι ο τρόπος επιλογής των δειγμάτων, ώστε η δειγματειλημμένη έκδοση της συνάρτησης να βρίσκεται όσο δυνατόν "κοντινότερα" στην αρχική. Από την άλλη, είναι επίσης ζητούμενο η δειγματοληψία να απαιτεί υπολογιστικό χρόνο τέτοιο ώστε να είναι εφικτό να χρησιμοποιηθεί σε πραγματικές εφαρμογές.

Η πρόταση μας για την δειγματοληψία για τους σκοπούς της εξομάλυνσης είναι η δειγματοληψία να λαμβάνει χώρα μοναδική φορά για κάθε νέα παράμετρο κλίμακας σ_k, για το ζητούμενο πλήθος δειγμάτων N_s. Αναφερόμαστε σε αυτή την πολιτική με τον όρο McRep, όπου χρειαστεί. Με αυτήν την δειγματοληπτική πολιτική αφενός "γλυτώνουμε" την επανάληψη μιας υπολογιστικά απαιτητικής πράξης πολλές φορές σε κάθε βήμα, αφού τα δείγματα $\mathbf{v}_i$ επαναχρησιμοποιούνται για διαφορετικές τιμές του $\mathbf{x}$. Αφετέρου, εγγυόμαστε το εξής:

Πρόταση 4. Έστω $J : \mathcal{W} \to \mathbb{R}$ συνεχής, ολοκληρώσιμη, και Lipschitz-συνεχής με παράμετρο L, δηλαδή $\forall \mathbf{x}, \mathbf{y} \in \mathcal{W}$ ισχύει

$$\frac{||J(\mathbf{y}) - J(\mathbf{x})||}{||\mathbf{y} - \mathbf{x}||} \leq L \tag{3.32}$$

Τότε, χρησιμοποιώντας την πολιτική McRep, η δειγματειλημμένη, εξομαλυμένη συνάρτηση $\hat{g}$ διατηρεί τη συνθήκη Lipschitz.

Απόδειξη. Η απόδειξη είναι συνέπεια της ιδιότητας της τριγωνικής ανισότητας που ικανοποιεί κάθε νόρμα. Γράφουμε

$$\hat{g}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} J(\mathbf{x} - \boldsymbol{v}_i), \quad \boldsymbol{v}_i \sim \mathcal{N}(0, \sigma_k^2 \boldsymbol{I}) \tag{3.33}$$

$$\hat{g}(\mathbf{y}) = \frac{1}{n}\sum_{i=j}^{n} J(\mathbf{y}-\boldsymbol{v}_j), \quad \boldsymbol{v}_j \sim \mathcal{N}(0,\sigma_k^2\boldsymbol{I}) \tag{3.34}$$

Σχηματίζουμε τη διαφορά $\hat{g}(\mathbf{y}) - \hat{g}(\mathbf{x})$, ως

$$\hat{g}(\mathbf{y}) - \hat{g}(\mathbf{x}) = \frac{1}{n}\left(\sum_{i=1}^{n} J(\mathbf{y}-\boldsymbol{v}_i) - \sum_{j=1}^{n} J(\mathbf{x}-\boldsymbol{v}_j)\right) \tag{3.35}$$

$$= \frac{1}{n}\left(\sum_{i=1}^{n} J(\mathbf{y}-\boldsymbol{v}_i) - J(\mathbf{x}-\boldsymbol{v}_i)\right) \quad (i=j \Rightarrow \boldsymbol{v}_i = \boldsymbol{v}_j \text{ από υπόθεση}) \tag{3.36}$$

και για τη νόρμα ισχύει

$$||\hat{g}(\mathbf{y}) - \hat{g}(\mathbf{x})|| = \frac{1}{n}\left|\left|\sum_{i=1}^{n} J(\mathbf{y}-\boldsymbol{v}_i) - J(\mathbf{x}-\boldsymbol{v}_i)\right|\right| \tag{3.37}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n} ||J(\mathbf{y}-\boldsymbol{v}_i) - J(\mathbf{x}-\boldsymbol{v}_i)|| \text{ από ιδιότητες νορμών} \tag{3.38}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n} L||\mathbf{y}-\mathbf{x}|| \text{ από (3.32)} \tag{3.39}$$

$$= \frac{1}{n}\cdot n||\mathbf{y}-\mathbf{x}||L \tag{3.40}$$

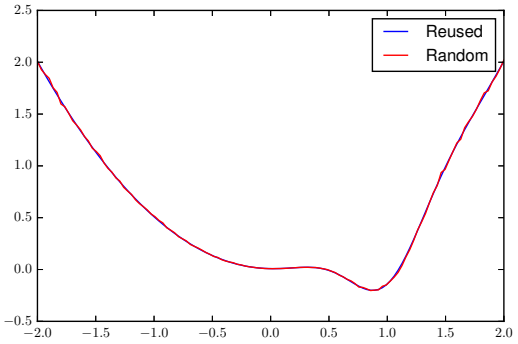
$$\frac{||\hat{g}(\mathbf{y}) - \hat{g}(\mathbf{x})||}{||\mathbf{y}-\mathbf{x}||} \leq L \tag{3.41}$$

που είναι το ζητούμενο. Η συνθήκη $i=j \Rightarrow \boldsymbol{v}_i = \boldsymbol{v}_j$ είναι σημαντική, καθώς διαφορετικά θα είχαμε ένα άθροισμα της μορφής

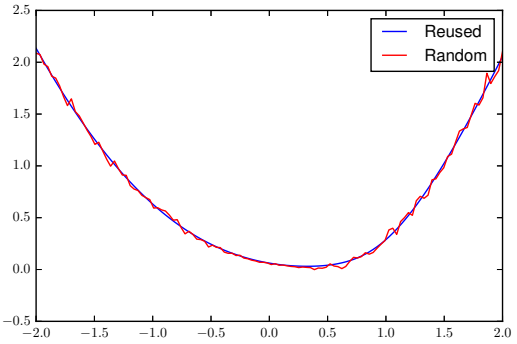
$$\sum_{i=1}^{n} J(\mathbf{y}-\boldsymbol{v}_i) - J(\mathbf{x}-\boldsymbol{v}_i')$$

για το οποίο η διαφορά $\mathbf{y}-\boldsymbol{v}_i-(\mathbf{x}-\boldsymbol{v}_i') = (\mathbf{y}-\mathbf{x}) + \underbrace{(\boldsymbol{v}_i'-\boldsymbol{v}_i)}_{\delta_i}$ δεν μας επιτρέπει να διατηρήσουμε τη συνθήκη Lipschitz, αφού η δ_i είναι τυχαία μεταβλητή. □

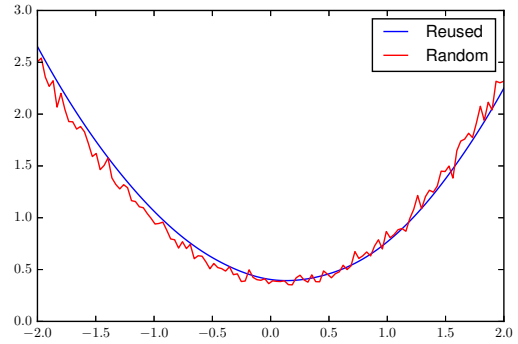
Η Πρόταση 4 εξασφαλίζει ουσιαστικά μια γκάμα συνθηκών που απορρέουν από τη συνθήκη Lipschitz, όπως lipschitz-συνέχεια των παραγώγων, ομαλότητα (smoothness) της συνάρτησης κλπ. Η σημασία της στην πράξη φαίνεται εάν συγκρίνουμε την $[f * k_\sigma](\mathbf{x})$ όταν ακολουθούμε την πολιτική McRep και όταν διαλέγουμε νέα δείγματα για κάθε σημείο $\mathbf{x}$. Στο Σχήμα 3.3 φαίνονται οι διαφορές των 2 πολιτικών, για τη συνάρτηση που ορίζεται στο [30], περιορισμένη στη διάσταση x_1. Είναι προφανές ότι αν επιλέγαμε νέα δείγματα για κάθε σημείο $\mathbf{x}$, ο Αλγόριθμος 1 ενδεχομένως να αποτύγχανε να βρει το ολικό ελάχιστο για παράμετρο σ_0, ειδικά εάν χρησιμοποιούσε μεθόδους πρώτης τάξης όπως η κατάβαση κλίσεων, καθώς η συνάρτηση $\hat{g}$ δεν είναι ομαλή. Ακόμη, μπορούμε με μια αρκετά "ασθενή" υπόθεση να εξάγουμε ένα σχετικό φράγμα για την απόκλιση της τιμής της $[f * k_\sigma](\mathbf{x})$ που εκτιμούμε



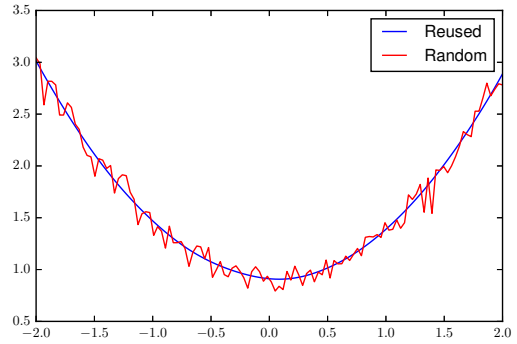
(α′) $\hat{g}, \sigma = 0.02$



(β′) $\hat{g}, \sigma = 0.25$



(γ′) $\hat{g}, \sigma = 1$



(δ′) $\hat{g}, \sigma = 2$

Σχήμα 3.3: Προσεγγίσεις της $g(\mathbf{x}) = [f * k_\sigma](\mathbf{x})$ για διαφορετικές πολιτικές δειγματοληψίας

από την πραγματική τιμή του ολοκληρώματος. Η υπόθεση που θα κάνουμε ουσιαστικά περιορίζει τις αποδεκτές συναρτήσεις κόστους που μπορούμε να λάβουμε υπόψιν για βελτιστοποίηση, αφού απαιτεί οι τιμές της συνάρτησης κόστους να είναι φραγμένες.

Πρόταση 5. Έστω $J : \mathcal{V} \to \mathbb{R}$ φραγμένη έτσι ώστε $\max_{\mathbf{x}}\{J(\mathbf{x})\} - \min_{\mathbf{x}}\{J(\mathbf{x})\} \leq M, \quad \forall \mathbf{x}$ για κάποια κατάλληλη σταθερά M. Τότε, η εκτίμηση

$$\hat{g}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} J(\mathbf{x} - \boldsymbol{v}_i), \quad v_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \tag{3.42}$$

ικανοποιεί τη σχέση

$$\mathbb{P}\left\{|\hat{g} - g| \geq \varepsilon\right\} \leq 2e^{-\frac{2\varepsilon^2 n}{M^2}} \tag{3.43}$$

Απόδειξη. Για την απόδειξη, πρέπει πρώτα να σημειώσουμε τα εξής:

- οι εκτιμήσεις $X_i = J(\mathbf{x} - \boldsymbol{v}_i)$ συνιστούν μια ακολουθία $\{X_n\}_{n\leq 1}$ ανεξάρτητων, φραγμένων μεταβλητών, αφού τα δείγματα $\boldsymbol{v}_i$ λαμβάνονται ως i.i.d δείγματα μιας ισοτροπικής γκαουσσιανής
- $X_i \in [\min X_i, \max X_i] \quad \forall i \in \{1, ...n\}$ με πιθανότητα 1

Ακόμη, ορίζουμε $S_n = \sum_{i=1}^{n} X_i$. Τότε, με βάση το [17, Thm. 2.1], οι εκτιμήσεις X_i ικανοποιούν την ανισότητα Hoeffding, ώστε να ισχύει ($\forall t > 0$):

$$\mathbb{P}\left\{S_n - \mathbb{E}[S_n] \geq t\right\} \leq e^{-\frac{2t^2}{\sum_{i=1}^{n} M^2}} \tag{3.44}$$

$$\mathbb{P}\left\{n\hat{g} - ng \geq t\right\} \leq e^{-\frac{2t^2}{\sum_{i=1}^{n} M^2}} \tag{3.45}$$

Διαιρώντας με n και θέτοντας $\varepsilon = \dfrac{t}{n}$ έχουμε

$$\mathbb{P}\left\{\hat{g} - g \geq \varepsilon\right\} \leq e^{-\frac{2\varepsilon^2 n^2}{nM^2}} \tag{3.46}$$

$$\mathbb{P}\left\{\hat{g} - g \geq \varepsilon\right\} \leq e^{-\frac{2\varepsilon^2 n}{M^2}} \tag{3.47}$$

Ακόμη, ισχύει ακριβώς το ίδιο φράγμα για την πιθανότητα $\mathbb{P}\left\{\hat{g} - g \leq -\varepsilon\right\}$, από το ίδιο θεώρημα. Επειδή η συνθήκη $\mathbb{P}\left\{|\hat{g} - g| \geq \varepsilon\right\}$ γράφεται ως

$$\mathbb{P}\left\{\{\hat{g} - g \geq \varepsilon\} \cup \{\hat{g} - g \leq -\varepsilon\}\right\} = \mathbb{P}\left\{\hat{g} - g \geq \varepsilon\right\} + \mathbb{P}\left\{\hat{g} - g \leq -\varepsilon\right\}$$

δηλαδή ως ένωση 2 ανεξάρτητων ενδεχομένων, τελικά ισχύει ότι

$$\mathbb{P}\left\{|\hat{g} - g| \geq \varepsilon\right\} = 2 \cdot e^{-\frac{2\varepsilon^2 n}{M^2}}$$

που είναι και το ζητούμενο. □

Η Πρόταση 5 ουσιαστικά μας λέει ότι, για δεδομένη ζητούμενη ακρίβεια προσέγγισης ε, η πιθανότητα να την τηρήσουμε "βελτιώνεται" εκθετικά αυξάνοντας τον αριθμό δειγμάτων n.
Εάν γνωρίζουμε ότι η συνάρτηση J ικανοποιεί συνθήκη Lipschitz, μπορούμε να εκμεταλλευτούμε ένα αποτέλεσμα του Wainwright [66], το οποίο μας παρέχει ένα βολικό concentration bound για Lipschitz-συνεχείς συναρτήσεις ισοτροπικών γκαουσσιανών τυχαίων μεταβλητών.

Αρχικά, ορίζουμε τη συνάρτηση

$$f_{\mathbf{x}} : \mathcal{W} \to \mathbb{R}, \quad f_{\mathbf{x}}(\boldsymbol{v}) = J(\boldsymbol{x} - \boldsymbol{v}) \tag{3.48}$$

Είναι προφανές ότι η $f_{\mathbf{x}}$ ικανοποιεί τη συνθήκη Lipschitz, αφού αν η J είναι L-Lipschitz συνεχής τότε ισχύει

$$\frac{||f_{\mathbf{x}}(\boldsymbol{v}_i) - f_{\mathbf{x}}(\boldsymbol{v}_j)||}{||\boldsymbol{v}_i - \boldsymbol{v}_j||} = \frac{||J(\mathbf{x} - \boldsymbol{v}_i) - J(\mathbf{x} - \boldsymbol{v}_j)||}{||\boldsymbol{v}_i - \boldsymbol{v}_j||} \tag{3.49}$$

$$= \frac{||J(\mathbf{x} - \boldsymbol{v}_i) - J(\mathbf{x} - \boldsymbol{v}_j)||}{||(\mathbf{x} - \boldsymbol{v}_i) - (\mathbf{x} - \boldsymbol{v}_j)||} \leq L \tag{3.50}$$

Για τα επόμενα βήματα, χρειαζόμαστε τον ορισμό μιας οικογένειας τυχαίων μεταβλητών, οι οποίες ονομάζονται sub-gaussian.

Ορισμός 11 (Sub-Gaussian T.M.)**.** Έστω $X \in \mathbb{R}$ μια τυχαία μεταβλητή με μέση τιμή $\mathbb{E}[X] = 0$, για την οποία ισχύει

$$\mathbb{E}\left[e^{sX}\right] \leq e^{\frac{\sigma^2 s^2}{2}} \tag{3.51}$$

Τότε, η X λέγεται ότι ανήκει στην υπο-γκαουσσιανή (sub-gaussian) οικογένεια μεταβλητών, με παράμετρο σ.

Θα συμβολίζουμε εφεξής $X \sim \text{SubG}(\sigma)$ όταν η X εμπίπτει στον Ορισμό 11. Διαισθητικά, μια Sub-Gaussian τυχαία μεταβλητή αποτελεί μια "κεντραρισμένη" τυχαία μεταβλητή της οποίας οι ουρές (δηλαδή οι πιθανότητες που αντιστοιχούν στις τιμές που απομακρύνονται από τη μέση τιμή) τείνουν στο 0 με ρυθμό γρηγορότερο από αυτόν μιας Gaussian τ.μ.

Το Θεώρημα που θα χρειαστούμε είναι το ακόλουθο:

Θεώρημα 1 (Theorem 2.4, [66])**.** Έστω $X = (X_1, \ldots, X_d)$ ένα διάνυσμα από ανεξάρτητες, ισόνομες τυχαίες μεταβλητές ($X_i \sim \mathcal{N}(0,1)$), και έστω $f : \mathbb{R}^d \to \mathbb{R}$ μια L-Lipschitz συνάρτηση. Τότε, για τη μεταβλητή

$$D(X) = f(X) - \mathbb{E}[f(X)]$$

ισχύει $D \sim \text{SubG}(L)$ και

$$\mathbb{P}\left[|f(X) - \mathbb{E}[f(x)]| \geq t\right] \leq 2e^{-\frac{2t^2}{\pi^2 L^2}} \quad \forall t \geq 0 \tag{3.52}$$

Μπορούμε να προσαρμόσουμε το Θεώρημα 1 για την περίπτωσή μας, ώστε να λάβουμε ένα ασυμπτωτικό φράγμα συναρτήσει του πλήθους των δειγμάτων n. Το πρώτο βήμα που χρειαζόμαστε είναι να δείξουμε ότι η τυχαία μεταβλητή

$$D_{\mathbf{x}}^{(i)} \triangleq J(\mathbf{x} - \boldsymbol{v}_i) - \mathbb{E}_{k_\lambda}\left[J(\mathbf{x} - \boldsymbol{v}_i)\right], \quad \boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) \tag{3.53}$$

ικανοποιεί $D_{\mathbf{x}}^{(i)} \sim \text{SubG}(h)$, για κάποια παράμετρο h. Είναι εύκολο να γενικεύσουμε το [66, Lemma 2.1], το οποίο έχει διατυπωθεί για ανεξάρτητες τυχαίες μεταβλητές $X, Y \sim \mathcal{N}(0, \boldsymbol{I})$. Έχουμε το εξής:

Λήμμα 1. Έστω $f : \mathbb{R}^d \to \mathbb{R}$ παραγωγίσιμη. Τότε, για κάθε κυρτή συνάρτηση $\phi : \mathbb{R} \to \mathbb{R}$ και ανεξάρτητες τυχαίες μεταβλητές $X, Y \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ ισχύει

$$\mathbb{E}\left[\phi(f(X) - \mathbb{E}[f(X)])\right] \leq \mathbb{E}\left[\phi\left(\frac{\pi}{2}(\nabla f(X))^T \cdot Y\right)\right] \tag{3.54}$$

Το Λήμμα 1 μπορεί να επεκταθεί και για συναρτήσεις f που απλά ικανοποιούν τη συνθήκη Lipschitz, αφού σύμφωνα με το Θεώρημα του Rademacher τέτοιες συναρτήσεις είναι παραγωγίσιμες σχεδόν παντού (ισοδύναμα, το σύνολο των σημείων όπου η f δεν είναι παραγωγίσιμη είναι σύνολο μηδενικού μέτρου). Πλέον είμαστε σε θέση να δείξουμε το νέο φράγμα για L-Lipschitz συναρτήσεις f:

Πρόταση 6. Έστω $f_{\mathbf{x}} : \mathcal{W} \to \mathbb{R}$ η συνάρτηση όπως ορίζεται στην Εξ. 3.48, η οποία είναι L-Lipschitz. Για την εκτίμηση

$$\hat{g}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}}(\boldsymbol{v}_i), \quad \boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) \tag{3.55}$$

όπου η ακολουθία $\{\boldsymbol{v}_i\}_{i=1}^n$ είναι μια ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών, ικανοποιεί την ανισότητα

$$\mathbb{P}\left[|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \geq \epsilon\right] \leq 2e^{-\frac{2\epsilon^2 n}{\pi^2 L^2 \sigma^2}} \tag{3.56}$$

Απόδειξη. Η απόδειξή μας είναι όμοια με την απόδειξη του [66, Theorem 2.4]. Αρχικά, δείχνουμε ότι για την τυχαία μεταβλητή

$$D_{\mathbf{x}}^{(i)} = f_{\mathbf{x}}(\boldsymbol{v}_i) - \mathbb{E}_{\boldsymbol{v}_i \sim k_\lambda}(f_{\mathbf{x}}(\boldsymbol{v}_i)) = f_{\mathbf{x}}(\boldsymbol{v}_i) - g(\mathbf{x})$$

ισχύει $D_{\mathbf{x}}^{(i)} \sim \text{SubG}\left(\frac{\pi L \sigma}{2}\right)$. Χρησιμοποιώντας το Λήμμα 1, έχουμε ότι, για ανεξάρτητες και ισόνομες gaussian τ.μ. X, Y:

$$\begin{aligned}
\mathbb{E}(\exp(\lambda\{f_{\mathbf{x}}(\boldsymbol{v}_i) - g(\mathbf{x})\})) &\leq \mathbb{E}_{X,Y}\left[\exp\left(\frac{\lambda\pi}{2}\sum_{k=1}^{d} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_Y\left[\exp\left(\frac{\lambda\pi}{2}\sum_{k=1}^{d} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right]\right] \text{ (X, Y ανεξάρτητες)} \\
&= \mathbb{E}_X\left[\prod_{k=1}^{d}\mathbb{E}_{Y_k}\left[\exp\left(\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right]\right] (Y \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}))
\end{aligned}$$

Στην τελευταία ισότητα, έχουμε εκμεταλλευτεί το γεγονός ότι μια d-διάστατη ισοτροπική gaussian τ.μ. μπορεί να γραφεί ως γινόμενο d ανεξάρτητων μονοδιάστατων gaussian τ.μ. Πλέον, το κλειδί είναι ο υπολογισμός του ολοκληρώματος

$$\mathbb{E}_{Y_k}\left[\exp\left(\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right]$$

το οποίο είναι το ολοκλήρωμα

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{Y_k^2}{2\sigma^2}} \cdot e^{\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}} dY_k = \frac{1}{\sqrt{2\pi}\sigma} \underbrace{\int_{-\infty}^{\infty} e^{-\frac{Y_k^2}{2\sigma^2} + \frac{\lambda\pi}{2}\frac{\partial f}{\partial x_k}(X) Y_k} dY_k}_{I_k}$$

Αντιπαραβάλλοντας το ολοκλήρωμα I_k με τη γενική μορφή του ολοκληρώματος μιας gaussian συνάρτησης:

$$\int_{-\infty}^{\infty} e^{-ax^2 + bx + c} dx = \sqrt{\frac{\pi}{a}} e^{\frac{b^2}{4a} + c} \tag{3.57}$$

ισχύει ότι

$$I_k = \sqrt{\frac{\pi}{1/(2\sigma^2)}} e^{\frac{\lambda^2\pi^2\left(\frac{\partial f}{\partial x_k}(X)\right)^2}{4\cdot 4/(2\sigma^2)}} = \sqrt{2\pi}\sigma e^{\frac{\lambda^2\pi^2\sigma^2\left(\frac{\partial f}{\partial x_k}\right)^2}{8}} \tag{3.58}$$

και άρα

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{Y_k^2}{2\sigma^2}} \cdot e^{\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}} dY_k = \exp\left(\frac{\lambda^2\pi^2\sigma^2\left(\frac{\partial f}{\partial x_k}\right)^2}{8}\right) \tag{3.59}$$

Με βάση το αποτέλεσμα της (3.59), ξαναγράφουμε την παράσταση

$$\prod_{k=1}^{d} \mathbb{E}_{Y_k}\left[\exp\left(\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right]$$

ως

$$\prod_{k=1}^{d} \mathbb{E}_{Y_k}\left[\exp\left(\frac{\lambda\pi}{2} Y_k \frac{\partial f}{\partial x_k}(X)\right)\right] = \prod_{k=1}^{d} \exp\left(\frac{\lambda^2\pi^2\sigma^2\left(\frac{\partial f(X)}{\partial x_k}\right)^2}{8}\right) \tag{3.60}$$

$$= \exp\left(\frac{\lambda^2\pi^2\sigma^2\sum_{k=1}^{d}\left(\frac{\partial f(X)}{\partial x_k}\right)^2}{8}\right) \tag{3.61}$$

$$= \exp\left(\frac{\lambda^2\pi^2\sigma^2||\nabla f(X)||^2}{8}\right) \tag{3.62}$$

$$\leq \exp\left(\frac{\lambda^2\pi^2\sigma^2 L^2}{8}\right) \tag{3.63}$$

όπου στην τελευταία σχέση εκμεταλλευτήκαμε την L-Lipschitz συμπεριφορά της f. Έτσι, βρήκαμε ότι

$$D_{\mathbf{x}}^{(i)} \sim \textsc{SubG}\left(\frac{\pi L\sigma}{2}\right)$$

Στη συνέχεια, δείχνουμε ότι το άθροισμα

$$\sum_{i=1}^{n} D_{\mathbf{x}}^{(i)} = \sum_{i=1}^{n} f_{\mathbf{x}}(\boldsymbol{v}_i) - \mathbb{E}\left[f_{\mathbf{x}}(\boldsymbol{v}_i)\right]$$

είναι και αυτό sub-gaussian. Επειδή οι τυχαίες μεταβλητές $\boldsymbol{v}_i$ σχηματίζουν μια ακολουθία ανεξάρτητων δοκιμών, έχουμε ότι η ακολουθία $\left\{D_{\mathbf{x}}^{(i)}\right\}_{i=1}^{n}$ είναι και αυτή μια ακολουθία ανεξάρτητων, subgaussian τυχαίων μεταβλητών. Όμως, γνωρίζουμε το εξής:

Λήμμα 2 ([54]). Έστω $\{X_i, X_j\}$ ανεξάρτητες τυχαίες μεταβλητές, με

$$X_i \sim \textsc{SubG}(\sigma_i)$$

$$X_j \sim \text{SubG}(\sigma_j)$$

Τότε, η τυχαία μεταβλητή $X_i + X_j$ είναι sub-gaussian με παράμετρο $\sqrt{\sigma_i^2 + \sigma_j^2}$.

Με ένα απλό επαγωγικό επιχείρημα, το Λήμμα 2 μας δίνει το εξής:

Λήμμα 3. Έστω $\{X_i\}_{i=1}^n$ μια ακολουθία ανεξάρτητων τ.μ., με $X_i \sim \text{SubG}(\sigma)$. Τότε, ισχύει

$$\sum_{i=1}^{n} X_i \sim \text{SubG}(\sqrt{n}\sigma)$$

Έτσι, με τη χρήση του Λήμματος 3 ισχύει

$$\sum_{i=1}^{n} D_{\mathbf{x}}^{(i)} \sim \text{SubG}\left(\frac{\sqrt{n}\pi L\sigma}{2}\right)$$

και άρα με τη χρήση του [66, Prop 2.1], συνάγουμε ότι

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} f_{\mathbf{x}}(\boldsymbol{v}_i) - \mathbb{E}\left[f_{\mathbf{x}}(\boldsymbol{v}_i)\right]\right| \geq t\right] \leq 2e^{-\frac{t^2}{2n\pi^2 L^2\sigma^2/4}} \tag{3.64}$$

$$= 2e^{-\frac{2t^2}{n\pi^2 L^2\sigma^2}} \tag{3.65}$$

Ξαναγράφοντας την αριστερή πλευρά ως

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} f_{\mathbf{x}}(\boldsymbol{v}_i) - \mathbb{E}\left[f_{\mathbf{x}}(\boldsymbol{v}_i)\right]\right| \geq t\right] = \mathbb{P}\left[n \cdot (\hat{g}(\mathbf{x}) - g(\mathbf{x})) \geq t\right] \tag{3.66}$$

$$= \mathbb{P}\left[|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \geq (t/n)\right] \tag{3.67}$$

$$= \mathbb{P}\left[|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \geq \epsilon\right] \quad \left(\epsilon = \frac{t}{n}\right) \tag{3.68}$$

λαμβάνοντας υπόψιν ότι

$$\epsilon = \frac{t}{n} \Rightarrow t^2 = \epsilon^2 n^2 \tag{3.69}$$

καταλήγουμε στο

$$\mathbb{P}\left[|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \geq \epsilon\right] \leq 2e^{-\frac{2\epsilon^2 n}{\pi^2 L^2\sigma^2}} \tag{3.70}$$

που είναι και το ζητούμενο. □

Κλείνουμε την ενότητα διατυπώνοντας τα φράγματα που λάβαμε για το n σε κάθε περίπτωση.

Πρόταση 7. Έστω $J : \mathcal{W} \to \mathbb{R}$ συνεχής συνάρτηση, k_σ ο ισοτροπικός gaussian πυρήνας με διασπορά σ^2. Έστω η προσέγγιση της συνέλιξης $g(\mathbf{x}) = [J * k_\sigma](\mathbf{x})$:

$$\hat{g}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} J(\mathbf{x} - \boldsymbol{v}_i), \quad \boldsymbol{v}_i \sim k_\sigma \tag{3.71}$$

Τότε, συμβολίζοντας με ϵ τη μέγιστη επιθυμητή απόκλιση της εκτίμησης $\hat{g}$ από την πραγματική g, διακρίνουμε 2 περιπτώσεις:

- Αν η J είναι φραγμένη με $\max_{\mathbf{x}}\{J(\mathbf{x})\} - \min_{\mathbf{x}}\{J(\mathbf{x})\} \leq M$, $\forall \mathbf{x}$, τότε επιλέγοντας

  $$n \geq \frac{M^2}{2\epsilon^2} \log\left(\frac{2}{\delta}\right) \tag{3.72}$$

  ισχύει ότι $|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \leq \epsilon$ με πιθανότητα τουλάχιστον $1 - \delta$.

- Αν η J είναι L-Lipschitz συνεχής, τότε επιλέγοντας

  $$n \geq \frac{\pi^2 L^2 \sigma^2}{2\epsilon^2} \log\left(\frac{2}{\delta}\right) \tag{3.73}$$

  ισχύει ότι $|\hat{g}(\mathbf{x}) - g(\mathbf{x})| \leq \epsilon$ με πιθανότητα τουλάχιστον $1 - \delta$.

Απόδειξη. Στα φράγματα των Προτάσεων 5, 6, απαιτούμε το δεξί μέλος να είναι $\leq \delta$, και έπειτα το ζητούμενο προκύπτει εύκολα με στοιχειώδεις υπολογισμούς. □

Εάν εκ πρώτης όψεως το πρώτο από τα φράγματα της Πρότασης 7 φαίνεται πιο ελκυστικό, στην πραγματικότητα ο παράγοντας M^2 του αριθμητή μπορεί να καθιστά απαγορευτικό το n για συναρτήσεις με μεγάλο εύρος τιμών. Αντίθετα, στη δεύτερη εκδοχή είναι πιθανόν να δώσει πιο ικανοποιητικό φράγμα για το n, ειδικά όταν η σταθερά Lipschitz είναι σχετικά μικρή. Η εμφάνιση της διασποράς σ^2 στον αριθμητή πρακτικά συνεπάγεται ότι μπορούμε να διατηρούμε σταθερό το λόγο σ^2/ϵ^2 ώστε για μικρές κλίμακες σ, να έχουμε και μεγαλύτερη ακρίβεια προσέγγισης ϵ χωρίς το n να αυξομειώνεται. Τέλος, ο παρατηρητικός αναγνώστης θα προσέξει ότι τα φράγματα που έχουμε εξάγει δεν εξαρτώνται από τη διάσταση d του πεδίου ορισμού / χώρου χαρακτηριστικών $\mathcal{W}$. Το αποτέλεσμα αυτό είναι σε συμφωνία με την ιδιότητα της ανεξαρτησίας ανά διάσταση του ισοτροπικού gaussian πυρήνα. Ωστόσο, αύξηση της διάστασης d συχνά συνεπάγεται και την παρουσία μεγαλύτερων σταθερών Lipschitz.

3.1.3 Αριθμητική Προσέγγιση της Ισοτροπικής Διάχυσης

Στην ενότητα αυτή, δίνουμε έναν αλγόριθμο βελτιστοποίησης συναρτήσεων που χρησιμοποιεί τις παρατηρήσεις της προηγούμενης ενότητας ώστε να λάβει μια αριθμητική προσέγγιση της ισοτροπικής διάχυσης. Αρχικά, περιγράφουμε τον αλγόριθμο προσέγγισης του κόστους που μπορεί να χρησιμοποιηθεί ως διαδικασία (procedure) στον αλγόριθμο βελτιστοποίησης μέσω ισοτροπικής διάχυσης.

Ο αλγόριθμος αυτός δέχεται ως είσοδο ένα value oracle της συνάρτησης κόστους J, μια παράμετρο Lipschitz L, την κλίμακα εξομάλυνσης σ, τις παραμέτρους δ, ϵ του διαστήματος εμπιστοσύνης της προσέγγισης καθώς και το σημείο $\mathbf{x}$ στο οποίο θέλουμε να προσεγγίσουμε την εξομαλυμένη συνάρτηση κόστους, $g(\mathbf{x}) = [J * k_\sigma](\mathbf{x})$.

Algorithm 2 Diffused Cost Approximation

Input: $J : \mathcal{W} \to \mathbb{R}, \sigma, L, \delta, \epsilon, \mathbf{x}$

$n \leftarrow \dfrac{\pi^2 L^2 \sigma^2}{2\epsilon^2} \log\left(\dfrac{2}{\delta}\right)$

$\hat{g}(\mathbf{x}) \leftarrow \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} J(\mathbf{x} - \boldsymbol{v}_i), \quad \boldsymbol{v}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$

return $\hat{g}(\mathbf{x})$ s.t. $|(\hat{g} - g)(\mathbf{x})| \leq \epsilon$ with probability $p \geq 1 - \delta$

Όταν η διάσταση d των δεδομένων είναι πολύ μεγάλη (και επομένως η σταθερά Lipschitz L μπορεί να παίρνει αρκετά μεγάλες τιμές), μπορούμε να εκμεταλλευτούμε ένα πρόσφατο αποτέλεσμα του Calafiore [10], ο οποίος προτείνει μια παράλληλη διαδικασία βελτιστοποίησης που περιλαμβάνει 2 φάσεις. Η πρώτη φάση πραγματοποιεί k παράλληλες βελτιστοποιήσεις σε k ξένα ανά 2 σύνολα μεταβλητών, ενώ η δεύτερη συντονίζει τις αλλαγές τιμών της πρώτης φάσης σε ένα συνολικό update. Στο [10], αποδεικνύεται η σύγκλιση στο ολικό ελάχιστο για κυρτές παραγωγίσιμες συναρτήσεις (με την προαιρετική ύπαρξη ενός μη-παραγωγίσιμου, αλλά διαχωρίσιμου όρου). Έτσι, έχουμε αυτόματα μια εγγύηση σύγκλισης για τις ασυμπτωτικά κυρτές συναρτήσεις, για $\sigma \geq \sigma_a$, χρησιμοποιώντας τον αλγόριθμο 1.

Επιβεβαιώνουμε και πειραματικά την επάρκεια των φραγμάτων της Πρότασης 7. Στο [44, Table 1], δίνονται σε κλειστή μορφή οι τύποι ορισμένων συναρτήσεων εξομαλυμένων από ισοτροπικούς gaussian πυρήνες. Για λόγους πληρότητας, η λίστα παρατίθεται αυτούσια στον Πίνακα 3.1. Γνωρίζουμε ότι

Πίνακας 3.1: Συναρτήσεις με τις εξομαλυμένες εκδοχές τους

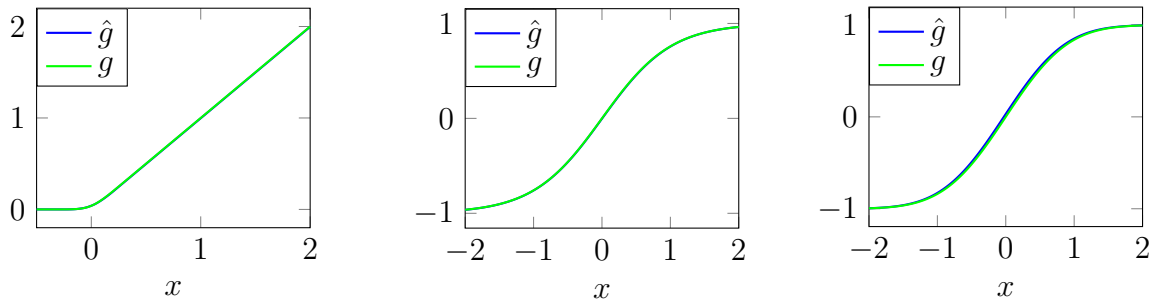
Όνομα	**Αρχική**	**Εξομαλυμένη**
Sign	$\operatorname{sign}(x)$	$\operatorname{erf}\left(\dfrac{x}{\sqrt{2}\sigma}\right)$
Error	$\operatorname{erf}(ax)$	$\operatorname{erf}\left(\dfrac{ax}{\sqrt{1+2(a\sigma)^2}}\right)$
Tanh	$\tanh(x)$	$\tanh\left(\dfrac{x}{\sqrt{1+\frac{\pi}{2}\sigma^2}}\right)$
ReLU	$\max(0, x)$	$\dfrac{\sigma}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} + \dfrac{1}{2}x\left(1 + \operatorname{erf}\left(\dfrac{x}{\sqrt{2}\sigma}\right)\right)$

οι Erf, ReLU και Tanh είναι Lipschitz συνεχείς με παραμέτρους $\dfrac{2}{\sqrt{\pi}}, 1, 1$ αντίστοιχα. Εκμεταλλευόμενοι αυτό το γεγονός, τις προσεγγίζουμε με τη χρήση των αποτελεσμάτων της προηγούμενης

Πίνακας 3.2: Πλήθος δειγμάτων, $L = 1$, $\delta = 0.001$

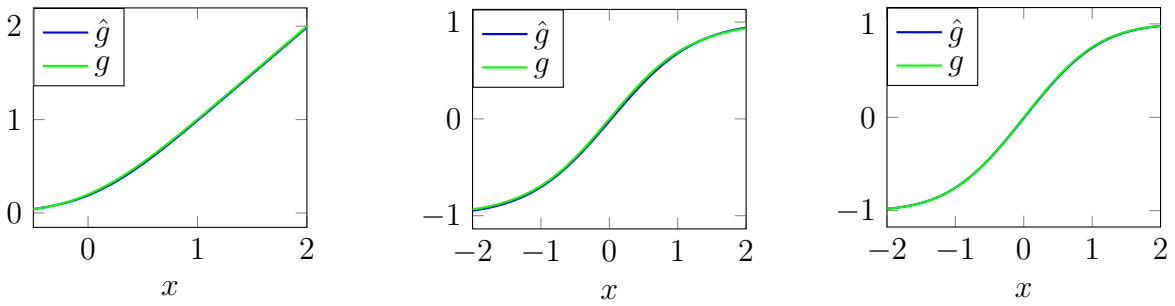
σ	ϵ	n	$\max_{\mathbf{x}}\{\hat{g}(\mathbf{x}) - g(\mathbf{x})\}$
0.1	0.01	3801	0.0001
	0.025	609	0.002
	0.05	153	0.004
	0.1	39	0.032
0.5	0.01	95012	0.002
	0.025	15202	0.004
	0.05	3801	0.006
	0.1	951	0.015

πρότασης για συναρτήσεις στη 1 διάσταση. Ενδεικτικές απεικονίσεις των προσεγγίσεων φαίνονται στις Εικόνες 3.4, 3.5, 3.6.



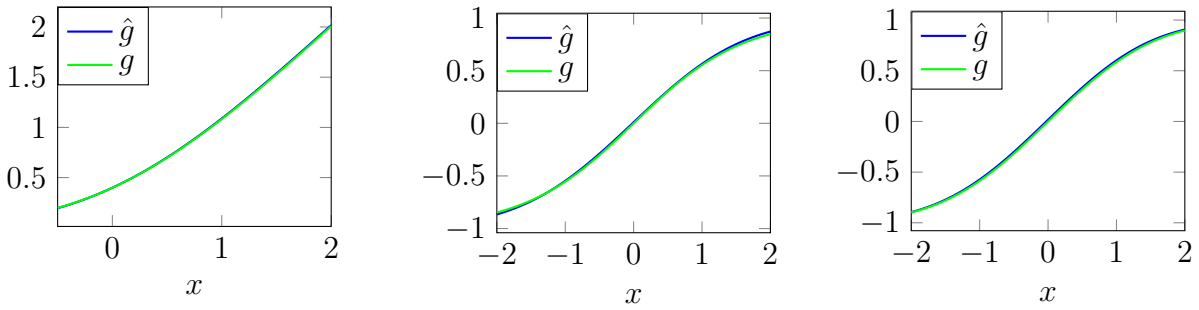
(α′) relu(x) **(β′)** tanh(x) **(γ′)** erf(x)

Σχήμα 3.4: Προσέγγιση $g_\sigma(x), \sigma = 0.1, \epsilon = 0.1, \delta = 0.001$



(α′) relu(x) **(β′)** tanh(x) **(γ′)** erf(x)

Σχήμα 3.5: Προσέγγιση $g_\sigma(x), \sigma = 0.5, \epsilon = 0.1, \delta = 0.001$



(α′) relu(x) **(β′)** tanh(x) **(γ′)** erf(x)

Σχήμα 3.6: Προσέγγιση $g_\sigma(x), \sigma = 1, \epsilon = 0.1, \delta = 0.001$

Στον Πίνακα 3.2 φαίνονται ενδεικτικές τιμές του πλήθους δειγμάτων n για κάθε συνάρτηση. Όπως φαίνεται, η αύξηση της κλίμακας σ επιφέρει μεγάλη αύξηση στο πλήθος των σημείων n που είναι απαραίτητα για την επίτευξη της επιθυμητής ακρίβειας, κάτι που είναι αναμενόμενο αφού αυξάνεται το πλήθος των γειτνιαζόντων σημείων που έχουν "σημαντική" συνεισφορά στην εξομαλυμένη τιμή $g(\mathbf{x})$.

3.2 Η Convex-Concave διαδικασία

Η Convex-Concave διαδικασία είναι μια μέθοδος βελτιστοποίησης η οποία προτάθηκε για μη-κυρτά προβλήματα τα οποία ακολουθούν τη μορφή Difference of Convex (DC), όπως αυτή ορίστηκε στο Κεφ. 1.5.3. Προτάθηκε από τους Yuille & Rangarajan [69] το 2003 και στηρίζεται στην διαδοχική προσέγγιση των συναρτήσεων που εμφανίζονται στο κόστος και τους περιορισμούς του προβλήματος με το ανάπτυγμα Taylor 1ης τάξης, του οποίου η μορφή υπενθυμίζεται:

Ορισμός 12 (Ανάπτυγμα Taylor πολλών μεταβλητών)**.** Έστω $f : \mathbb{R}^d \to \mathbb{R}$ μια συνάρτηση απείρως παραγωγίσιμη γύρω από το σημείο $\mathbf{x} = \boldsymbol{a}$. Το ανάπτυγμα Taylor της f στο σημείο $\boldsymbol{a}$ ορίζεται ως

$$T_{\boldsymbol{a}}(\mathbf{x}) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} \frac{(x_1 - a_1)^{n_1} \cdots (x_d - a_d)^{n_d}}{n_1! \cdots n_d!} \left(\frac{\partial^{(n_1+\cdots+n_d)} f}{\partial x_1^{n_1} \dots x_d^{n_d}} \right) (a_1, \dots a_d) \tag{3.74}$$

Για παράδειγμα, το ανάπτυγμα Taylor 2ης τάξης μιας συνάρτησης f που ικανοποιεί τις προϋποθέσεις του Ορισμού 12, γράφεται ως

$$f_{\boldsymbol{a}}(\mathbf{x}) = f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})^T (\mathbf{x} - \boldsymbol{a}) + \frac{1}{2} (\mathbf{x} - \boldsymbol{a})^T H_f(\mathbf{a}) (\mathbf{x} - \boldsymbol{a}) \tag{3.75}$$

όπου H_f ο Hessian πίνακας της f. Το ανάπτυγμα Taylor 1ης τάξης θα γραφόταν ως

$$f_{\boldsymbol{a}}(\mathbf{x}) = f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})^T (\mathbf{x} - \boldsymbol{a}) \tag{3.76}$$

από τη μορφή του οποίου είναι εύκολο να συνάγουμε ότι η προσέγγιση μιας συνάρτησης με 1οτάξιο ανάπτυγμα Taylor είναι η εφαπτομένη της στο σημείο $\boldsymbol{a}$. Αυτή ακριβώς η προσέγγιση είναι που χρησιμοποιείται στην Convex-Concave Procedure: συγκεκριμένα, για κάθε παραγωγίσιμη κυρτή συνάρτηση g, γνωρίζουμε ότι

$$\begin{aligned} g(\mathbf{y}) &\geq g(\mathbf{x}) + \nabla g(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \Rightarrow \\ -g(\mathbf{y}) &\leq - \left[g(\mathbf{x}) + \nabla g(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \right] \end{aligned} \tag{3.77}$$

Παρατηρώντας την (3.77), διαπιστώνουμε πως το δεξί της μέλος είναι σε κάθε περίπτωση μια affine συνάρτηση του $\mathbf{y}$. Επομένως, μια προσέγγιση της DC συνάρτησης

$$d(\mathbf{x}) = f(\mathbf{x}) - g(\mathbf{x})$$

γύρω από τυχόν σημείο $\mathbf{x}^k$, θα μπορούσε να είναι η

$$\tilde{d}(\mathbf{x};\mathbf{x}^k) \triangleq f(\mathbf{x}) - g\left(\mathbf{x}^k\right) - \nabla g\left(\mathbf{x}^k\right)^T (\mathbf{x}-\mathbf{x}^k) \tag{3.78}$$

Η προσέγγιση που χρησιμοποιεί η Convex-Concave Procedure δεν είναι η μοναδική δυνατότητα για τέτοια προβλήματα (π.χ. θα μπορούσε κάποιος να εκμεταλλευθεί την ανισότητα Fenchel-Young [7, Prop. 3.3.4]), ωστόσο είναι ένας από τους πιο πρακτικούς τρόπους να απλοποιήσουμε ένα DC πρόβλημα.

3.2.1 Convex-Concave Procedure χωρίς περιορισμούς

Αν θεωρήσουμε το $\mathbf{x}^k$ παράμετρο, η (3.78) είναι προφανώς μια κυρτή συνάρτηση του $\mathbf{x}$, αφού είναι άθροισμα ενός κυρτού και ενός affine όρου. Αυτός είναι ο τρόπος με τον οποίο οι Yuille & Rangarajan επιλύουν το πρόβλημα που παρουσιάζουν τα DC προβλήματα: η κοίλη συνάρτηση $-g(\mathbf{x})$ που εμφανίζεται αντικαθιστάται από μια affine προσέγγισή της, χρησιμοποιώντας το 1ο-τάξιο ανάπτυγμα Taylor. Ακόμη, χρησιμοποιώντας τις ιδιότητες της κυρτότητας της g, γνωρίζουμε ότι

$$\begin{aligned} -g(\mathbf{x}) &\leq -g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) \\ f(\mathbf{x}) - g(\mathbf{x}) &\leq f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) \Rightarrow \\ d(\mathbf{x}) &\leq \tilde{d}(\mathbf{x};\mathbf{x}^k) \quad \forall \mathbf{x}^k \end{aligned} \tag{3.79}$$

Η (3.79) είναι ιδιαίτερα σημαντική, καθώς από αυτήν συνάγουμε ότι η προσέγγιση κατά Taylor της συνάρτησης προς βελτιστοποίηση είναι υπερεκτιμητής της - άρα ισχύουν τα εξής:

- Αν έχουμε στη διάθεσή μας μια τοπικά βέλτιστη λύση $\mathbf{z}^*$ της προσέγγισης γύρω από το σημείο $\mathbf{x}^k$, τότε το ολικό ελάχιστο d_{opt} της d ικανοποιεί $d(\mathbf{x}_{\text{opt}}) \leq \tilde{d}(\mathbf{z}^*;\mathbf{x}^k)$.
- Αν θεωρήσουμε

  $$\mathcal{X}_{\text{opt}} = \left\{\mathbf{x} \in \mathbf{dom}(d) : \exists \epsilon > 0 \rightarrow \tilde{d}(\mathbf{x};\mathbf{x}^k) \leq \tilde{d}(\mathbf{x}';\mathbf{x}^k), \forall \mathbf{x}' \in \mathcal{B}_\epsilon\right\}$$

  δηλαδή αν $\mathcal{X}_{\text{opt}}$ το σύνολο τοπικών ελαχίστων της $\tilde{d}(\mathbf{x};\mathbf{x}^k)$, τότε προφανώς ισχύει

  $$\mathbf{x}_{\text{opt}} \in \mathcal{X}_{\text{opt}}$$

Με βάση τα παραπάνω, μπορούμε να διατυπώσουμε την Convex-Concave Procedure ως μια ευριστική για την ελαχιστοποίηση DC συναρτήσεων χωρίς περιορισμούς (unconstrained minimization), η οποία επιλέγει ένα αρχικό σημείο $\mathbf{x}^0$ και σε κάθε βήμα υπολογίζει το ανάπτυγμα Taylor της $g(\mathbf{x})$ γύρω από το σημείο $\mathbf{x}^k$, επιλύοντας ένα κυρτό υποπρόβλημα για την εύρεση του σημείου $\mathbf{x}^{k+1}$. Ακριβώς επειδή ο Αλγόριθμος 3 χρησιμοποιεί την προσέγγιση $\tilde{d}$ ως ευριστική, το σημείο σύγκλισής του εξαρτάται από την επιλογή του αρχικού σημείου $\mathbf{x}^0$, ιδιαίτερα όταν η d αναμένεται να έχει πολλά τοπικά ελάχιστα. Οι Lipp & Boyd [37] προτείνουν την επανάληψη της εφαρμογής του με διαφορετικά σημεία αρχικοποίησης $\mathbf{x}^0$.

Algorithm 3 Convex-Concave Procedure for Unconstrained Minimization

function UNCONSTRAINEDCCP($d(\mathbf{x}) = f(\mathbf{x}) - g(\mathbf{x}), \mathbf{x}^0$, maximum iterations K)
  $k \leftarrow 0$
  while $k \leq K$ and stopping criterion not satisfied **do**
    $\tilde{d}(\mathbf{x}; \mathbf{x}_k) \leftarrow f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k)$
    Set x_{k+1} to the solution of

$$\begin{aligned} &\text{Minimize } \tilde{d}(\mathbf{x}; \mathbf{x}_k) \\ &\text{s.t. } \mathbf{x} \in \mathbf{dom}(d) \end{aligned}$$

    $k \leftarrow k + 1$
  end while
end function

3.2.2 Convex-Concave Procedure με DC περιορισμούς

Η διαδικασία που παρουσιάζεται στον Αλγόριθμο 3 της προηγούμενης ενότητας αναφερόταν μονάχα σε DC προβλήματα χωρίς περιορισμούς. Σε περίπτωση που οι περιορισμοί ενός DC προβλήματος έχουν τη μορφή

$$f_i(\mathbf{x}) \leq 0, h_i(\mathbf{x}) = 0, \quad f \text{ convex }, h \text{ affine}$$

ο Αλγόριθμος 3 ουσιαστικά μπορεί να παραμείνει ίδιος, ενσωματώνοντας τη συνθήκη για το feasibility των περιορισμών. Ωστόσο, είναι πολύ εύκολο να επεκτείνουμε τον Αλγόριθμο ώστε να περιλαμβάνει και περιορισμούς της μορφής

- $f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0$, f_i, g_i convex (DC constraint) ή
- $f_i(\mathbf{x}) = g_i(\mathbf{x})$, f_i, g_i convex. Η περίπτωση αυτή μπορεί να γραφεί ως η συναλήθευση των 2 περιορισμών
$$f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0, g_i(\mathbf{x}) - f_i(\mathbf{x}) \leq 0 \Rightarrow f_i(\mathbf{x}) - g_i(\mathbf{x}) = 0$$

Το σκεπτικό που ακολουθείται κι εδώ εφαρμόζει το ανάπτυγμα Taylor 1ης τάξης στις συναρτήσεις $g_i(\mathbf{x})$ των περιορισμών. Και πάλι χρησιμοποιώντας την ιδιότητα της κυρτότητας της g, γνωρίζουμε ότι

$$f(\mathbf{x}) - g(\mathbf{x}) \leq f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) \tag{3.80}$$

και από την (3.80) συνάγουμε ότι

$$f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) \leq 0 \Rightarrow f(\mathbf{x}) - g(\mathbf{x}) \leq 0 \tag{3.81}$$

Η (3.81) μας λέει ότι όταν οι γραμμικοποιημένοι περιορισμοί ικανοποιούνται, είναι σίγουρο ότι θα ικανοποιούνται οι περιορισμοί του αρχικού προβλήματος. Έτσι, καταλήγουμε με φυσικό τρόπο στη διαδικασία που είναι ευρύτερα γνωστή ως Convex-Concave Procedure και παρατίθεται ως Αλγόριθμος 4. Προφανώς, περιπτώσεις όπου $g_i(\mathbf{x}) = 0, \forall i$ αντιστοιχούν σε προβλήματα κυρτής βελτιστοποίησης. Ο Αλγόριθμος CCP είναι μια περίπτωση αλγορίθμου τύπου Majorization-Minimization, όρος

Algorithm 4 Convex-Concave Procedure

function CCP($d(\mathbf{x}) = f(\mathbf{x}) - g(\mathbf{x})$, constraint functions $f_i, g_i, h_i, \mathbf{x}^0$, maximum iterations K)
  $k \leftarrow 0$
  while $k \leq K$ and stopping criterion not satisfied **do**
    $\tilde{d}(\mathbf{x}; \mathbf{x}_k) \leftarrow f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k)$
    Set x_{k+1} to the solution of

$$\begin{aligned} &\text{Minimize } \tilde{d}(\mathbf{x}; \mathbf{x}_k) \\ &\quad \text{s.t. } \mathbf{x} \in \mathbf{dom}(d) \\ &\qquad f_i(\mathbf{x}) - g_i(\mathbf{x}^k) - \nabla g_i(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) \leq 0, i = 1, \dots m \\ &\qquad h_i(\mathbf{x}) = 0, i = 1, \dots p \end{aligned}$$

    $k \leftarrow k + 1$
  end while
end function

που χρησιμοποιείται εύστοχα από τους Lange et. al. [34]: προσεγγίζει το πρόβλημα προς βελτιστοποίηση με ένα άλλο, ευεπίλυτο πρόβλημα το οποίο είναι υπερεκτίμηση του αρχικού, και προσπαθεί ελαχιστοποιώντας αυτό να ελαχιστοποιήσει το αρχικό πρόβλημα.

3.2.3 Penalty CCP

Μέχρι στιγμής, η επιλογή αρχικών σημείων $\mathbf{x}^0$ στο Convex-Concave Procedure δεν έχει σχολιαστεί, αλλά φαίνεται πως έχουμε υποθέσει πως έχουμε στη διάθεσή μας κάποιον τρόπο να παράξουμε αρχικά σημεία τα οποία ανήκουν στο σύνολο εφικτών λύσεων του προβλήματός μας. Ωστόσο, κάτι τέτοιο δεν είναι πάντα δυνατόν. Χάρη σε μια πρόσφατα προτεινόμενη επέκταση του CCP, τελικά καταλήγει να μην είναι ούτε απαραίτητο. Ο Αλγόριθμος 5 παρουσιάζεται από τους Lipp & Boyd [37] και στηρίζεται στην εξής απλή λογική: επιτρέπουμε σε κάποιους από τους γραμμικοποιημένους περιορισμούς να παραβιάζονται, με την μεταβλητή χαλάρωσης s_i να υποδηλώνει "πόσο" παραβιάζεται ο περιορισμός i. Το κόστος $\sum_{i=1}^{m} s_i$ είναι η ℓ_1 νόρμα των μεταβλητών χαλάρωσης, η οποία είναι αρκετά αποτελεσματική για προβλήματα που απαιτούν αραιές λύσεις [8], και άρα αναμένουμε να μην επιτραπεί σε πολλούς περιορισμούς να παραβιάζονται. Τέλος, επιλέγοντας αρχικά μικρό τ, είμαστε αρκετά "ελαστικοί" σε παραβιάσεις, με την ελπίδα ο αλγόριθμος να καταλήξει σε κάποια περιοχή με μικρότερο κόστος της αντικειμενικής συνάρτησης, και άρα "κοντά" σε ένα χαμηλότερο τοπικό ελάχιστο. Αυξάνοντας το τ σε κάθε βήμα, ακολουθούμε μια στρατηγική εκλέπτυνσης της περιοχής των επιτρεπόμενων λύσεων που περιλαμβάνουν παραβιάσεις των περιορισμών. Μια πρακτική συνέπεια των παραπάνω είναι ότι μπορούμε να επιλέξουμε οποιοδήποτε αρχικό $\mathbf{x}^0$, ακόμα κι αν δεν είναι feasible.
Ο Αλγόριθμος 5 είναι ο αλγόριθμος που υλοποιεί το υπολογιστικό πακέτο `Dccp` [59], το οποίο επεκτείνει τη βιβλιοθήκη κυρτής βελτιστοποίησης CVXPY [18]. Όλα τα πειράματα της παρούσας εργασίας που στηρίζονται στην Convex-Concave Procedure χρησιμοποιούν αυτή τη βιβλιοθήκη.

Algorithm 5 Penalty Convex-Concave Procedure

function PENALTYCCP(Given $\mathbf{x}^0, \tau_0 \geq 0, \tau_{\max}, \mu > 1, K$)
  $k \leftarrow 0$
  while $k \leq K$ and stopping criterion not satisfied **do**
    $\tilde{d}(\mathbf{x}; \mathbf{x}_k) \leftarrow f(\mathbf{x}) - g(\mathbf{x}^k) - \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k)$
    Set x_{k+1} to the solution of

$$\begin{aligned} &\text{Minimize } \tilde{d}(\mathbf{x}; \mathbf{x}_k) + \tau_k \sum_{i=1}^{m} s_i \\ &\quad \text{s.t. } \mathbf{x} \in \mathbf{dom}(d) \\ &\qquad f_i(\mathbf{x}) - g_i(\mathbf{x}^k) - \nabla g_i(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) \leq s_i, \ i = 1, \dots m \\ &\qquad h_i(\mathbf{x}) = 0, \ i = 1, \dots p \\ &\qquad s_i \leq 0, \ i = 1, \dots m \end{aligned}$$

    $k \leftarrow k + 1$
    $\tau_{k+1} := \min(\tau_{\max}, \mu\tau_k)$
  end while
end function

Stopping Criterion

Σε όλους τους Αλγορίθμους που αναφέρονται παραπάνω, υποτίθεται η ύπαρξη ενός κριτηρίου πρόωρου τερματισμού της διαδικασίας βελτιστοποίησης. Στην πράξη, οι συγγραφείς του [37] προτείνουν τη χρήση ενός κατωφλίου βελτίωσης ε, ώστε η βελτιστοποίηση να σταματάει όταν

$$\left(f_0(\mathbf{x}^k) - g_0(\mathbf{x}^k) + \sum_{i=1}^{m} s_i^k\right) - \left(f_0(\mathbf{x}^{k+1}) - g_0(\mathbf{x}^{k+1}) + \sum_{i=1}^{m} s_i^{k+1}\right) \leq \varepsilon \tag{3.82}$$

Στην Εξ. (3.82), η ποσότητα s_i^k είναι η μεταβλητή χαλάρωσης s_i όπως βρέθηκε στο βήμα k του αλγορίθμου.

Κεφάλαιο 4

Το Τροπικό Perceptron

4.1 Το τροπικό perceptron

Η $(\max, +)$ άλγεβρα ανήκει στην λεγόμενη "τροπική" οικογένεια, η οποία περιλαμβάνει και την $(\min, +)$ άλγεβρα η οποία χρησιμοποιείται ευρέως στο δυναμικό προγραμματισμό, την αλληλούχιση DNA και εν γένει διάφορες εφαρμογές στην τεχνητή νοημοσύνη και τη θεωρία βελτιστοποίησης. Για παράδειγμα, οι Maragos & Koutras [42] πρότειναν πρόσφατα ένα μοντέλο δυναμικού συστήματος στηριζόμενο στη $(\max, \times)$ άλγεβρα για την αναγνώριση γεγονότων ενδιαφέροντος σε οπτικοακουστική πληροφορία. Ένα μεγάλο κομμάτι της θεωρίας της τροπικής άλγεβρας έχει θεμελιωθεί ήδη από την δεκαετία του 1980, μέσα από τη δουλειά του Raymond Cuninghame-Green [14], στην οποία αναφέρεται ως minimax algebra. Καθώς το μοντέλο του perceptron που θα παρουσιάσουμε εδώ λειτουργεί με βάση πράξεις από αυτήν, θα το ονομάσουμε τροπικό και θα αναφερόμαστε σε αυτό με τη συντομογραφία TP (tropical perceptron).

Αν και το βασικό μοντέλο που θα παρουσιάσουμε έχει εμφανιστεί ήδη στην εργασία των Ritter-Urcid [53], θα δείξουμε πώς μπορεί να προκύψει ως "τροπικοποίηση" ενός κλασσικού μοντέλου. Έστω $\mathbb{K}$ ένα σώμα (field) το οποίο "εφοδιάζουμε" με μια αποτίμηση (valuation) $\text{val} : \mathbb{K} \to \mathbb{R}$. Μια αποτίμηση, στα πλαίσια της $(\max, +)$ άλγεβρας, είναι μια συνάρτηση που ικανοποιεί:

- $\text{val}(0) = -\infty$
- $\text{val}(x \cdot y) = \text{val}(x) + \text{val}(y)$
- $\text{val}(x + y) \leq \max\{\text{val}(x), \text{val}(y)\}$

Μπορούμε να "μετατρέψουμε" ένα πολυώνυμο από τη γνωστή μορφή

$$f = \sum_i c_i \mathbf{x}^{\boldsymbol{u}_i}$$

στη μορφή

$$\text{trop}(f) = \max_i \{\text{val}(c_i) + \boldsymbol{u}_i \mathbf{x}\} \tag{4.1}$$

δηλαδή προκειμένου να μεταβούμε στη $(\max, +)$ άλγεβρα οι διμελείς πράξεις γίνονται $+ :\to \max$, $\times :\to +$ και οι συντελεστές του πολυωνύμου αντικαθίστανται από τις αποτιμήσεις τους.[1]
Γράφοντας την εξίσωση της διέγερσης ενός νευρώνα ως

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

διαπιστώνουμε ότι είναι ένα άθροισμα μονωνύμων στον πολυωνυμικό δακτύλιο $\mathbb{K}[x_1, x_2, \ldots x_n]$ με μοναδιαίους εκθέτες. Επομένως μπορούμε να ορίσουμε

$$\text{trop}(f) = \max\{\text{val}(b), \max_i\{\text{val}(w_i) + x_i\}\}$$

με βάση την εξίσωση (4.1). Έτσι, καταλήγουμε στο εξής μοντέλο:

[1] Η διαδικασία αυτή μπορεί να συναντηθεί στη βιβλιογραφία ως *τροπικοποίηση* (tropicalization)

Ορισμός 13 (TP)**.** Το τροπικό perceptron (TP) με είσοδο πρότυπα $\mathbf{x} \in \mathbb{R}^n_{\max}$ αποτελείται από ένα σύνολο βαρών $\boldsymbol{w} \in \mathbb{R}^{n+1}_{\max}$ και υπολογίζει στην έξοδό του τη συνάρτηση

$$\tau(\mathbf{x}) = w_0 \vee (w_1 + x_1) \vee \cdots \vee (w_n + x_n) \tag{4.2}$$

ή ισοδύναμα

$$\tau(\mathbf{x}) = \boldsymbol{w} \boxplus [0\ \mathbf{x}]^T$$

Στον παραπάνω ορισμό, η παράμετρος w_0 μπορεί να είναι απούσα (ή να τεθεί ισοδύναμα $w_0 = \varepsilon$), όπως συμβαίνει για παράδειγμα στο μοντέλο που παρουσιάζεται στην [53]. Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε την $(\min, +)$ - άλγεβρα προκειμένου να καταλήξουμε στο δυϊκό μοντέλο του TP, το οποίο θα δίνεται από την εξίσωση

$$\tau'(\mathbf{x}) = m_0 \wedge (m_1 + x_1) \cdots \wedge (m_n + x_n) \tag{4.3}$$

όπου $a \wedge b = \min(a, b)$ και $\boldsymbol{m} \in \mathbb{R}^{n+1}_{\min}$ το διάνυσμα βαρών. Όπως και στην προηγούμενη περίπτωση θέτουμε την παράμετρο $m_0 = \varepsilon' = \infty$ σε περίπτωση που θέλουμε να μην χρησιμοποιήσουμε "κατωφλίωση".

4.1.1 Το τροπικό perceptron ως μορφολογικό φίλτρο

Στο σημείο αυτό, είναι θεμιτό να εξετάσουμε μια εναλλακτική αντιμετώπιση του τροπικού perceptron και των 2 variants του (max ή min), σαφώς επηρεασμένη από το πεδίο της όρασης υπολογιστών και συγκεκριμένα της μαθηματικής μορφολογίας. Στην πραγματικότητα, πράξεις αντίστοιχες με αυτές που πραγματοποιεί ένα tropical perceptron χρησιμοποιούνται ευρέως στο πεδίο της μαθηματικής μορφολογίας, στο οποίο η ονοματοδοσία είναι αρκετά πιο διαισθητική.
Στο πεδίο αυτό, όπως και της όρασης υπολογιστών, ονομάζουμε *εικόνα*(image) μια συνάρτηση $I : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, \infty\}$, δηλαδή μια απεικόνιση ενός ευκλείδειου διανυσματικού χώρου (ή διατεταγμένου πλέγματος, αν $\mathbf{dom}(I) = \mathbb{Z}^d$) στην εκτεταμένη ευθεία των πραγματικών αριθμών. Η βασική ιδέα της μαθηματικής μορφολογίας περιστρέφεται γύρω από τα λεγόμενα **δομικά στοιχεία** (*structuring elements*): μικρά, προκαθορισμένα σχήματα ή σύνολα τα οποία αντιπαρατίθενται με ένα σύνολο ή εικόνα προς εξέταση και εξετάζεται κατά πόσο ομοιάζουν σε περιοχές της εικόνας. Συνήθως τα structuring elements αναπαρίστανται με το σύμβολο B. Με B^s συμβολίζουμε το "συμμετρικό" δομικό στοιχείο, έτσι ώστε

$$B^s = \{\mathbf{x} \in E : -\mathbf{x} \in B\}$$

όπου E ένας διανυσματικός χώρος ή πλέγμα με διάταξη.

Ορισμός 14 (Dilation/Erosion)**.** Έστω $B : E \to \mathbb{R}$ ένα structuring element και $I : E \to \overline{\mathbb{R}}$ μια εικόνα. Ορίζουμε τους τελεστές διαστολής (*dilation*) και συστολής (*erosion*) ως προς το στοιχείο B ως εξής:

1. $(I \oplus B)(x) = \bigvee_{y \in E} [I(y) + B(x - y)]$ για το dilation
2. $(I \ominus B)(x) = \bigwedge_{y \in E} [I(y) - B(y - x)]$ για το erosion

Έτσι, δεν είναι δύσκολο να παρατηρήσουμε ότι κάθε τύπος τροπικού perceptron αντιστοιχεί σε ένα μορφολογικό φίλτρο πεπερασμένου μεγέθους, το οποίο αποτιμάται σε ένα συγκεκριμένο σημείο του χώρου/χρόνου: το τροπικό perceptron με χρήση του max ουσιαστικά επιδρά στο πλέγμα $[1, \dots n]$ που αντιστοιχεί στα στοιχεία $[x_1, \dots x_n]$ του $\mathbf{x}$ και υπολογίζει την παράσταση

$$\tau(\mathbf{x}) = \boldsymbol{w}^T \boxplus \mathbf{x} = \bigvee_{i=1}^{n} x_i + w_i \tag{4.4}$$

ενώ με τη χρήση του min έχουμε

$$\tau'(\mathbf{x}) = \boldsymbol{w}^T \boxplus' \mathbf{x} = \bigwedge_{i=1}^{n} x_i + m_i \tag{4.5}$$

Σε κάθε περίπτωση, και χωρίς να μας απασχολεί η ακριβής σειρά των συντελεστών w_i, m_i στις παραπάνω παραστάσεις, μπορούμε να θεωρήσουμε ότι:

- κάθε tropical max-perceptron αποτελεί ένα dilation filter της μορφής $\mathbf{x} \oplus \boldsymbol{W}_{\max}$
- κάθε tropical min-perceptron αποτελεί ένα erosion filter της μορφής $\mathbf{x} \ominus \boldsymbol{W}_{\min}$

Για περισσότερες λεπτομέρειες σχετικά με τη θεωρία και τις εφαρμογές της μαθηματικής μορφολογίας, ο αναγνώστης παραπέμπεται στα [40, 58, 60]. Μια εναλλακτική αντιμετώπιση της θεωρίας της μαθηματικής μορφολογίας με σκοπό τη σχεδίαση μηχανών μάθησης μπορεί να βρεθεί στο [63].

4.1.2 Η γεωμετρία του τροπικού perceptron

Σε αυτή την ενότητα, θα εξετάσουμε τη γεωμετρία της επιφάνειας απόφασης ενός τροπικού perceptron (στη $(\max, +)$ άλγεβρα) για την περίπτωση δυαδικής ταξινόμησης. Στην περίπτωση αυτή, θεωρούμε ότι διαθέτουμε 2 κλάσεις προτύπων, $\mathcal{C}_0, \mathcal{C}_1$, και μια συλλογή από K ζεύγη $(\mathbf{x}^k, y_k), k = 1 \dots K$.
Για κάθε ζεύγος $(\mathbf{x}^k, y_k)$, θεωρούμε ότι ισχύει $y_k = -1$ εάν $\mathbf{x}^k \in \mathcal{C}_0$ και $y_k = 1$ εάν $\mathbf{x}^k \in \mathcal{C}_1$. Στη συνέχεια, μας δίνονται πρότυπα $\mathbf{x}$ προς ταξινόμηση, και στην πράξη συνήθως ορίζουμε ότι

$$\mathbf{x} \in \begin{cases} C_1 & \text{εαν } \phi(\mathbf{x}) \geq 0 \\ C_0 & \text{εαν } \phi(\mathbf{x}) < 0 \end{cases}$$

όπου $\phi(\mathbf{x})$ η έξοδος του ταξινομητή (π.χ. του TP) για είσοδο το πρότυπο $\mathbf{x}$. Έτσι, ένας δυαδικός ταξινομητής χωρίζει το χώρο, $\mathbb{R}^d$ σε 2 περιοχές που ονομάζονται επιφάνειες απόφασης(decision surfaces) $\mathcal{R}_1, \mathcal{R}_0$. Η περιοχή $(\mathcal{R}_1 \cap \mathcal{R}_0)$ ονομάζεται σύνορο απόφασης (decision boundary).
Θα δείξουμε ότι οι επιφάνειες απόφασης που ορίζει ένα τροπικό perceptron είναι τροπικά πολύεδρα.

Ορισμός 15 (Τροπικός ημιχώρος)**.** Έστω $a, b \in \mathbb{R}^{n+1}_{\max}$. Ένας **αφφινικός τροπικός ημιχώρος** (affine tropical halfspace) είναι ένα υποσύνολο του $\mathbb{R}^n_{\max}$ της μορφής

$$\max\{a_1 + x_1, \ldots, a_n + x_n, a_{n+1}\} \geq \max\{b_1 + x_1, \ldots, b_n + x_n, b_{n+1}\} \tag{4.6}$$

ή, στο γνωστό πλέον συμβολισμό

$$\left(\bigvee_{i=1}^{n} a_i + x_i\right) \vee a_{n+1} \geq \left(\bigvee_{i=1}^{n} b_i + x_i\right) \vee b_{n+1} \tag{4.7}$$

Χωρίς βλάβη της γενικότητας και με βάση την παρατήρηση στο [23, Lemma 1], μπορούμε να θεωρήσουμε ότι σε κάθε ανισότητα της μορφής (4.7), ισχύει:

$$\begin{aligned} \max_{i=1\ldots n+1} \min\{a_i, b_i\} &= -\infty \\ \rightarrow \min\{a_i, b_i\} &= -\infty \ \forall i \in [1, n+1] \end{aligned} \tag{4.8}$$

δηλαδή μπορούμε εν τέλει να γράψουμε κάθε ανισότητα ως

$$\left(\bigvee_{i\in I} a_i + x_i\right) \vee a_{n+1} \geq \left(\bigvee_{j\in J} a_j + x_j\right) \vee b_{n+1} \tag{4.9}$$

όπου $I \cap J = \emptyset, I \cup J = N, \min\{a_{n+1}, b_{n+1}\} = -\infty$.
Κατ' αντιπαραβολή με τη μορφή (4.2), καταλήγουμε στο εξής:

Πρόταση 8. Έστω $(\mathbf{x}^k)$ ένα γνωστό πρότυπο εκπαίδευσης με γνωστή τιμή y_k. Τότε, το ζεύγος $(\mathbf{x}^k, y_k)$ προς ταξινόμηση από ένα perceptron της μορφής (4.2) ορίζει έναν τροπικό αφφινικό ημιχώρο.

Απόδειξη. Εάν $y_k = 1$, θέλουμε το perceptron να δίνει έξοδο $\tau(\mathbf{x}^k) \geq 0$ (και αντίστοιχα $\tau(\mathbf{x}^k) < 0$ σε περίπτωση που $y_k = -1$. Χωρίς βλάβη της γενικότητας, θα υποθέσουμε ότι $y_k = 1$. Έτσι, το $\mathbf{x}^{(k)}$ ορίζει την επιφάνεια

$$\begin{aligned} \tau(\mathbf{x}^k) &\geq 0 \\ \max\{w_1 + x_1, \ldots, w_n + x_n\} &\geq 0 \end{aligned} \tag{4.10}$$

Όμως αυτή είναι ακριβώς η μορφή (4.7) με

- $b_1, b_2, \ldots b_n = \varepsilon, b_{n+1} = 0$

- $a_i = \mathbf{x}_i^k, a_{n+1} = \varepsilon$
- $x_i = w_i \; \forall i \in N$

Έτσι, κάθε ζεύγος $(\mathbf{x}^k, y_k)$ ορίζει έναν τροπικό ημιχώρο της μορφής $\boldsymbol{w}^T \boxplus \mathbf{x}^k \gtrless 0$, ανάλογα με το πρόσημο της τιμής y_k. □

Στην πράξη, στην φάση της εκπαίδευσης ενός ταξινομητή παρουσιάζεται ένα πεπερασμένο σύνολο ζευγών $(X, Y) = \{(\mathbf{x}^k, y_k) : k \in 1 \dots K\}$ από πρότυπα εκπαίδευσης και οι παράμετροι του ταξινομητή προσαρμόζονται για την ελαχιστοποίηση κάποιας συνάρτησης κόστους. Κάθε τέτοιο ζεύγος ορίζει έναν τροπικό ημιχώρο, επομένως μια πεπερασμένη συλλογή προτύπων εκπαίδευσης ορίζει ένα *τροπικό πολύεδρο*.

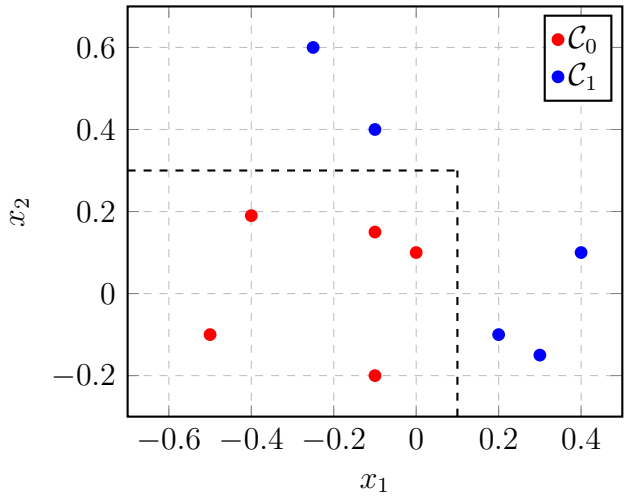
Ορισμός 16 (Τροπικό πολύεδρο)**.** Έστω $\boldsymbol{A} \in \mathbb{R}_{\max}^{m \times n}, \boldsymbol{B} \in \mathbb{R}_{\max}^{m \times n}, \boldsymbol{c} \in \mathbb{R}_{\max}^{m}, \boldsymbol{d} \in \mathbb{R}_{\max}^{m}$. Η ανισότητα

$$\boldsymbol{c} \vee (\boldsymbol{A} \boxplus \mathbf{x}) \leq \boldsymbol{d} \vee (\boldsymbol{B} \boxplus \mathbf{x}) \tag{4.11}$$

ορίζει ένα σύνολο από ανισότητες της μορφής

$$\max\left(c_i, \max_{j \in N}\{A_{ij} + x_j\}\right) \leq \max\left(d_i, \max_{j \in N}\{B_{ij} + x_j\}\right), \; \forall i \in M \tag{4.12}$$

Το σύνολο $P \subseteq \mathbb{R}_{\max}^n$ για το οποίο ικανοποιούνται οι παραπάνω ανισότητες λέγεται **τροπικό πολύεδρο** και είναι η τομή ενός πεπερασμένου αριθμού τροπικών ημιχώρων.



Σχήμα 4.1: Τροπικώς γραμμικά διαχωρίσιμα πρότυπα, $(w_1, w_2) = (-0.1, -0.3)$

Πρόταση 9 (Το σύνολο εφικτών λύσεων είναι Τροπικό Πολύεδρο)**.** Έστω ένα σύνολο προτύπων $(\mathbf{x}^k, y_k)_{k\in K}$, από 2 κλάσεις $\mathcal{C}_0, \mathcal{C}_1$, χωριζόμενο στους πίνακες $\boldsymbol{X}_{\text{pos}}$ με γραμμές τα "θετικά" και $\boldsymbol{X}_{\text{neg}}$ με γραμμές τα "αρνητικά" πρότυπα, αντίστοιχα. Αν $\mathbf{x}^k \in \mathcal{C}_1 \Rightarrow y_k > 0, \mathbf{x}^k \in \mathcal{C}_0 \Rightarrow y_k < 0$, θεωρούμε το τροπικό πολύεδρο που ορίζεται ως

$$\mathcal{T}(\boldsymbol{X}_{\text{pos}}, \boldsymbol{X}_{\text{neg}}) = \{\boldsymbol{w} \in \mathbb{R}^n : \boldsymbol{X}_{\text{pos}} \boxplus \boldsymbol{w} \geq 0,\ \boldsymbol{X}_{\text{neg}} \boxplus \boldsymbol{w} \leq 0\} \tag{4.13}$$

Έτσι, είναι εύκολο να παρατηρήσουμε ότι ένα σύνολο προτύπων είναι διαχωρίσιμο από ένα τροπικό perceptron αν και μόνο αν το πολύεδρο που ορίζει η (4.13) είναι μη-κενό.

Στην εργασία των Akian, Gaubert & Guterman [1], αποδεικνύεται ότι το γενικό ερώτημα για το αν ένα τροπικό πολύεδρο είναι κενό ανάγεται στην ύπαρξη μιας "νικήτριας" αρχικής κατάστασης σε μια κλάση αλγοριθμικών παιχνιδιών τύπου minimax, τα οποία είναι γνωστά στη βιβλιογραφία ως mean payoff games. Οι πίνακες $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{c}, \boldsymbol{d}$ δίνουν τις επιλογές των παικτών σε αυτή την κλάση παιχνιδιών. Στη συνέχεια της εργασίας, θα αποκαλούμε ένα σύνολο προτύπων προς (δυαδική) ταξινόμηση **τροπικώς γραμμικά διαχωρίσιμο** (tropically linearly separable) εάν το τροπικό πολύεδρο που ορίζουν είναι μη-κενό. Ένα παράδειγμα από τροπικώς γραμμικά διαχωρίσιμα πρότυπα φαίνεται στην Εικόνα 4.1. Ακόμη, στην ενότητα 4.2, θα δείξουμε ότι η διαχωρισιμότητα ενός συνόλου προτύπων από ένα τροπικό perceptron μπορεί να ελεγχθεί σε πολυωνυμικό χρόνο, άρα ταυτόχρονα μπορούμε να αποφανθούμε αν το σχετικό τροπικό πολύεδρο είναι κενό.

4.1.3 Convex-Concave Programming για τον προσδιορισμό των w_i

Δεδομένου ενός προβλήματος δυαδικής ταξινόμησης, μπορούμε να εκφράσουμε το πρόβλημα του προσδιορισμού ενός συνόλου βαρών w_i για το τροπικό perceptron ως ένα πρόβλημα βελτιστοποίησης, όπως συμβαίνει με την περίπτωση ενός Support Vector Machine. Στην περίπτωσή μας, το πρόβλημα διατυπώνεται ως εξής:

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{K} ||\tau(\mathbf{x}^k) - y_k||_p \\ \text{s.t. } & y_k \cdot \max(w_1 + x_1^k, \dots w_n + x_n^k) \geq 0 \end{aligned} \tag{4.14}$$

Στο πρόβλημα (4.14), η $||\cdot||_p$ είναι η ℓ_p νόρμα. Η παράσταση $y_k \cdot \boldsymbol{w}^T \boxplus \mathbf{x}^k$ δεν είναι πλέον affine, όπως συμβαίνει στην περίπτωση του απλού support vector machine και άρα το πρόβλημα δεν αποτελεί στιγμιότυπο προβλήματος κυρτής βελτιστοποίησης. Μπορούμε να διατυπώσουμε το πρόβλημα με την εναλλακτική μορφή:

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{K} \max(\xi_k, 0) \\ \text{s.t. } & \max(w_1 + x_1^k, \dots w_n + x_n^k) \leq \xi_k && \text{αν } y_k = -1 \\ & \max(w_1 + x_1^k, \dots w_n + x_n^k) \geq -\xi_k && \text{αν } y_k = 1 \end{aligned} \tag{4.15}$$

Η συνάρτηση κόστους συνεισφέρει για κάποιο k μόνο όταν $\xi_k > 0$, δηλαδή μόνο στην περίπτωση που υπάρχει κάποιο misclassification. Στην παραπάνω μορφή, η συνάρτηση κόστους είναι κυρτή, όπως και η πρώτη από τις 2 οικογένειες περιορισμών, ενώ η 2η οικογένεια περιορισμών είναι κοίλη. Μπορούμε να χρησιμοποιήσουμε κάποιον αλγόριθμο convex-concave βελτιστοποίησης, όπως αυτόν που παρουσιάζεται στο [59, Algorithm 1.1] αλλά και στο Κεφάλαιο 3.2 ως Αλγόριθμος 5. Ο αλγόριθμος επιλύει προβλήματα που εκφράζονται ως διαφορές κυρτών συναρτήσεων (difference-of-convex, DC). Είναι εύκολο να εκφράσουμε ανισότητες με κοίλες συναρτήσεις (που δεν επιτρέπονται σε διατυπώσεις για κυρτή βελτιστοποίηση) στη μορφή

$$g_i(\mathbf{x}) \geq 0 \to f_i = 0, f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0$$

ενώ περιορισμοί που περιλαμβάνουν ισότητες της μορφής $g_i(\mathbf{x}) = f_i(\mathbf{x})$ μπορούν να γραφούν ως

$$\begin{cases} f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0 \\ g_i(\mathbf{x}) - f_i(\mathbf{x}) \leq 0 \end{cases}$$

Το πρόβλημά μας, (4.15), αποτελεί προφανώς στιγμιότυπο αυτής της οικογένειας προβλημάτων βελτιστοποίησης, καθώς η παράσταση

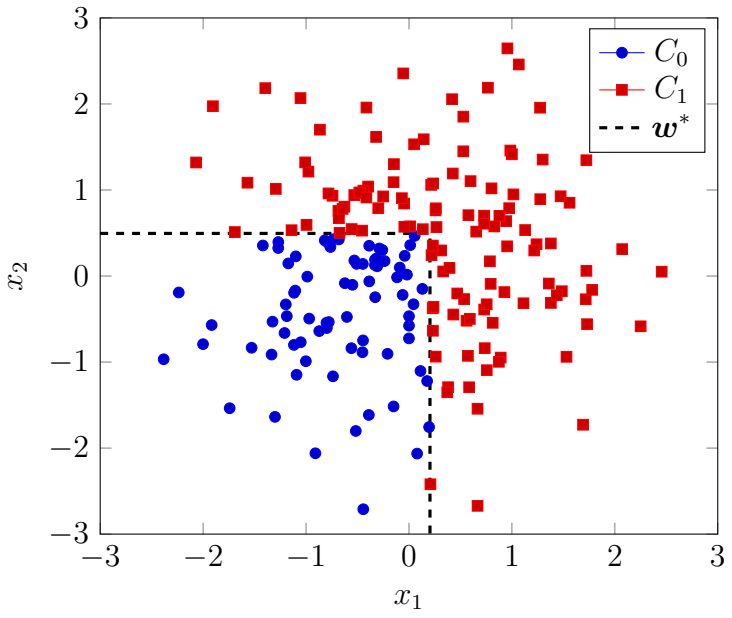
$$f_k(\boldsymbol{w}) = \max(w_1 + x_1^k, \dots w_n + x_n^k)$$

αποτελεί κυρτή συνάρτηση ως προς $\boldsymbol{w}$ αφού είναι το maximum από αφφινικές (affine) συναρτήσεις της μορφής $ax + b$, ενώ το άθροισμα $f_k(\mathbf{x}) \pm \xi_k$ είναι κι αυτό κυρτό αφού είναι άθροισμα μιας κυρτής συνάρτησης και ενός affine όρου. Ακόμη, το κόστος

$$f_0(\boldsymbol{w}) = \sum_k \max(\xi_k, 0)$$

είναι κι αυτό κυρτό αφού είναι το άθροισμα από κυρτούς όρους, καθώς το maximum από affine συναρτήσεις είναι κυρτή συνάρτηση. Η σχετική θεωρία μπορεί να βρεθεί στο [8], ενώ η τεχνική του disciplined convex programming που χρησιμοποιείται από διάφορους solvers για προβλήματα κυρτής βελτιστοποίησης περιγράφεται λεπτομερώς στο [27] και παρουσιάστηκε στο Κεφάλαιο 1.5.2.

Για πειραματικούς σκοπούς, χρησιμοποιήσαμε τον solver σε `python` που παρέχουν οι συγγραφείς του [59], ο οποίος στηρίζεται στη γλώσσα μοντελοποίησης CVXPY [18] και παρατηρήσαμε ότι επιστρέφει κατάλληλες αναθέσεις βαρών w_i για τροπικώς γραμμικά διαχωρίσιμα πρότυπα, παρ'όλο που οι συγγραφείς δεν παρέχουν κάποια εγγύηση σύγκλισης στο ολικό ελάχιστο, όπως φαίνεται π.χ. στην εικόνα 4.2.



Σχήμα 4.2: Διαχωριστική επιφάνεια, $\boldsymbol{w} = (-0.20394, -0.4951)$

Καταλληλότητα της Μεθόδου

Στο σημείο αυτό, είναι χρήσιμο να σχολιάσουμε την επιλογή της Convex-Concave Procedure για την εκπαίδευση του τροπικού perceptron. Αν παρατηρήσουμε προσεκτικά τη μορφή των περιορισμών,

θα παρατηρήσουμε ότι είναι piecewise affine περιορισμοί, δηλαδή για όλα τα πιθανά σημεία του πεδίου ορισμού $\mathbf{x}^k$ εκτός από ένα πεπερασμένο σύνολο σημείων όπου η έκφραση

$$g(\mathbf{x}) = \max(w_1 + x_1^k, \ldots, w_n + x_n^k)$$

δεν είναι παραγωγίσιμη, ισχύει ότι

$$\exists \epsilon > 0, \mathcal{B}_\epsilon : \forall \mathbf{x} \in \mathcal{B}_\epsilon, g(\mathbf{x}) = g(\mathbf{x}^k) + \nabla g(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k)$$

Δηλαδή, με κέντρο σημεία όπου η συνάρτηση του περιορισμού είναι παραγωγίσιμη, υπάρχει ευκλείδεια μπάλα $\mathcal{B}_\epsilon$ όπου η γραμμικοποιημένη έκδοση του περιορισμού τον περιγράφει με απόλυτη ακρίβεια. Δυστυχώς, η συμπεριφορά του προγράμματος (4.15) δεν είναι εύρωστη σε περίπτωση που υπάρχει "θόρυβος" στα δεδομένα εκπαίδευσης ή παρουσία outliers. Επομένως είναι σκόπιμο να επαναδιατυπωθεί το πρόβλημα ταξινόμησης χρησιμοποιώντας κάποια εύρωστη συνάρτηση κόστους (π.χ. hinge loss) ή να προηγείται κάποιο στάδιο προεπεξεργασίας για την αφαίρεση των outliers. Ένα παράδειγμα τέτοιων τεχνικών "ευρωστοποίησης" ενός κλασσικού SVM, μπορεί να βρεθούν στο [68].

4.1.4 Ευρετικές μέθοδοι για περιπτώσεις μη-διαχωρισιμότητας

Ορμώμενοι από την προηγούμενη παρατήρηση, αναζητούμε αρχικά μια ευρετική μέθοδο για την τροποποίηση του προγράμματος (4.15) προκειμένου ο ταξινομητής μας να είναι εύρωστος σε περίπτωση που τα δεδομένα εκπαίδευσης δεν είναι τροπικώς γραμμικά διαχωρίσιμα. Διαισθητικά, αυτό που αναζητούμε είναι ένα μια ποσοτικοποίηση της έννοιας "καλό πρότυπο εκπαίδευσης".

Μια απλή προσέγγιση του προβλήματος είναι η ακόλουθη: θεωρούμε πως τα πρότυπα των κλάσεων $\mathcal{C}_0, \mathcal{C}_1$ είναι δείγματα κατανομών με μέσους

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{k \in \mathcal{C}_i} \mathbf{x}^k$$

Χωρίς να κάνουμε κάποια επιπλέον υπόθεση για την κατανομή, μπορούμε να ορίσουμε τον εξής όρο "κανονικοποίησης" για κάθε πρότυπο που ανήκει στην κλάση $\mathcal{C}_i$:

$$\lambda_k = \frac{1}{||\mathbf{x}^k - \boldsymbol{\mu}_i||_p} \tag{4.16}$$

δηλαδή το κόστος λάθους ταξινόμησης (misclassification penalty) που "συνεισφέρει" κάθε πρότυπο σταθμίζεται με το αντίστροφο της απόστασής του από το δειγματικό μέσο της κλάσης του. Σαν ένα έξτρα βήμα, μπορούμε να κανονικοποιήσουμε τους συντελεστές λ στο εύρος $(0, 1]$ αντικαθιστώντας τους με

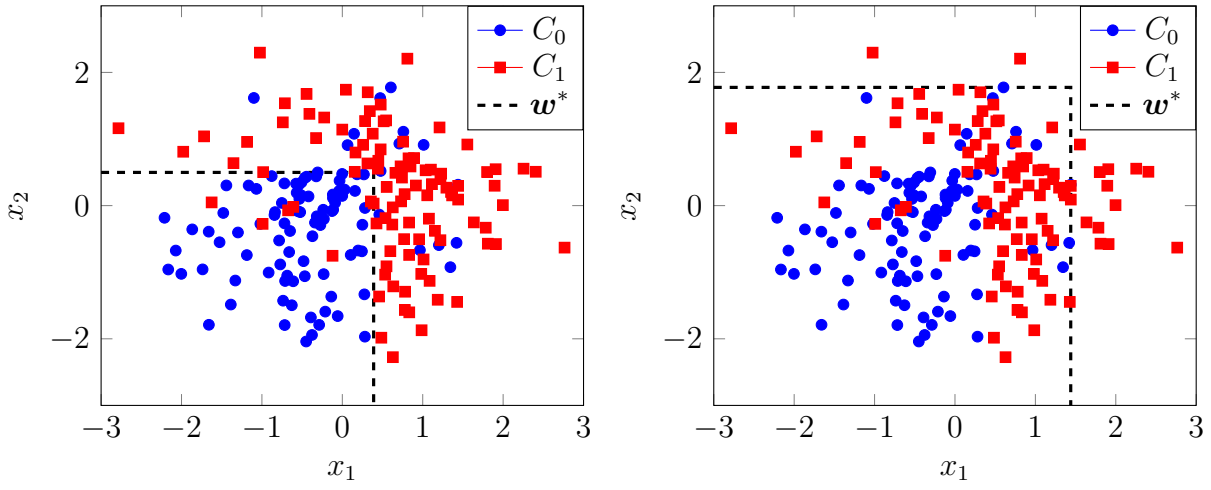
$$\nu_k = \frac{\lambda_k}{\max_{i:\mathbf{x}^{(i)} \in \mathcal{C}_j} \lambda_i}$$

Συνεπώς, το νέο κόστος γράφεται ως

$$\text{minimize} \sum_{k=1}^{K} \nu_k \cdot \max(\xi_k, 0) \tag{4.17}$$

$$\begin{aligned} \text{s.t. } & \max(w_1 + x_1^k, \dots w_n + x_n^k) \leq \xi_k & \text{αν } y_k = -1 \\ & \max(w_1 + x_1^k, \dots w_n + x_n^k) \geq -\xi_k & \text{αν } y_k = 1 \end{aligned}$$

Διαισθητικά, περιμένουμε τα πρότυπα που αποτελούν outliers να βρίσκονται μακριά από τον δειγματικό μέσο της κλάσης και επομένως να συνεισφέρουν λιγότερο στο συνολικό κόστος της διαδικασίας βελτιστοποίησης σε κάθε βήμα. Πράγματι, επιλέγοντας $p = \infty$ (απόσταση Chebyshev), βλέπουμε πως αυτή η απλή ευρετική μέθοδος αυξάνει εντυπωσιακά την ευρωστία της διαδικασίας προσδιορισμού βαρών. Προσαρμόζουμε το πείραμά μας επιλέγοντας ένα σύνολο τροπικώς γραμμικά διαχωρίσιμων προτύπων και αλλάζοντας τα class labels ενός τυχαία επιλεγμένου ποσοστού τους, r. Κάποια αποτελέσματα φαίνονται παρακάτω:

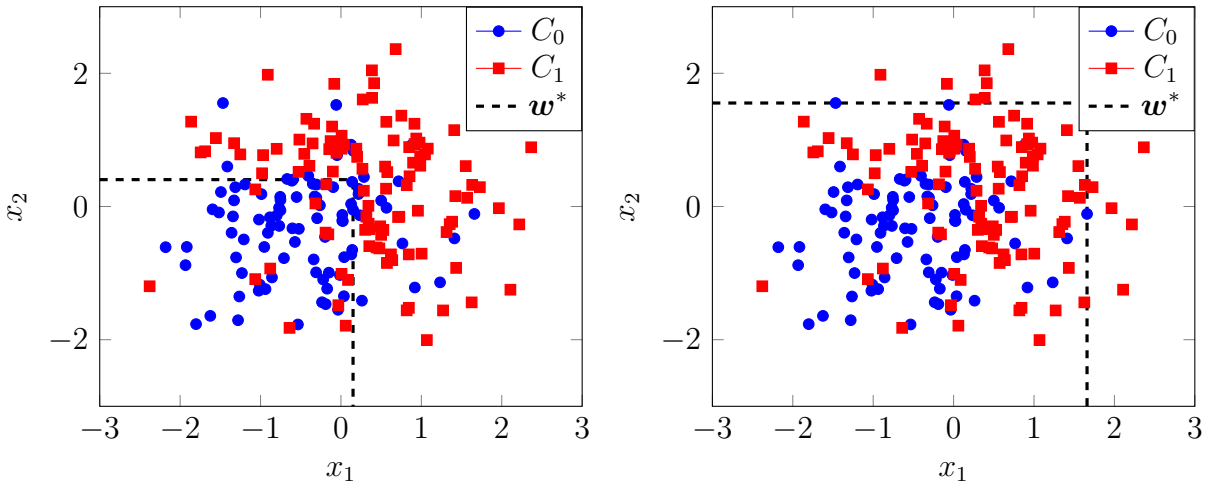


(α′) Με ομαλοποίηση **(β′)** Χωρίς ομαλοποίηση

Σχήμα 4.3: Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 10\%$

Παρατηρούμε ότι το πρόγραμμα (4.15) είναι ιδιαίτερα ευαίσθητο στα outliers, καθώς αυτά συνεισφέρουν πολύ περισσότερο στο κόστος $\sum_k \max(\xi_k, 0)$ χωρίς την ομαλοποίηση. Αντίθετα, η ομαλοποίηση σταθμίζει αυτό το κόστος με το πόσο "αντιπροσωπευτικό" είναι το κάθε misclassified πρότυπο και οι προκύπτουσες διαχωριστικές επιφάνειες που παράγονται από το πρόγραμμα (4.17) είναι αρκετά κοντά στις βέλτιστες. Η ακρίβεια για διάφορες περιπτώσεις μπορεί να φανεί συγκριτικά στην Εικόνα 4.6. Η παραπάνω μέθοδος φαίνεται να προσφέρει ένα σημαντικό βαθμό ευρωστίας στην απόδοση του τροπικού perceptron στην περίπτωση ενός συνόλου τροπικώς μη-γραμμικά διαχωρίσιμων προτύπων σε σχέση με την απλή υλοποίηση. Ωστόσο, στηρίζεται στην υπεραπλουστευτική υπόθεση ότι τα πρότυπα μιας κλάσης ακολουθούν την ίδια κατανομή (ή ακόμη και ότι οι 2 κλάσεις χαρακτηρίζονται από ίδιο τύπο κατανομής, π.χ. gaussian). Μπορούμε παρ'όλ'αυτά να επεκτείνουμε τη μέθοδό μας για πιο μεγάλα και πολύπλοκα σύνολα δεδομένων. Δυο προτάσεις ακολουθούν παρακάτω:

- χρήση κάποιας μεθόδου συσταδοποίησης (clustering) - ανά κλάση - όπως ο αλγόριθμος K-means και προσαρμογή των όρων ομαλοποίησης λ_k με βάση το κεντροειδές (centroid), $\boldsymbol{\mu}_i^c$,



(α′) Με ομαλοποίηση (β′) Χωρίς ομαλοποίηση

Σχήμα 4.4: Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 15\%$

στο οποίο ανατίθεται κάθε πρότυπο

- χρήση ενός μοντέλου μίξης γκαουσσιανών (Gaussian mixture model, GMM) χαμηλής τάξης για τη μοντελοποίηση των δεδομένων κάθε κλάσης και προσαρμογή του με τη χρήση του αλγορίθμου EM (expectation-maximization). Οι όροι ομαλοποίησης μπορούν κατόπιν να τεθούν ίσοι με

$$\nu_k = p(\mathbf{x}^{(k)}|\theta)$$

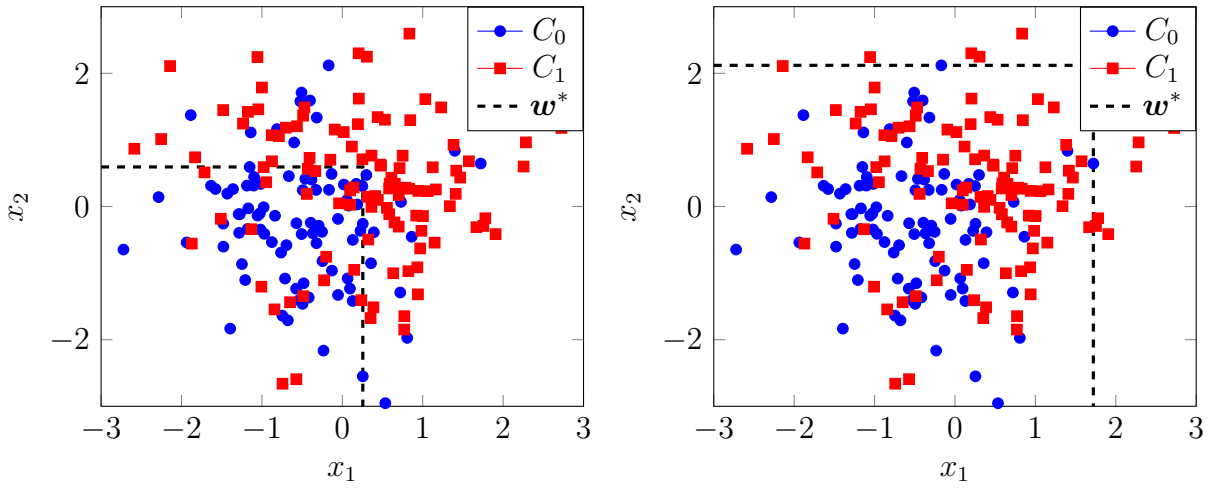
αφού αυτό εξασφαλίζει τον περιορισμό τους στο εύρος $[0, 1]$ αλλά και την κατάλληλη στάθμιση των outliers

- Συνδυασμός οποιωνδήποτε εκ των παραπάνω μεθόδων με κάποιο κατάλληλο γραμμικό μετασχηματισμό που μεγιστοποιεί την διασπορά μεταξύ των κλάσεων, όπως είναι π.χ. η μέθοδος LDA (linear discriminant analysis)

Τέλος, πρέπει να σημειώσουμε ότι με αυτή τη μέθοδο αποφεύγουμε να στηριζόμαστε σε κατασκευαστικές μεθόδους για την εκπαίδευση τροπικών perceptrons, όπως για παράδειγμα οι Αλγόριθμοι που προτείνονται στα [53, 61].

Αναλλοίωτες της μεθόδου Ένα ερώτημα που είναι φυσικό να έχει προκύψει σε αυτό το σημείο, είναι κατά πόσο ο αλγόριθμός μας για την προσαρμογή των βαρών ενός Tropical Perceptron είναι εύρωστος σε μετασχηματισμούς των δεδομένων. Ειδικότερα, είναι εύλογο να ζητούμε ο αλγόριθμός μας να είναι:

- *translation invariant* - αναλλοίωτος ως προς τη μετατόπιση, δηλαδή να μπορούμε να εφαρμόσουμε το μετασχηματισμό $\lambda_a(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n = a + \mathbf{x}$ σε όλα τα πρότυπα



(**α′**) Με ομαλοποίηση (**β′**) Χωρίς ομαλοποίηση

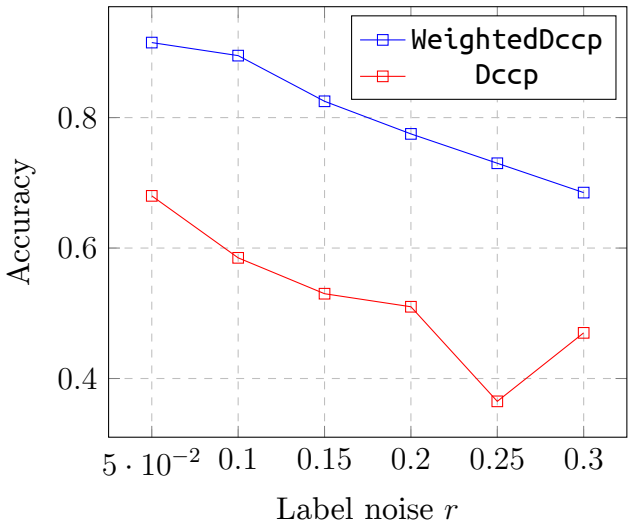
Σχήμα 4.5: Παραγόμενες διαχωριστικές επιφάνειες για συνθετικά δεδομένα, ποσοστό "θορύβου" $r = 20\%$

- *scaling invariant* - αναλλοίωτος ως προς την κλιμάκωση, δηλαδή να μπορούμε να εφαρμόσουμε το μετασχηματισμό $\sigma_a(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n = a \cdot \mathbf{x}$ σε όλα τα πρότυπα
- *rotation invariant* - αναλλοίωτος ως προς τη στροφή, δηλαδή να μπορούμε να εφαρμόσουμε ένα μετασχηματισμό $\rho : \mathbb{R}^n \to \mathbb{R}^n = \boldsymbol{T} \cdot \mathbf{x}$ ο οποίος στρέφει ως προς κάποιους άξονες σε όλα τα πρότυπα $\mathbf{x}$

Όσον αφορά την πρώτη απαίτηση, είναι εύκολο να διαπιστώσουμε ότι μια μετατόπιση κατά οποιοδήποτε διάνυσμα $\vec{a} \in \mathbb{R}^n$ δεν επηρεάζει τις σχετικές αποστάσεις μεταξύ των προτύπων, αφού μπορούμε κατά σύμβαση να θέσουμε τη νέα "αρχή των αξόνων" στο σημείο $\mathbf{0}' = \mathbf{0} + \vec{a}$ και να καταλήξουμε στην ίδια ακριβώς ανάθεση βαρών όπως και προηγουμένως. Η ικανοποίηση της δεύτερης απαίτησης είναι αντίστοιχα απλή. Ωστόσο, η περίπτωση της περιστροφής αναμένεται να παρουσιάζει εμπόδια, ειδικά στην περίπτωση των μη-διαχωρίσιμων (κατά την τροπική έννοια) προτύπων. Για παράδειγμα, αν στα πρότυπα έχει εφαρμοστεί μόνο μια περιστροφή $\boldsymbol{T}$, αυτόματα το νέο πρόβλημα έχει νέους περιορισμούς:

$$\max(w_1 + \boldsymbol{T}_1\mathbf{x}^k, \dots w_n + \boldsymbol{T}_n\mathbf{x}^k) \leq \xi_k \text{ αν } y_k = -1$$
$$\max(w_1 + \boldsymbol{T}_1\mathbf{x}^k, \dots w_n + \boldsymbol{T}_n\mathbf{x}^k) \geq \xi_k \text{ αν } y_k = 1$$

όπου $\boldsymbol{T}_i$ η i-οστή γραμμή του πίνακα $\boldsymbol{T}$, δηλαδή καλούμαστε να εκιμήσουμε όχι μόνο ένα σύνολο βαρών $\boldsymbol{w}$ αλλά και έναν γραμμικό μετασχηματισμό $\boldsymbol{T}$, κάτι που αυξάνει άμεσα την πολυπλοκότητα του προβλήματος. Ακόμη, ένα ζήτημα που προκύπτει για την `WeightedDccp` μέθοδο, είναι το ότι η διαδικασία για τον προσδιορισμό του "discounting" ν_k θα πρέπει κι αυτή να είναι rotation-invariant. Στο παράδειγμα των πειραμάτων που παρουσιάστηκαν παραπάνω, δε θα μπορούσαμε πλέον να επιλέξουμε την απόσταση Chebyshev, καθώς δεν είναι αναλλοίωτη ως προς τη στροφή. Αντίθετα, θα



Σχήμα 4.6: Ακρίβεια μεθόδου με και χωρίς ομαλοποίηση για συνθετικά δεδομένα

μπορούσαμε να χρησιμοποιήσουμε την ευκλείδεια απόσταση.
Στην Εικόνα 4.7, φαίνεται η ακρίβεια των 2 μεθόδων για ένα σύνολο από πρότυπα τα οποία αρχικά ήταν τροπικώς γραμμικά διαχωρίσιμα και έπειτα περιστράφηκαν με ένα γραμμικό μ/μό T. Πριν την περιστροφή, ένα τυχαία επιλεγμένο ποσοστό r των class labels άλλαξε, όπως και στα προηγούμενα πειράματα. Όπως μαρτυρούν τα αποτελέσματα, η απλή μέθοδος `Dccp` αποτελεί παράδειγμα προς αποφυγή αφού οι επιδόσεις της είναι χειρότερες ακόμα και από μια baseline προσέγγιση, η οποία απλά θα διάλεγε για κάθε πρότυπο ένα label στην τύχη με ίση πιθανότητα. Επιβεβαιώνουν ταυτόχρονα και την αρνητική επίδραση που έχει η εφαρμογή της περιστροφής στα αρχικά πρότυπα, υποδεικνύοντας ότι η γνώση του "μοντέλου" (γενικά, της φυσικής διαδικασίας) που παρήγαγε τα δεδομένα είναι σημαντικός παράγοντας για την απόδοση ενός Tropical Perceptron.

4.2 Πολυωνυμικός Αλγόριθμος Ανάθεσης Βαρών

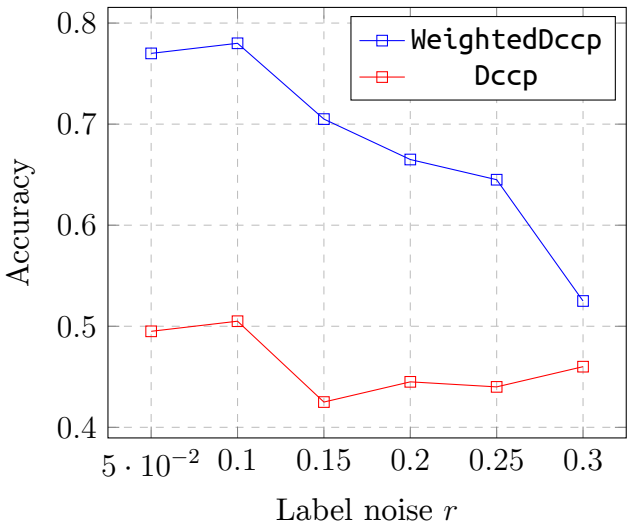
Για την περίπτωση της δυαδικής κατηγοριοποίησης, μπορούμε να κατασκευάσουμε έναν αλγόριθμο που να αποφαίνεται εάν τα πρότυπα είναι τροπικώς γραμμικά διαχωρίσιμα σε πολυωνυμικό χρόνο.

Παρατηρούμε ότι σε κάθε διάσταση d, το τροπικό perceptron δημιουργεί τους ημιχώρους που ορίζονται από $w_d + x_d \lesseqgtr 0$. Για $\mathbf{x}^{(k)} \in \mathcal{C}_1$, αρκεί να ισχύει $w_d + \mathbf{x}_d^{(k)} \geq 0$ για ένα οποιοδήποτε d, ώστε $\max_d\{\mathbf{x}_d^{(k)} + w_d\} \geq 0$. Αντίθετα, για $\mathbf{x}^{(l)} \in \mathcal{C}_0$, πρέπει να ισχύει

$$\mathbf{x}_d^{(l)} + w_d \leq 0, \forall d \Leftrightarrow \max_d\{\mathbf{x}_d^{(l)} + w_d\} \leq 0 \tag{4.18}$$

Οι όροι του max-αθροίσματος στο δεξί μέλος της (4.18) λαμβάνουν τη μεγαλύτερη τιμή τους σε κάθε διάσταση d για

$$\max_{\mathbf{x}^{(l)} \in \mathcal{C}_0} \{\mathbf{x}_d^{(l)} + w_d\} = w_d + \max_{\mathbf{x}^{(l)} \in \mathcal{C}_0} \mathbf{x}_d^{(l)}$$



Σχήμα 4.7: Ακρίβεια μεθόδου με και χωρίς ομαλοποίηση για συνθετικά δεδομένα, περιστραμμένα με τυχαίο τρόπο

Επομένως, αρκεί να προσδιορίσουμε τη μεγαλύτερη συντεταγμένη των αρνητικών προτύπων ανά διάσταση j, $\overline{x}_j$ και να ελέγξουμε ότι κάθε θετικό πρότυπο $\mathbf{x}^{(k)}$ διαθέτει τουλάχιστον μια συντεταγμένη $\mathbf{x}_d^{(k)} > \overline{x}_d$. Ο ψευδοκώδικας περιγράφεται στον αλγόριθμο 6.
Η ανάλυση του χρόνου εκτέλεσης του αλγορίθμου 6 είναι απλή:

- Η συνάρτηση FindLargest διατρέχει όλες τις συντεταγμένες όλων των προτύπων $\in \mathcal{C}_0$ πραγματοποιώντας μια μεγιστοποίηση (σταθερός χρόνος) σε κάθε βήμα, επομένως απαιτεί χρόνο $T = \Theta(n \cdot d)$, όπου n ο αριθμός των προτύπων και d η διαστατικότητα.
- Η DecideSeparable διατρέχει όλες τις συντεταγμένες των προτύπων $\in \mathcal{C}_1$ και σε κάθε βήμα πραγματοποιεί μια σύγκριση σε σταθερό χρόνο, αφού χρησιμοποιεί τον προϋπολογισμένο πίνακα `maxCoords`. Συνεπώς και αυτή έχει χρόνο $T' = \Theta(n \cdot d)$

Με βάση τα παραπάνω, ο χρόνος εκτέλεσης του αλγορίθμου 6 είναι

$$T_{\text{total}} = \Theta(n \cdot d)$$

δηλαδή πολυωνυμικός ως προς το πλήθος των προτύπων εκπαίδευσης αλλά και της διάστασης των δεδομένων. Μπορούμε να χρησιμοποιήσουμε την ιδέα του προκειμένου να υπολογίσουμε μια ανάθεση βαρών w_i, ορίζοντας αρχικά:

$$x_j^* = \min_{\substack{\mathbf{x}^k \in \mathcal{C}_1: \\ \mathbf{x}_j^k > \texttt{maxCoords}[j]}} \mathbf{x}_j^k$$

Ουσιαστικά το $\mathbf{x}^*$ είναι το "περιθώριο" που υπάρχει ανά διάσταση μεταξύ των 2 πλησιέστερων προτύπων των 2 κλάσεων. Έτσι, μπορούμε να θέσουμε

$$w_j = -(\texttt{maxCoords}[j] + x_j^*)$$

Algorithm 6 Εύρεση συνόλου τροπικώς διαχωρίσιμων προτύπων

function FINDLARGEST($X \in \mathbb{R}^{n\times d}$)
  `maxCoords`$[1 \dots d] \leftarrow -\infty$
  for $\mathbf{x}^{(k)} \in \mathcal{C}_0$ **do**
    for $j \in 1 \dots d$ **do**
      `maxCoords`$[j] \leftarrow \max(\mathbf{x}_j^{(k)}$, `maxCoords`$[j])$
    end for
  end for
  return `maxCoords`
end function

function DECIDESEPARABLE($X \in \mathbb{R}^{n\times d}$)
  `maxCoords` $\leftarrow$ FINDLARGEST(X)
  for $\mathbf{x}^{(k)} \in \mathcal{C}_1$ **do**
    if $\exists j : \mathbf{x}_j^{(k)} \leq$ `maxCoords`$[j]$ **then**
      return `False`
    end if
  end for
  return `True`
end function

Ο αλγόριθμος που παρουσιάστηκε επιλύει το πρόβλημα του προσδιορισμού ενός συνόλου παραμέτρων w_i έτσι ώστε να ικανοποιεί τις συνθήκες

$$\begin{cases} \bigvee\limits_i \mathbf{x}_i^k + w_i \geq 0 & \forall\ \mathbf{x}^k \in \mathcal{C}_1 \\ \bigvee\limits_i \mathbf{x}_i^k + w_i \leq 0 & \forall\ \mathbf{x}^k \in \mathcal{C}_0 \end{cases}$$

Έτσι, ο αλγόριθμος απαντά στο ερώτημα που επισημαίνεται στην ενότητα 4.1.2, για το τροπικό πολύεδρο (16) που περιγράφεται από τα εξής:

$$\begin{aligned} \boldsymbol{A} &= [\mathbf{x}^{i_1}, \dots, \mathbf{x}^{i_k}]\ \forall\ \mathbf{x}^i \in \mathcal{C}_1 \\ \boldsymbol{B} &= [\mathbf{x}^{j_1}, \dots, \mathbf{x}^{j_l}]\ \forall\ \mathbf{x}^j \in \mathcal{C}_0 \\ \boldsymbol{A} \boxplus \boldsymbol{w} &\geq 0 \\ \boldsymbol{B} \boxplus \boldsymbol{w} &\leq 0 \end{aligned}$$

όπου $\boldsymbol{A} \in \mathbb{R}^{k\times d}, \boldsymbol{B} \in \mathbb{R}^{l\times d}$ οι πίνακες με γραμμές τα διανύσματα που περιγράφουν τα πρότυπα των κλάσεων $\mathcal{C}_1, \mathcal{C}_0$ αντίστοιχα. Είναι εύκολο να διαπιστώσουμε ότι οι περιπτώσεις όπου $\boldsymbol{c}, \boldsymbol{d} \neq 0$ μπορούν

να αναχθούν στον αλγόριθμο 6. Το πρόβλημα προς επίλυση είναι

$$\begin{cases} \bigvee_i \boldsymbol{a}_i^k + w_i \geq c_k & \forall k \\ \bigvee_i \boldsymbol{b}_i^l + w_i \leq d_l & \forall l \end{cases}$$

το οποίο είναι ισοδύναμο με το πρόβλημα

$$\begin{cases} \bigvee_i -c_k + \boldsymbol{a}_i^k + w_i \geq 0 & \forall k \\ \bigvee_i -d_l + \boldsymbol{b}_i^l + w_i \leq 0 & \forall l \end{cases}$$

Συμβολίζοντας με $\boldsymbol{A}_{i,\cdot}$ την i-οστή γραμμή του πίνακα $\boldsymbol{A}$, θέτουμε

$$\boldsymbol{A}'_{k,\cdot} = \boldsymbol{A}_{k,\cdot} - c_k, \boldsymbol{B}'_{l,\cdot} = \boldsymbol{B}_{l,\cdot} - d_l$$

καταλήγοντας στο πρόβλημα

$$\begin{cases} \boldsymbol{A}' \boxplus \boldsymbol{w} & \geq 0 \\ \boldsymbol{B}' \boxplus \boldsymbol{w} & \leq 0 \end{cases}$$

το οποίο φυσικά λύνεται με τον αλγόριθμο 6. Έτσι, έχουμε περιγράψει έναν αλγόριθμου που αποφαίνεται εάν ένα τροπικό πολύεδρο όπως περιγράφεται στον ορισμό 16 είναι μη-κενό (και υπολογίζει μια ανάθεση τιμών).

Πόρισμα 1. Το πρόβλημα της διαχωρισιμότητας ενός συνόλου δεδομένων $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ από ένα τροπικό perceptron της μορφής (4.2) είναι επιλύσιμο σε πολυωνυμικό χρόνο.

Μπορούμε να εξετάσουμε την μαθηματική θεμελίωση του παραπάνω αλγορίθμου με βάση τις λύσεις που δίνει ο λεγόμενος *αντίστροφος Cuninghame-Green*. Στην περίπτωσή μας, αφού η κλάση $\mathcal{C}_0$ πρέπει να αντιστοιχεί σε πρότυπα $\mathbf{x}^k$ τέτοια ώστε $y_k < 0$, αναζητούμε τη λύση της ανίσωσης

$$\boldsymbol{X}_{\text{neg}} \boxplus w \leq \mathbf{0}_{1 \times K} \tag{4.19}$$

όπου $\boldsymbol{X}_{\text{neg}} \in \mathbb{R}^{K \times N}$ ένας πίνακας του οποίου οι γραμμές είναι τα πρότυπα της κλάσης $\mathcal{C}_0$. Με βάση την πρόταση 1, ξέρουμε ότι η μεγαλύτερη λύση της είναι η $\overline{w}$, για την οποία ισχύουν

$$\overline{w} = -\boldsymbol{X}_{\text{neg}}^T \boxplus' \mathbf{0}_{1 \times K}$$
$$\boldsymbol{X}_{\text{neg}} \boxplus \overline{w} \leq \mathbf{0}_{1 \times K}$$

Έχοντας προσδιορίσει τη λύση $\overline{w}$, το ζητούμενο είναι να δούμε εάν ικανοποιεί και το ζητούμενο για τα "θετικά" πρότυπα, δηλαδή εάν ισχύει

$$\boldsymbol{X}_{\text{pos}} \boxplus \overline{w} \geq \mathbf{0}_{1 \times L}$$

όπου εδώ $\boldsymbol{X}_{\text{pos}}$ ορίζεται ανάλογα με το $\boldsymbol{X}_{\text{neg}}$ για την κλάση $\mathcal{C}_1$. Συνδυάζοντας τις παραπάνω σχέσεις, έχουμε τη συνθήκη

$$\begin{aligned} \boldsymbol{X}_{\text{pos}} \boxplus \left(-\boldsymbol{X}_{\text{neg}}^{T} \boxplus' \mathbf{0}_{1\times K}\right) &\geq \mathbf{0}_{1\times L} \\ \boldsymbol{X}_{\text{pos}} \boxplus \left(\boldsymbol{X}_{\text{neg}}^{\sharp} \boxplus' \mathbf{0}_{1\times K}\right) &\geq \mathbf{0}_{1\times L} \end{aligned} \tag{4.20}$$

Στο [1], οι συγγραφείς αντιμετωπίζουν τους πίνακες $\boldsymbol{A}, \boldsymbol{B}$ ως *πυρήνες* (kernels) των πράξεων $\boxplus, \boxplus'$, τις οποίες αντιμετωπίζουν αντίστοιχα ως *τελεστές* (operators), για τους οποίους ισχύει

$$\begin{aligned} (Bx)_i &:= \bigvee_{j\in N} (B_{ij} + x_j) \\ (A^{\sharp}x)_j &:= \bigwedge_{i\in M} (-A_{ij} + x_i) \end{aligned}$$

Για την ανίσωση $\boldsymbol{A} \boxplus \mathbf{x} \leq \boldsymbol{B} \boxplus \mathbf{x}$, οι συγγραφείς συσχετίζουν τον τελεστή $f := A^{\sharp}Bx$, καθώς ισχύει

$$A \boxplus \mathbf{x} \leq B \boxplus \mathbf{x} \leftrightarrow x \leq f(x)$$

ενώ ο τελεστής f χρησιμοποιείται ως τελεστής δυναμικού προγραμματισμού για τα mean payoff games. Στην περίπτωσή μας, έχουμε καταλήξει με φυσικό τρόπο στον τελεστή

$$g := X_{\text{pos}} X_{\text{neg}}^{\sharp} x$$

αφού η συνθήκη (4.20) γράφεται και ως $g(\mathbf{0}_{1\times K}) \geq 0$.

4.3 Μη-γραμμικότητα: η συνάρτηση ReLU

Στην ενότητα αυτή, θα δείξουμε πώς η συνάρτηση ReLU (rectified linear unit) που επιλέγεται ευρέως ως συνάρτηση ενεργοποίησης σε βαθιές αρχιτεκτονικές νευρωνικών δικτύων [47] μπορεί να προκύψει από μια άλλη δημοφιλή επιλογή, τη συνάρτηση softplus [19], χρησιμοποιώντας μεθόδους τροπικής άλγεβρας. Μια παρόμοια διαδικασία με αυτή που θα χρησιμοποιήσουμε ονομάζεται *αποκβαντισμός Maslov* (Maslov Dequantization) και μια επισκόπηση για τη σχέση της με την τροπική γεωμετρία μπορεί να βρεθεί στο [38].

Ορισμός 17 (Dequantization Transform [38])**.** Συμβολίζουμε με $\mathbb{R}_{+}^{n}$ το σύνολο $\{(x_1, x_2, \ldots x_n) \in X : x_i \geq 0\}$, όπου X ένας τοπολογικός χώρος (συνήθως $X = \mathbb{R}^n$ ή $X = \mathbb{C}^n$). Έστω $\mathcal{F}(\mathbb{C}^n)$ το σύνολο των συναρτήσεων που είναι ορισμένες και συνεχείς σε ένα πυκνό ανοικτό υποσύνολο $U \subset \mathbb{C}^n$ ώστε $\mathbb{R}_{+}^{n} \subset U$. Για $f \in \mathcal{F}(\mathbb{C}^n)$, ορίζουμε τη συνάρτηση

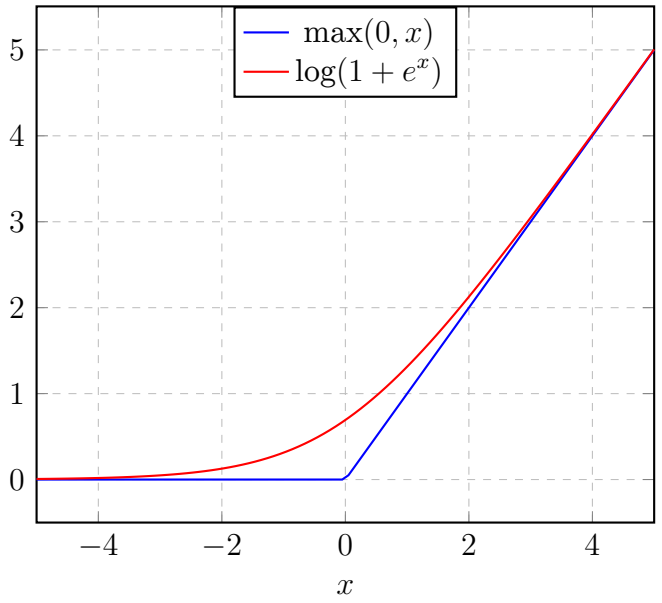
$$f_h(x) = h \cdot \log \left| f\left(e^{x/h}\right) \right| \tag{4.21}$$

και ακόμη

$$\hat{f}(x) = \lim_{h\to 0} f_h(x) \tag{4.22}$$

Η συνάρτηση $\hat{f}$ λέγεται **αποκβαντισμός** της f και η απεικόνιση $f \longmapsto \hat{f}$ **μετασχηματισμός αποκβαντισμού**.

Η συνάρτηση ενεργοποίησης *softplus* είναι μια εναλλακτική που προτάθηκε από τους Dugas et al. [19] για την αντικατάσταση της σιγμοειδούς συνάρτησης ενεργοποίησης.



Σχήμα 4.8: $\zeta(x) = \log(1 + e^x), \zeta_h(x) = \lim_{h \to 0^+} h\zeta(x/h)$

Ορισμός 18 (Συνάρτηση softplus [19])**.** Έστω $h(x) = \dfrac{1}{1 + e^{-x}}$. Η παράγουσά της, $\zeta(x)$, δίνεται από τον τύπο

$$\zeta(x) = \log(1 + e^x) \tag{4.23}$$

καθώς ισχύει

$$\begin{aligned} \zeta'(x) &= \frac{d}{dx}[\log(1 + e^x)] \\ &= \frac{1}{1 + e^x} \cdot \frac{d}{dx}(1 + e^x) \\ &= \frac{e^x}{1 + e^x} \\ &\overset{\times e^{-x}}{=} \frac{1}{1 + e^{-x}} = h(x) \end{aligned}$$

και θα αναφερόμαστε σε αυτήν με την ονομασία *softplus*.

Παρατηρούμε ότι η κλιμακούμενη εκδοχή της συνάρτησης softplus, δηλαδή

$$\zeta_h(x) = h\zeta(x/h) = h \log(1 + e^{x/h})$$

είναι η παράγουσα της $\phi(x) = \dfrac{1}{1+e^{-x/h}}$. Έτσι, έχουμε το εξής απλό πόρισμα:

Πρόταση 10. Για $h \to 0$, έχουμε $\zeta_h(x) \to \text{relu}(x) = \max\{0, x\}$.

Απόδειξη. Έχουμε αναλυτικά

$$\begin{aligned}
\lim_{h\to 0^+} h\log(1+e^{x/h}) &= \lim_{h\to 0^+} \frac{\log(1+e^{x/h})}{1/h} \\
&= \lim_{h\to 0^+} \frac{\frac{1}{1+e^{-x/h}} \cdot (-x/h^2)}{\frac{-1}{h^2}} \qquad (4.24) \\
&= \lim_{h\to 0^+} \frac{x}{1+e^{-x/h}} \qquad (4.25)
\end{aligned}$$

όπου στην σχέση (4.24) κάναμε χρήση της ιδιότητας de l'Hospital, παίρνοντας παραγώγους ως προς h. Στη σχέση (4.25), είναι ανάγκη να πάρουμε περιπτώσεις.

- $x > 0$: ισχύει $\lim_{h\to 0^+} x/h = \infty$ και άρα

  $$\lim_{h\to 0^+} \frac{1}{1+e^{-x/h}} = \frac{1}{1+e^{-\infty}} = 1$$

  οπότε το αρχικό όριο είναι

  $$\lim_{h\to 0^+} \frac{x}{1+e^{-x/h}} = x$$

- $x < 0$: ισχύει $\lim_{h\to 0^+} x/h = -\infty$ άρα

  $$\lim_{h\to 0^+} \frac{x}{1+e^{-x/h}} = \frac{x}{1+e^{\infty}} = 0$$

Οπότε, συνυπολογίζοντας την περίπτωση $x = 0$ όπου έχουμε

$$\lim_{h\to 0^+} h \cdot \log(1+e^0) = \lim_{h\to 0^+} h \cdot \log 2 = 0$$

τελικά ισχύει

$$\lim_{h\to 0^+} h\log(1+e^{x/h}) = \max\{0, x\} = x \oplus 0 \qquad (4.26)$$

□

Σημείωση. Η συνάρτηση softplus εμφανίζεται στην συνάρτηση ενέργειας των μηχανών Boltzmann (RBMs). Στο [12], οι Cueto & Sturmfels χρησιμοποιούν τη μετάβαση στην τροπική γεωμετρία μέσω του Maslov Dequantization προκειμένου να εξάγουν αποτελέσματα σχετικά με τη διάσταση των παραμέτρων μιας μηχανής Boltzmann, εξετάζοντας το τροπικό ανάλογό της.

4.4 Τροπικές Μηχανές Διανυσμάτων Υποστήριξης

Σε ένα πρόσφατο technical report, οι Gärtner και Jaggi [22] πρότειναν ένα μοντέλο, παρόμοιο με αυτό του τροπικού perceptron, ως αντικατάσταση των "παραδοσιακών" μηχανών διανυσμάτων υποστήριξης (με χρήση πυρήνων (kernels) ή μη). Αν και η έξοδος που υπολογίζει είναι όμοια με αυτή του τροπικού perceptron, οι περιοχές αποφάσεων που προτείνονται δεν εξαρτώνται από την τιμή της εξόδου αλλά από τον όρο της ενεργοποίησης που τη μεγιστοποιεί.

4.4.1 Τροπικές καμπύλες

Μια τροπική πολυωνυμική συνάρτηση $p : \mathbb{R}^n \to \mathbb{R}$ δίνεται ως το ανά-όρο μέγιστο (ή ελάχιστο) ενός πεπερασμένου συνόλου από γραμμικές ή αφφινικές συναρτήσεις. Κατ' αντιστοιχία με τις υπερεπιφάνειες της (κλασσικής) αλγεβρικής γεωμετρίας, που ορίζονται ως οι ρίζες ενός πολυωνύμου ή ενός συστήματος πολυωνυμικών εξισώσεων, ορίζεται και η έννοια της *τροπικής υπερεπιφάνειας*:

Ορισμός 19 (Tropical Hypersurface)**.** Έστω $p : \mathbb{R}^n \to \mathbb{R}$ ένα τροπικό πολυώνυμο n μεταβλητών. Ορίζουμε την υπερεπιφάνεια $V(p)$ ως το σύνολο

$$V(p) \triangleq= \{\boldsymbol{w} \in \mathbb{R}^n : p(\mathbf{w}) \text{ δεν είναι παραγωγίσιμο } \}$$

Τα σημεία αυτά ονομάζονται οι ρίζες του p.

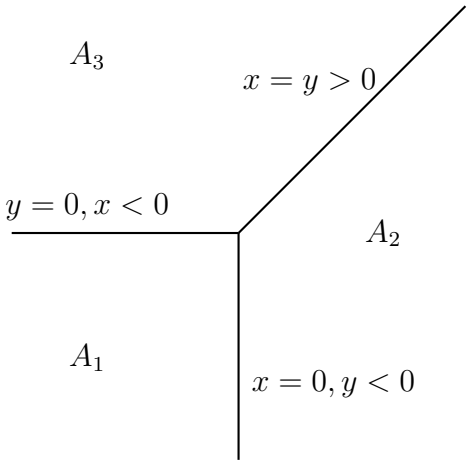
Πρακτικά, η $V(p)$ περιλαμβάνει τα σημεία όπου η $p(\mathbf{x})$ ως συνάρτηση μεγιστοποιείται από περισσότερους από έναν όρους. Η $V(p)$ περιγράφει μια τροπική καμπύλη, και στην ειδική περίπτωση των 2 μεταβλητών x, y ονομάζεται τροπική καμπύλη στο επίπεδο (plane tropical curve). Το πλέον διαδεδομένο παράδειγμα είναι η τροπική καμπύλη της

$$p(x, y) = \max\{x, y, 0\} \tag{4.27}$$

Στην εικόνα 4.9, η καμπύλη απεικονίζει τα σημεία του $\mathbb{R}^2$ στα οποία το μέγιστο της $p(x, y)$ δεν αντιστοιχεί σε μοναδικό όρο. Μια σημαντική παρατήρηση είναι το γεγονός ότι η τροπική καμπύλη της εικόνας έχει διαμερίσει τον $\mathbb{R}^2$ σε 3 περιοχές:

$$A_1 := \{(x, y) : 0 \geq x, 0 \geq y\} \tag{4.28}$$
$$A_2 := \{(x, y) : x \geq 0, x \geq y\} \tag{4.29}$$
$$A_3 := \{(x, y) : y \geq 0, y \geq x\} \tag{4.30}$$

Η παρατήρηση αυτή γενικεύεται ως ιδιότητα για τον λεγόμενο *τροπικό προβολικό χώρο*, τον οποίο ορίζουμε ως εξής:



Σχήμα 4.9: Τροπική καμπυλη της Εξ. (4.27)

Ορισμός 20 (Τροπικός Προβολικός Χώρος)**.** Ο τροπικός προβολικός χώρος $\mathbb{TP}^n$ είναι ο χώρος $\mathbb{R}^n_{\max}/\{-\infty\}^n$ εφοδιασμένος με τη σχέση ισοδυναμίας $\sim$, σύμφωνα με την οποία

$$(x_0, x_1, \dots x_n) \sim \lambda + (x_0, x_1, \dots x_n), \quad \lambda \in \mathbb{R} \tag{4.31}$$

Κατ' αντιστοιχία με τον προβολικό χώρο της κλασσικής προβολικής γεωμετρίας, στον οποίο τα σημεία κανονικοποιούνται έτσι ώστε η n-οστή συντεταγμένη να είναι μοναδιαία, στον τροπικό προβολικό χώρο τα σημεία κανονικοποιούνται ώστε αυτή να είναι μηδενική.

Επαναδιατυπώνουμε την παρατήρηση της εικόνας 4.9 ως πρόταση για τον τροπικό προβολικό χώρο $\mathbb{TP}^d$.

Πρόταση 11. Έστω η τροπική γραμμική παράσταση στον $\mathbb{R}^n_{\max}$:

$$\boldsymbol{w}^T \boxplus \mathbf{x} = \max\{w_1 + x_1, \dots w_n + x_n\}$$

Η τροπική υπερεπιφάνεια που αντιστοιχεί διαμερίζει τον τροπικό προβολικό χώρο $\mathbb{TP}^{d-1}$ σε d συνεκτικές περιοχές, οι οποίες λέγονται ανοικτοί τομείς (*open sectors*) και ορίζονται ως

$$\mathcal{R}_j = \{\mathbf{x} \in \mathbb{TP}^{d-1} : w_j + x_j > w_i + x_i, \ \forall i \neq j\} \tag{4.32}$$

Ακόμη, οι περιοχές $\mathcal{R}_j$ είναι κυρτές.

Το γεγονός ότι οι ανοικτοί τομείς της εξίσωσης (4.32) περιγράφουν κυρτά σύνολα είναι πάρα πολύ σημαντικό, αφού θα χρησιμοποιηθεί για να εκφράσουμε τον αλγόριθμο μάθησης μιας τροπικής μηχανής διανυσμάτων υποστήριξης ως πρόβλημα κυρτής βελτιστοποίησης. Τα προβλήματα κυρτής βελτιστοποίησης διαθέτουν ολικό ελάχιστο [8] και αποτελούν ένα από τα καλύτερα μελετημένα προβλήματα από αλγοριθμικής απόψεως.

Το μοντέλο TA-SVM

Στο report των Gärtner & Jaggi [22], επισημαίνεται ότι η διαμέριση που επάγεται από μια τροπική υπερεπιφάνεια και περιγράφεται στην Πρόταση (11) προσφέρει ένα φυσικό τρόπο για d-μερή κατηγοριοποίηση. Στην περίπτωση που οι τομείς κάθε κατηγορίας είναι μοναδικοί και γνωστοί a priori, το πρόβλημα αποτελεί ένα πρόβλημα γραμμικού προγραμματισμού (LP) και λύνεται αποδοτικά. Στην περίπτωση που αυτό δεν ισχύει, το πρόβλημα πλέον είναι ένα στιγμιότυπο Ελεγχόμενου Γραμμικού Προγραμματισμού (CLP) [5] και εν γένει μη-κυρτό. Μπορούμε να παρακάμψουμε αυτό το πρόβλημα εισάγοντας μια επιπλέον παράμετρο στην έξοδο που υπολογίζει ο ταξινομητής μας, η οποία θα είναι ένας γραμμικός μετασχηματισμός. Θα αναφερόμαστε στο προτεινόμενο μοντέλο με την ονομασία τροπική μηχανή αφφινικών διανυσμάτων υποστήριξης (Tropical Affine Support Vector Machine), ή εν συντομία TA-SVM. Το μοντέλο περιγράφεται λεπτομερώς παρακάτω.

Ορισμός 21 (TA-SVM)**.** Έστω ένα σύνολο n-διάστατων δεδομένων $\boldsymbol{X} \in \mathbb{R}^{K\times n}$, για τα οποία είναι γνωστό ότι ανήκουν σε d κλάσεις/κατηγορίες. Η τροπική μηχανή αφφινικών διανυσμάτων υποστήριξης με παραμέτρους $\boldsymbol{W} \in \mathbb{R}^{d\times n}$ και $\boldsymbol{b} \in \mathbb{R}^d$ υπολογίζει την παράσταση

$$y(\mathbf{x}) = \bigvee_{i=1}^{d} \boldsymbol{W}_i^T \mathbf{x} + b_i \tag{4.33}$$

στην οποία $\mathbf{x}$ αποτελεί ένα πρότυπο εισόδου και $\boldsymbol{W}_i$ είναι το διάνυσμα-στήλη που αντιστοιχεί στην i-οστή γραμμή του πίνακα $\boldsymbol{W}$. Οι περιοχές αποφάσεων ορίζονται ως

$$\mathcal{R}_j = \{\mathbf{x} \in \mathbb{TP}^{d-1} : \boldsymbol{W}_j^T \mathbf{x} + b_j \geq \boldsymbol{W}_i^T \mathbf{x} + b_i,\ \forall i \neq j\} \tag{4.34}$$

και το πρότυπο $\mathbf{x}$ ανατίθεται στην j κατηγορία αν ανήκει στην περιοχή $\mathcal{R}_j$.

Μια ισοδύναμη γραφή για τις περιοχές $\mathcal{R}_j$ της εξίσωσης (4.34) είναι η

$$\boldsymbol{W}_j^T \mathbf{x} + b_j \geq \max_i \{\boldsymbol{W}_i^T \mathbf{x} + b_i\} \tag{4.35}$$

$$\max_i \{\boldsymbol{W}_i^T \mathbf{x} + b_i\} - \left(\boldsymbol{W}_j^T \mathbf{x} + b_j\right) \leq 0 \tag{4.36}$$

στην οποία είναι εύκολο να δούμε ότι αποτελεί μια ανισότητα της μορφής $f_j(\mathbf{x}) \leq 0$, όπου f_j κυρτή. Ο Ορισμός 21 καλύπτει και το προτεινόμενο μοντέλο του [22], αφού εκείνο αντιστοιχεί στην περίπτωση $d = n, \boldsymbol{W}_i = \boldsymbol{I}_{n\times n}$. Δεν είναι δύσκολο να διαπιστώσουμε ότι σε αυτή την περίπτωση οι διαμερίσεις του χώρου που επάγονται είναι και πάλι ένα τροπικό πολύεδρο / πολύτοπο. Το ακριβώς ισχύει και για το σύνολο των επιτρεπτών διανυσμάτων $\boldsymbol{w} \in \mathbb{R}^n_{\max}$, σε περίπτωση που το σύνολο των προτύπων $\boldsymbol{X} \in \mathbb{R}^{k\times n}_{\max}$ είναι διαχωρίσιμο. Χωρίζοντας το $\boldsymbol{X}$ στους πίνακες

$$\mathcal{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$$

το σύνολο των ανισοτήτων που ορίζει ένα διάνυσμα $\boldsymbol{w}$ που ταξινομεί τα πρότυπα σωστά είναι το

$$\boldsymbol{X}_1 \boxplus \boldsymbol{w} \leq (x_1, -\infty, \ldots, -\infty) \boxplus \boldsymbol{w} \tag{4.37}$$

$$\boldsymbol{X}_2 \boxplus \boldsymbol{w} \leq (-\infty, x_2, \ldots, -\infty) \boxplus \boldsymbol{w} \tag{4.38}$$

$$\vdots \tag{4.39}$$

$$\boldsymbol{X}_n \boxplus \boldsymbol{w} \leq (-\infty, -\infty, \ldots, x_n) \boxplus \boldsymbol{w} \tag{4.40}$$

Ενώνοντας τους πίνακες στα αριστερά και δεξιά μέλη των παραπάνω ανισοτήτων, καταλήγουμε σε ένα σύνολο της μορφής

$$\boldsymbol{X} \boxplus \boldsymbol{w} \leq \boldsymbol{C}_k \boxplus \boldsymbol{w}$$

το οποίο περιγράφει έναν *τροπικό κώνο*, κατ' αντιστοιχία με τον ορισμού ενός κώνου

$$K = \{\mathbf{x} \in \mathbb{R}^n : \boldsymbol{A} \cdot \mathbf{x} \leq \boldsymbol{B} \cdot \mathbf{x}\}$$

στην Ευκλείδεια Γεωμετρία.

Στη γενική περίπτωση, δε γνωρίζουμε αν τα πρότυπα εισόδου είναι διαχωρίσιμα με βάση τις περιοχές απόφασης $\mathcal{R}_j$, οπότε είναι σκόπιμο να εισάγουμε μεταβλητές χαλάρωσης που να επιτρέπουν misclassification για το πρόβλημά μας. Διατυπώνουμε εδώ το πλήρες πρόβλημα βελτιστοποίησης. Έστω $C = \{\mathcal{C}_1 \ldots \mathcal{C}_m\}$ το σύνολο των κλάσεων από τις οποίες προέρχονται τα πρότυπα εκπαίδευσης. Τότε το πρόβλημα μπορεί να γραφεί ως εξής:

$$\text{Minimize } J(\boldsymbol{W}, \boldsymbol{b}) = \sum_{k=1}^{K} \max\{\xi_k, 0\} \tag{4.41}$$

$$\text{s.t. } \left(\bigvee_{i=1}^{m} \boldsymbol{W}_i^T \mathbf{x}^{(k)} + b_i\right) - \left(\boldsymbol{W}_j^T \mathbf{x}^{(k)} + b_j\right) \leq \xi_k, \text{ if } \mathbf{x}^{(k)} \in \mathcal{C}_j, \ \forall k \in 1 \ldots K$$

Σύμφωνα με την παραπάνω διατύπωση

$$\left(\bigvee_{i=1}^{m} \boldsymbol{W}_i^T \mathbf{x}^{(k)} + b_i\right) - \left(\boldsymbol{W}_j^T \mathbf{x}^{(k)} + b_j\right) > 0 \Rightarrow \xi_k > 0 \tag{4.42}$$

ενώ σε διαφορετική περίπτωση, $\xi_k \leq 0$. Επομένως η μεταβλητή χαλάρωσης ξ_k θα συνεισφέρει στο κόστος $J(\boldsymbol{W}, \boldsymbol{b})$ μόνο εάν το πρότυπο $\mathbf{x}^{(k)}$ ανατεθεί σε λάθος κατηγορία. Παρατηρώντας τη μορφή του κόστους και των περιορισμών, συμπεραίνουμε:

Πρόταση 12. Η εκπαίδευση ενός TA-SVM μέσω της επίλυσης του προβλήματος (4.41) αποτελεί ένα πρόβλημα κυρτής βελτιστοποίησης.

Απόδειξη. Αρχικά, παρατηρούμε ότι κάθε περιορισμός αποτελείται από το άθροισμα μιας κυρτής και αφφινικών συναρτήσεων, καθώς η παράσταση

$$\left(\bigvee_{i=1}^{m} \boldsymbol{W}_i^T \mathbf{x}^{(k)} + b_i\right)$$

είναι το μέγιστο ενός συνόλου αφφινικών συναρτήσεων, και ως εκ τούτου κυρτή. Η παράσταση του κόστους

$$J = \sum_{k=1}^{K} \max\{\xi_k, 0\}$$

είναι το άθροισμα κυρτών συναρτήσεων, επομένως το πρόβλημα συνολικά βρίσκεται στη μορφή

$$\begin{aligned} &\text{Minimize } f_0(\mathbf{x}) \\ &\text{s.t. } f_i(\mathbf{x}) \leq 0 \end{aligned} \tag{4.43}$$

όπου f_i κυρτές συναρτήσεις. Η μορφή αυτή είναι η πρωτοτυπική μορφή ενός προβλήματος κυρτής βελτιστοποίησης. [8]. □

Η Πρόταση 12 είναι σημαντική, καθώς τα προβλήματα κυρτής βελτιστοποίησης αποτελούν μια κατηγορία προβλημάτων στην οποία το ολικό ελάχιστο μπορεί να βρεθεί με αποδοτικό τρόπο μέσω επαναληπτικών μεθόδων. Λόγω της μορφής του προβλήματος βελτιστοποίησης και συγκεκριμένα λόγω του τρόπου που εκμεταλλευόμαστε την έξοδο του TA-SVM, σε περίπτωση ενός πολυεπίπεδου νευρωνικού δικτύου το TA-SVM θα πρέπει αναγκαστικά να βρίσκεται στο επίπεδο εξόδου. Ένα βαθύ νευρωνικό δίκτυο αποτελούμενο από k κρυφά επίπεδα με κυρτές συναρτήσεις ενεργοποίησης (όπως είναι η ReLU) και ένα TA-SVM στο επίπεδο εξόδου, ορίζει μια συνάρτηση κόστους

$$J(\boldsymbol{W}) = J(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)}, \boldsymbol{W}^{(k+1)})$$

όπου $\{\boldsymbol{W}\}_1^k$ τα συναπτικά βάρη των k κρυφών επιπέδων και $\boldsymbol{W}^{(k+1)}$ τα συναπτικά βάρη του επιπέδου εξόδου. Σταθεροποιώντας τις τιμές όλων των συναπτικών βαρών εκτός από ενός συγκεκριμένου επιπέδου k, η J είναι κυρτή συνάρτηση ως προς το $\boldsymbol{W}^{(k)}$. Επομένως, μπορεί να ελαχιστοποιηθεί αποδοτικά με κάποια παραλλαγή alternating minimization, χωρίς ωστόσο να υπάρχει εγγύηση εύρεσης του ολικού ελαχίστου. Για παράδειγμα, στον Πίνακα 4.1, η μέθοδος εναλλασσόμενης ελαχιστοποίησης φαίνεται να δίνει σταθερά καλύτερα αποτελέσματα για ένα δίκτυο 2 επιπέδων (1 κρυφό + TA-SVM στην έξοδο) από ελαχιστοποίηση στο συνολικό δίκτυο χρησιμοποιώντας backpropagation[2]. Το μέγεθος του κρυφού επιπέδου συμβολίζεται με n_h, το πλήθος των εναλλαγών εκπαίδευσης μεταξύ επιπέδων με a, οι εποχές εκπαίδευσης ανά επίπεδο με N. Για την εκπαίδευση με backpropagation, οι εποχές ορίζονται ίσες με $N_{\text{full}} = 2 \cdot a \cdot N$, ώστε να είναι συνολικά ίσες με τις εποχές της alternating μεθόδου. Στην περίπτωση της alternating μεθόδου, η εκπαίδευση μπορεί να επιταχυνθεί σημαντικά χρησιμοποιώντας κάποια μέθοδο που εκμεταλλεύεται την τοπική καμπυλότητα (curvature) της συνάρτησης κόστους, όπως η μέθοδος Newton.

Μια άλλη δυνατότητα είναι ο συνδυασμός πολλαπλών TA-SVM σε κάποια προσθετική/boosting μέθοδο, όπως η AdaBoost [21], όπου η δυνατότητα να εκπαιδεύουμε τα επιμέρους τμήματα του τελικού δικτύου αποδοτικά είναι σημαντική.

[2]Τονίζεται ότι το δίκτυο από το οποίο εξήχθησαν τα αποτελέσματα του Πίνακα 4.1 δεν έχει βελτιστοποιηθεί ως προς τις παραμέτρους εκπαίδευσης.

Πίνακας 4.1: Σύγκριση μεθόδων εκπαίδευσης, 2 layers, `MNIST` dataset

n_h	a	N	Full	Alternating
16	3	10	0.856 ± 0.001	$\mathbf{0.877} \pm 0.001$
32	3	10	0.856 ± 0.003	$\mathbf{0.881} \pm 0.001$
64	3	10	0.866 ± 0.002	$\mathbf{0.874} \pm 0.001$
16	4	10	0.840 ± 0.001	$\mathbf{0.885} \pm 0.001$
32	4	10	0.856 ± 0.003	$\mathbf{0.885} \pm 0.001$
64	4	10	0.859 ± 0.003	$\mathbf{0.875} \pm 0.001$

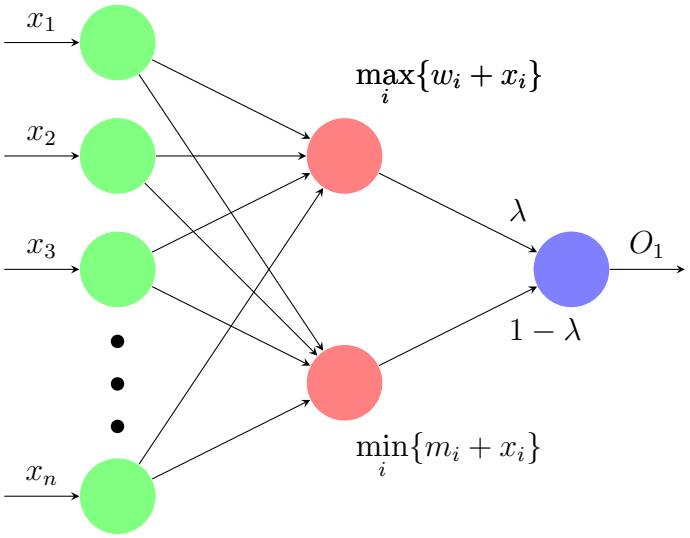
Κεφάλαιο 5

Μορφολογικά Νευρωνικά δίκτυα

5.1 Μορφολογικά νευρωνικά δίκτυα

5.1.1 Αρχιτεκτονική

Στην ενότητα αυτή αρχικά παρουσιάζουμε μια απλή αρχιτεκτονική ενός δικτύου που αποτελείται από 2 τροπικά perceptrons, εκ των οποίων το πρώτο στηρίζεται στη $(\max, +)$ και το δεύτερο στη $(\min, +)$ άλγεβρα, ενώ η συνολική έξοδος είναι το άθροισμα των 2 αποκρίσεων. Η αρχιτεκτονική του δικτύου αυτού φαίνεται συνοπτικά στο επόμενο σχήμα:



Σχήμα 5.1: Max-min tropical perceptron

Όπως φαίνεται στο σχήμα 5.1, δεδομένου ενός προτύπου εισόδου $\mathbf{x} \in \mathbb{R}^n$, το δίκτυο αυτό υπολογίζει 2 αποκρίσεις, εκ των οποίων η πρώτη είναι μια διαστολή (*dilation*) και η δεύτερη μιας συστολή (*erosion*) αντίστοιχα:

$$\delta(\mathbf{x}) = \bigvee_{i=1}^{n} w_i + x_i \tag{5.1}$$

$$\epsilon(\mathbf{x}) = \bigwedge_{i=1}^{n} m_i + x_i \tag{5.2}$$

Επιλέγοντας $y(\mathbf{x}) = a\delta(\mathbf{x}) + b\epsilon(\mathbf{x})$, έχουμε ένα "υβρίδιο" που αποτελείται από το γραμμικό συνδυασμό 2 μη-γραμμικών τελεστών. Για παράδειγμα, επιλέγοντας $a = 1, b = -1$ και κατάλληλα διανύσματα $\boldsymbol{w} = B^s, \boldsymbol{m} = B$, καταλήγουμε στην λεγόμενη μορφολογική παράγωγο (*morphological gradient* [60]), η οποία ορίζεται ως

$$\nabla^B I(\mathbf{x}) = (I \oplus B) - (I \ominus B)$$

και χρησιμοποιείται ευρέως στην όραση υπολογιστών, όπως για παράδειγμα ως τελεστής ανίχνευσης ακμών. Στη δικιά μας αντιμετώπιση, περιορίζουμε το μοντέλο μας, απαιτώντας η συνολική έξοδος να είναι ο κυρτός (convex) συνδυασμός των 2 αποκρίσεων, δηλαδή

$$y(\mathbf{x}) = \lambda\delta(\mathbf{x}) + (1-\lambda)\epsilon(\mathbf{x}) \tag{5.3}$$

Όπως φαίνεται, δε χρησιμοποιείται κάποια συνάρτηση ενεργοποίησης στην έξοδο των $(\max, \min)$ νευρώνων. Εναλλακτικά, μπορούμε να προσαρμόσουμε κάθε όρο των 2 perceptrons ώστε να αποτελείται από ένα *τροπικό μονώνυμο* της μορφής

$$w_i \vee x_i^{c_i} = w_i + c_i x_i$$

οπότε οι αποκρίσεις γίνονται

$$\delta(\mathbf{x}) = \bigvee_{i=1}^{n} w_i + c_i x_i \tag{5.4}$$

$$\epsilon(\mathbf{x}) = \bigwedge_{i=1}^{n} m_i + d_i x_i \tag{5.5}$$

5.1.2 Σχέση με νευρωνικά δίκτυα τύπου Maxout

Επιλέγοντας $\lambda = 0.5$, ουσιαστικά έχουμε ένα νευρωνικό δίκτυο το οποίο στην έξοδό του υπολογίζει το άθροισμα 2 τμηματικά γραμμικών συναρτήσεων εκ των οποίων η μεν είναι κυρτή $(\max, +)$ και η δε κοίλη $(\min, +)$. Καθώς ισχύει ότι f : convex $\Rightarrow -f$: concave , η έξοδος της (5.3) είναι η διαφορά 2 κυρτών συναρτήσεων, και γνωρίζουμε από [67, Theorem 2.1], [26, Prop 4.1] ότι κάθε τμηματικά συνεχής συνάρτηση μπορεί να γραφεί ως η διαφορά 2 τμηματικά συνεχών κυρτών συναρτήσεων της μορφής

$$h_i(\mathbf{x}) = \max_{j\in 1,2,\dots k}\{W_{ij}\mathbf{x} + b_{ij}\} = \bigvee_{j=1}^{k}\left[\left(\sum_{m=1}^{n} W_{ijm}x_m\right) + b_{ij}\right]$$

Με βάση το αποτέλεσμα αυτό οι Goodfellow et al. [26, Theorem 4.3] απέδειξαν ότι ένα δίκτυο maxout με 2 συναρτήσεις h_i είναι καθολικός προσεγγιστής (universal approximator) για κάθε συνεχή συνάρτηση f. Η περίπτωση του max-min perceptron δεν αποτελεί καθολικό προσεγγιστή, ωστόσο η επιλογή της προσθήκης ενός min-νευρώνα στην αρχιτεκτονική μας επηρεάζεται από αυτό το αποτέλεσμα. Συνολικά αποτελεί μια περίπτωση maxout δικτύου με

$$W_1 = \boldsymbol{I}_{n\times n}, W_2 = -\boldsymbol{I}_{n\times n}$$
$$b_{1j} = w_j, b_{2j} = -m_j$$

όπου $\boldsymbol{I}_{n\times n}$ ο $n \times n$ μοναδιαίος πίνακας. [1] Έτσι, ισχύει

$$z_{1j} = \mathbf{e}_j^T \mathbf{x} + w_j$$
$$z_{2j} = -(\mathbf{e}_j^T \mathbf{x} + m_j)$$

Ένα μελλοντικό πρόβλημα προς διερεύνηση που προκύπτει είναι το πώς να επιλέξουμε μη-διαγώνιους πίνακες W_1, W_2 (δηλαδή ουσιαστικά να γενικεύσουμε τους όρους του max/min perceptron) ώστε να συνδυάζουμε την εκφραστικότητα ενός γενικευμένου maxout δικτύου με μια αραιή αναπαράσταση.

Εφαρμογή σε προβλήματα regression

Μια εφαρμογή που προκύπτει συχνά στο πεδίο της αναγνώρισης προτύπων είναι η λεγόμενη παλινδρόμηση (πιο δόκιμα, *regression*), στην οποία ζητείται η εξαγωγή ενός μοντέλου που προβλέπει μια βαθμωτή τιμή, y, συναρτήσει μιας ή περισσότερων μεταβλητών εισόδου $(x_{i1}, \ldots, x_{ip})$. Υπάρχουν αρκετά μοντέλα παλινδρόμησης, εκ των οποίων τα πλέον γνωστά είναι:

- η γραμμική (linear regression): θεωρεί ότι οι μεταβλητές x_{ip} συνδέονται με την έξοδο y_i μέσω της σχέσης
$$y_i = b_0 + b_1 x_{i1} + \cdots + b_n x_{in} + \varepsilon_i \tag{5.6}$$
όπου $b_1, \ldots b_n$ οι συντελεστές του μοντέλου και ε_i ένας όρος που μοντελοποιεί το "θόρυβο" ως τυχαία μεταβλητή.

- η πολυωνυμική (polynomial regression): αποτελεί γενίκευση της απλής περίπτωσης της γραμμικής, στην οποία πλέον το μοντέλο θεωρείται πως είναι
$$y_i = b_0 + b_1 x^{a_1} + b_2 x^{a_2} + \ldots b_n x^{a_n} + \varepsilon_i \tag{5.7}$$

- η λογιστική (logistic regression): εφαρμόζεται κυρίως στην περίπτωση που η μεταβλητή y_i ειναι κατηγορική, δηλαδή εκφράζει το γεγονός
$$y_i = \begin{cases} 1 & i \in \mathcal{C} \\ 0 & \text{αλλιώς} \end{cases}$$
όπου $\mathcal{C}$ κάποια (γνωστή ή μη) κλάση/κατηγορία. Σε τέτοιου τύπου παλινδρόμηση, συνήθως αναζητούμε τις βέλτιστες παραμέτρους b_i ώστε να "ταιριάζουν" καλύτερα στο μοντέλο
$$y_i = \begin{cases} 1 & b_0 + b_1 x_{i1} + \cdots + \ldots b_n x_{in} + \varepsilon > 0 \\ 0 & \text{αλλιώς} \end{cases} \tag{5.8}$$

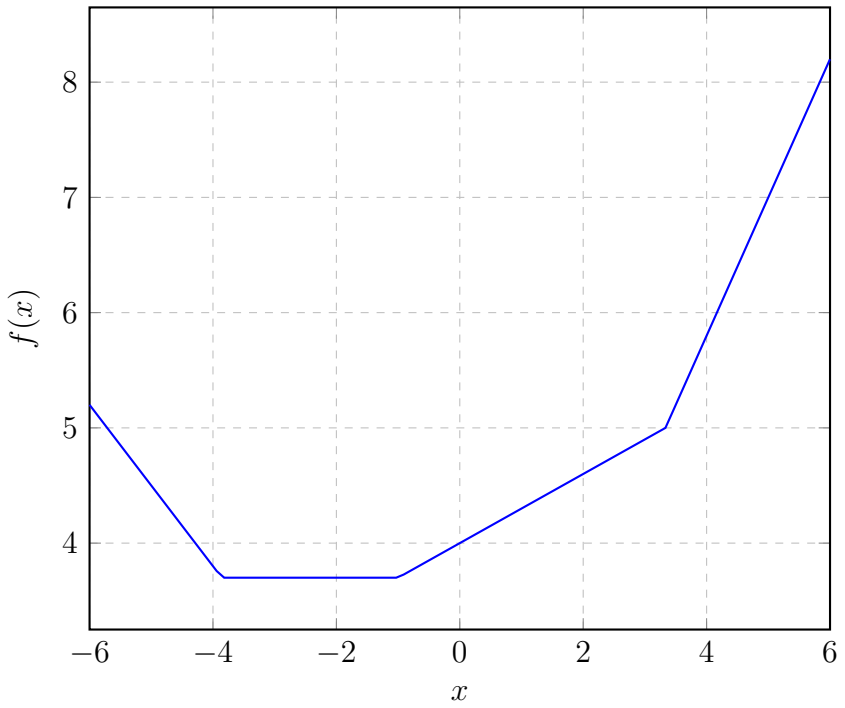
[1] Στην περίπτωση που οι όροι είναι της μορφής $w_i + c_i x_i$, οι πίνακες W_1, W_2 γίνονται

$$W_1 = \text{diag}(c_1, \ldots c_n)$$
$$W_2 = \text{diag}(d_1, \ldots d_n)$$

στο οποίο ε είναι και πάλι μια τυχαία μεταβλητή η οποία θεωρείται πως ακολουθεί τη λογιστική κατανομή, η οποία είναι

$$f(x;\mu,s) = \frac{e^{-(x-\mu)/s}}{s(1+e^{-(x-\mu)/s})^2}$$

Προσπαθώντας να διερευνήσουμε την "εκφραστικότητα" ενός max-min TP, θα δείξουμε ότι μπορεί να προσεγγίσει με αυθαίρετη ακρίβεια συναρτήσεις $f(x)$ οι οποίες είναι τμηματικά γραμμικές (piecewise linear, PWL εν συντομία).
Αρχικά, θα δείξουμε ότι κάθε κυρτή PWL συνάρτηση μιας μεταβλητής μπορεί να προσεγγιστεί με αυθαίρετη ακρίβεια από ένα tropical max-perceptron. Αν παρατηρήσουμε ένα max-πολυώνυμο, μπορούμε εύκολα να διαπιστώσουμε ότι αποτελεί το ίδιο μια κυρτή PWL συνάρτηση, όπως για παράδειγμα στην εικόνα 5.2.



Σχήμα 5.2: $f(x) = 3.7 \vee (-0.7x + 1) \vee (0.3x + 4) \vee (1.2x + 1)$

Γενικά, για μια PWL συνάρτηση ισχύει το εξής:

Πρόταση 13. Έστω $f : \mathbb{R} \to R$ μια κυρτή (αντ. κοίλη) τμηματικά γραμμική συνάρτηση η οποία αποτελείται από k γραμμικές περιοχές. Τότε η f γράφεται ισοδύναμα ως ένα max-πολυώνυμο (αντ. min-πολυώνυμο) αποτελούμενο από k όρους της μορφής

$$x^{\boxplus a_i} + b_i = a_i x + b_i \quad (x^{\boxplus k} \triangleq kx_i)$$

στο οποίο a_i είναι η κλίση (slope) του i-οστού τμήματος της f και τα b_i είναι προσαρμοσμένα έτσι ώστε οι ρίζες του max-πολυωνύμου να συμπίπτουν με τα σημεία συνάντησης των περιοχών $i, i+1$ της f.

Απόδειξη. Έστω το i-οστό γραμμικό τμήμα της συνάρτησης f, με όρια $[x_i, x_{i+1}]$ (προφανώς ισχύει $x_1 = -\infty, x_{k+1} = \infty$). Το τμήμα αυτό είναι κομμάτι μιας ευθείας που γράφεται ως $y_i(x) = a_i x + b_i$, δηλαδή έχει κλίση a_i και το σημείο τομής του με το τμήμα $i+1$ βρίσκεται ως εξής:

$$\begin{aligned} y_i(x) &= y_{i+1}(x) \\ a_i x + b_i &= a_{i+1} x + b_{i+1} \\ x(a_i - a_{i+1}) &= b_{i+1} - b_i \\ x &= \frac{b_{i+1} - b_i}{a_i - a_{i+1}} \end{aligned}$$

Καθώς μπορούμε να θεωρήσουμε χωρίς βλάβη της γενικότητας ότι $a_i \neq a_j, \forall\ j \neq i$, το παραπάνω σημείο τομής είναι καλώς ορισμένο. Όμως, ανατρέχοντας π.χ. στο [9, Chapter 5], αυτό είναι ακριβώς το σημείο τομής των αντίστοιχων όρων του max-πολυωνύμου. Επομένως, μπορούμε να αναπαραστήσουμε κάθε κυρτή PWL συνάρτηση με ένα max-πολυώνυμο. Αντίστοιχη είναι η απόδειξη για την περίπτωση που η f είναι κοίλη. □

Έχοντας αποδείξει την πρόταση 13, μπορούμε να δείξουμε το εξής:

Πρόταση 14. Αν $f(\cdot)$ μια τμηματικά γραμμική συνάρτηση στη μεταβλητή x, τότε μπορεί να προσεγγιστεί με αυθαίρετη ακρίβεια από ένα max-min tropical perceptron.

Απόδειξη. Γνωρίζουμε (βλ. π.χ. [67]) ότι κάθε PWL συνάρτηση μπορεί να γραφεί ως η διαφορά 2 κυρτών PWL συναρτήσεων (ή ισοδύναμα ως το άθροισμα μιας κυρτής και μιας κοίλης PWL συνάρτησης). Ωστόσο από τις προηγούμενες προτάσεις γνωρίζουμε ότι το max κομμάτι του perceptron μπορεί να εκφράσει οποιαδήποτε κυρτή PWL συνάρτηση (αντίστοιχα το min κομμάτι οποιαδήποτε κοίλη), επομένως η απόκριση $\delta(x) + \epsilon(x)$ του max-min tropical perceptron αποτελεί την ζητούμενη προσέγγιση. □

5.1.3 Γραμμικές περιοχές

Δεδομένου ενός νευρωνικού δικτύου που αποτελείται από (πιθανώς) πολλαπλά κρυφά επίπεδα, είναι ενδιαφέρον να διερευνήσουμε τον τρόπο με τον οποίο κάθε επίπεδο "διαχωρίζει" τον χώρο παραμέτρων στον οποίο επιδρά. Στην περίπτωση του (απλού) τροπικού perceptron, ο διαχωρισμός αυτός

γίνεται βάσει των γραμμικών περιοχών (regions of linearity) που προκύπτουν από τους όρους του max-αθροίσματος. Στη συνέχεια, θα χρησιμοποιήσουμε τη $(\max, +)$ εκδοχή του τροπικού perceptron για τα αποτελέσματά μας.

Πρόταση 15. Κάθε τροπικό Perceptron αποτελεί μια απεικόνιση $\phi : \mathbb{R}^n \to \mathbb{R}$ η οποία διαχωρίζει τον χώρο παραμέτρων σε n το πολύ γραμμικές περιοχές.

Απόδειξη. Έστω $f_i = c_i x_i + w_i$ ο $i-$οστός όρος του max-αθροίσματος. Αν για κάθε f_i υπάρχει κάποια περιοχή του χώρου παραμέτρων $S \subseteq R^n$ ώστε $f_i = \max_j f_j$, τότε το πλήθος των γραμμικών περιοχών του max-αθροίσματος $\bigvee_i f_i$ είναι ακριβώς n. □

Μια ακριβώς αντίστοιχη πρόταση ισχύει και για ένα min-TP. Εν γένει, για ένα (κρυφό) επίπεδο ενός νευρωνικού δικτύου το οποίο αποτελείται από max-TP (ή min-TP) με n εισόδους και m εξόδους, μπορούμε να εκμεταλλευτούμε το αποτέλεσμα των Montufar et. al. [46, Πρόταση 7] για να συμπεράνουμε:

Πρόταση 16. Ο μέγιστος αριθμός γραμμικών περιοχών για ένα επίπεδο αποτελούμενο από max (ή min) τροπικά perceptrons με n εισόδους και m εξόδους διαθέτει τουλάχιστον

$$n^{\min\{m,n\}}$$

και το πολύ

$$\min\left\{n^m, \sum_{j=0}^{n}\binom{n^2 m}{j}\right\}$$

γραμμικές περιοχές

Απόδειξη. Οι Montufar et. al [46] απέδειξαν ότι για ένα maxout δίκτυο με k όρους στο max-άθροισμα κάθε μονάδας, τα άνω και κάτω φράγματα για ένα επίπεδο με n εισόδους και m εξόδους είναι

$$k^{\min\{m,n\}}, \min\left\{k^m, \sum_{j=0}^{n}\binom{k^2 m}{j}\right\}$$

Στην περίπτωσή μας, $k = n$, οπότε προκύπτει το ζητούμενο με απλή αντικατάσταση. □

Τι συμβαίνει όμως στην περίπτωση ενός max-min TP; Σε αυτή την περίπτωση έχουμε τη διαφορά 2 κυρτών τμηματικά γραμμικών συναρτήσεων ή ισοδύναμα το άθροισμα 2 τμηματικά γραμμικών συναρτήσεων εκ των οποίων η μια κυρτή και η άλλη κοίλη. Έτσι είναι λογικό να περιμένουμε το μέγιστο πλήθος των γραμμικών περιοχών να είναι μεγαλύτερο. Αρχικά, θα μελετήσουμε την περίπτωση ενός max-min TP που χρησιμοποιείται για regression με ανεξάρτητη μεταβλητή x και τάξη k. Το αποτέλεσμά μας είναι το ακόλουθο:

Πρόταση 17. Έστω $f(\mathbf{x})$ η συνάρτηση ενός max-min tropical perceptron τάξης k με $\mathbf{x} \in \mathbb{R}$, δηλαδή

$$\begin{aligned} f(x) &= \lambda\delta(x) + (1-\lambda)\varepsilon(x) \\ &= \lambda\left(\bigvee_{i=1}^{k} c_i x + w_i\right) + (1-\lambda)\left(\bigwedge_{i=1}^{k} d_i x + m_i\right) \end{aligned}$$

Τότε, για το πλήθος $\mathrm{LR}_k(f)$ των γραμμικών περιοχών της f ισχύει

$$\mathrm{LR}_k(f) \leq 2k - 1$$

Απόδειξη. Η απόδειξη γίνεται με επαγωγή στην τάξη k του perceptron. Όπως αναφέραμε και στην ενότητα 5.1.2, οποιαδήποτε τμηματικά γραμμική συνάρτηση γράφεται στη μορφή

$$\{\max_i, \min_i\} a_i x + b_i$$

αποτελείται από K τμήματα με κλίση (slope) a_i και 2 διαδοχικά τμήματα τέμνονται στο σημείο $\dfrac{b_{i+1} - b_i}{a_i - a_{i+1}}$. Για λόγους απλότητας και χωρίς βλάβη της γενικότητας, θεωρούμε

$$f(x) = \delta(x) + \varepsilon(x)$$

- $k = 1$: Για μοναδικό όρο, η $f(x)$ έχει τη μορφή

  $$(c_1 x + w_1) + (d_1 x + m_1) = (c_1 + d_1)x + (w_1 + m_1)$$

  επομένως ικανοποιεί την υπόθεση αφού $2 \cdot 1 - 1 = 1$.

- $k = 2$: Σε αυτή την περίπτωση, έχουμε 3 ενδεχόμενα. Συμβολίζουμε με x_h^*, x_l^* το σημείο τομής των 2 όρων του max και του min αθροίσματος αντίστοιχα:

  1. $x_h^* = x_l^*$ - στην περίπτωση αυτή, η $f(x)$ έχει τη μορφή

     $$f(x) = \begin{cases} (c_1 + d_1)x + (w_1 + m_1) & x \leq x_h^* \\ (c_2 + d_2)x + (w_2 + m_2) & x \geq x_h^* \end{cases}$$

  2. $x_h^* > x_l^*$ - πλέον, έχουμε 3 περιοχές της εισόδου με 3 διαφορετικές κλίσεις:

     $$f(x) = \begin{cases} (c_1 + d_1)x + (w_1 + m_1) & x \leq x_l^* \\ (c_1 + d_2)x + (w_1 + m_2) & x_l^* \leq x \leq x_h^* \\ (c_2 + d_2)x + (w_2 + m_2) & x \geq x_h^* \end{cases}$$

     Μια τέτοια περίπτωση για $k = 2$ απεικονίζεται και στην Εικόνα 5.3.

3. $x_h^* < x_l^*$ - αντίστοιχη με την προηγούμενη περίπτωση

$$f(x) = \begin{cases} (c_1 + d_1)x + (w_1 + m_1) & x \leq x_h^* \\ (c_2 + d_1)x + (w_2 + m_1) & x_h^* \leq x \leq x_l^* \\ (c_2 + d_2)x + (w_2 + m_2) & x \geq x_l^* \end{cases}$$

Σε κάθε περίπτωση έχουμε $\mathrm{LR}_2(f) \leq 2 \cdot 2 - 1$

- Έστω ότι έχουμε αποδείξει το ζητούμενο για τυχαίο k, δηλαδή $k \leq LR_k(f) \leq 2 \cdot k - 1$. Θα δείξουμε ότι ισχύει για $k + 1$. Χωρίς βλάβη της γενικότητας, μπορούμε να θεωρήσουμε πως τα νέα σημεία τομής x_h^k, x_l^k μεταξύ των όρων $k, k + 1$ του εκάστοτε max/min-αθροίσματος είναι μεγαλύτερα από $x_K = \max(x_h^*, x_l^*)$ για την περίπτωση των k όρων (αλλιώς μπορούμε να φτάσουμε σε αυτή την υπόθεση με κατάλληλη αναδιάταξη των όρων). Έστω

$$f_k(x) = sx + t$$

η γραμμική εξίσωση της f στο διάστημα $[x_K, \infty]$. Τότε, όπως και στην περίπτωση $k = 2$ διακρίνουμε τις εξής περιπτώσεις:

1. $x_h^k = x_l^k$ - σε αυτή την περίπτωση, προστίθεται μόνο ένας όρος $f_{k+1}(x) = (s + c_{k+1} + d_{k+1})x + (t + w_{k+1} + m_{k+1})$ στους όρους της f. Επομένως έχουμε

$$k + 1 \leq \mathrm{LR}_{k+1}(f)$$

2. $x_h^k \neq x_l^k$ - σε αυτή την περίπτωση, προστίθενται 2 όροι, όπως στην περίπτωση για $k = 2$, και επομένως έχουμε

$$\mathrm{LR}_{k+1}(f) \leq 2 \cdot k - 1 + 2 = 2 \cdot k + 1 (= 2 \cdot (k + 1) - 1)$$

Άρα τελικά ισχύει

$$\mathrm{LR}_{k+1}(f) \leq 2 \cdot (k + 1) - 1$$

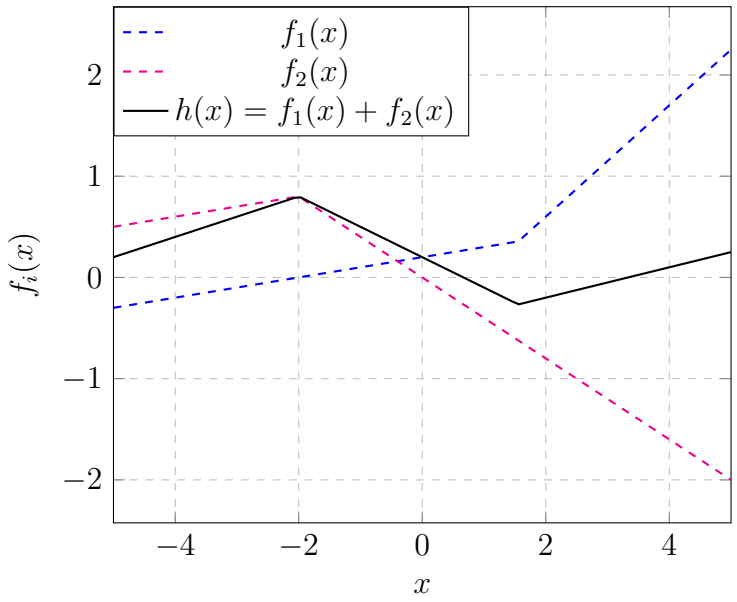
και το επαγωγικό επιχείρημα έχει ολοκληρωθεί.

□

Δεν έχουμε καταφέρει να γενικεύσουμε αυτό το αποτέλεσμα για περισσότερες διαστάσεις όσον αφορά το max-min perceptron, ωστόσο στην επόμενη ενότητα διερευνούμε σε μεγαλύτερο βάθος και γενικότητα περιπτώσεις όπου ο min-όρος απουσιάζει.

5.1.4 Πολύτοπα Newton και δίκτυα maxout

Στο σημείο αυτό, μπορούμε να δείξουμε ένα ακόμη αποτέλεσμα που καταδεικνύει τη σύνδεση μη-γραμμικών classifiers όπως τα δίκτυα maxout με την τροπική γεωμετρία. Για το σκοπό αυτό,



Σχήμα 5.3: Max-min TP με $k = 2$ όρους

ανακαλούμε τον ορισμό της τροπικής υπερεπιφάνειας $V(p)$ ενός πολυωνύμου p όπως αυτός δόθηκε στο Κεφ. 4.4.1. Δεδομένου ενός maxout όρου της μορφής

$$h_i(\mathbf{x}) = \bigvee_j f_j(\mathbf{x}) = \bigvee_j W_{ij,\cdot}\mathbf{x} + b_{ij}$$

τα σημεία που ανήκουν στο $V(p)$ είναι το σύνολο

$$\{\boldsymbol{w} \in \mathbb{R}^n : f_j(\mathbf{w}) = f_i(\mathbf{w}) = \max_k f_k, i \neq j\}$$

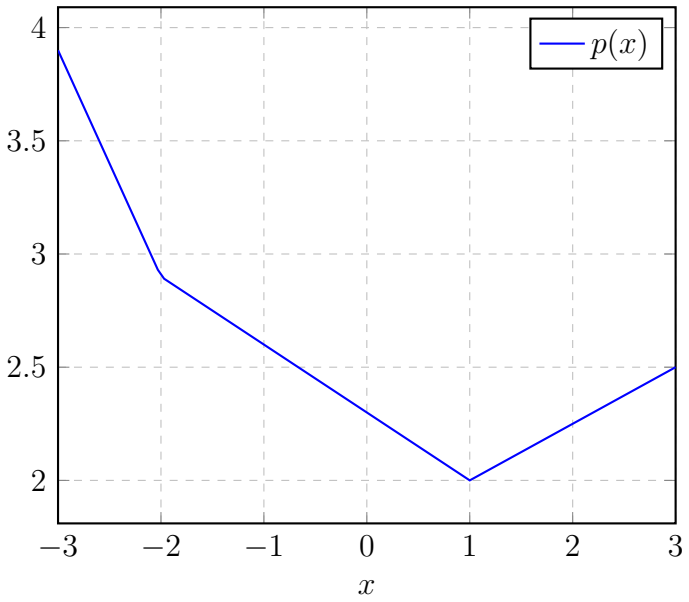
δηλαδή το σύνολο των σημείων στα οποία το μέγιστο επιτυγχάνεται από ≥ 2 όρους του max-αθροίσματος. Ένα παράδειγμα φαίνεται στην εικόνα 5.4. Όπως είδαμε προηγουμένως, η έξοδος ενός τροπικού perceptron αλλά και μιας μονάδας ενός maxout δικτύου αποτελεί ένα max-πολυώνυμο της μορφής

$$h_i(x) = \bigvee_j W_{ij,\cdot}\mathbf{x} + b_{ij} = \bigvee_j W_{ij1}x_1 + \cdots + W_{ijn}x_n + b_{ij}$$

άρα μπορούμε να συνάγουμε ότι:

Σημείωση. Οι μη-γραμμικές περιοχές του χώρου χαρακτηριστικών (feature space) μιας maxout μονάδας με εξίσωση $h_i(\mathbf{x})$ είναι ακριβώς η τροπική υπερεπιφάνεια $V(h_i)$.

Στα περισσότερα εισαγωγικά κεφάλαια άρθρων ή βιβλίων τροπικής γεωμετρίας (π.χ. [39]) ορίζεται το λεγόμενο *πολύτοπο Newton* για πολυώνυμα σε ένα δακτύλιο $K[x_1, \ldots, x_n]$ (ή για την τροπική εκδοχή αυτών). Εάν $n = 2$, αναφερόμαστε στο *πολύγωνο* Newton. Στην τροπική γεωμετρία, οι τροπικές καμπύλες είναι δυϊκές ως προς τα πολύγωνα Newton, γεγονός που αποτέλεσε έναυσμα



Σχήμα 5.4: $V(p) = \{-2, 1\}$

για τη μελέτη μας σε αυτό το κεφάλαιο. Αρχικά δίνουμε τον ορισμό μόνο για την περίπτωση ενός τροπικού πολυωνύμου, μιας και δε θα χρειαστούμε πολύγωνα Newton κλασσικών πολυωνύμων στη συνέχεια.

Ορισμός 22 (Newton Polytope)**.** Έστω $p : \mathbb{R}^n \to \mathbb{R}$ ένα τροπικό πολυώνυμο με τύπο

$$p(\mathbf{x}) = \max_{i \in I}\{c_{i1}x_1 + \cdots + c_{in}x_n\} = \max_i\{\mathbf{c}_i^T\mathbf{x}\} \tag{5.9}$$

Τότε το πολύτοπο Newton που σχετίζεται με το p δίνεται από τον τύπο

$$\text{Newt}(p) = \text{conv}\{\boldsymbol{c}_i : i \in I\} = \text{conv}\{(c_{i1}, \ldots, c_{in}) : i \in I\}$$

δηλαδή αποτελεί το κυρτό κάλυμμα (convex hull) των (τροπικών) διανυσμάτων εκθετών (exponent vectors).

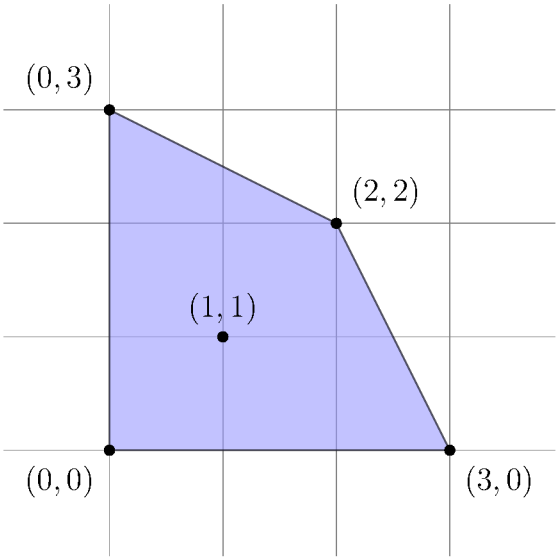
Στο σχήμα 5.5 απεικονίζεται το πολύγωνο Newton του τροπικού πολυωνύμου

$$p(x, y) = 0 \vee x + y \vee 3x \vee 2x + 2y \vee 3y \tag{5.10}$$

το οποίο αντιστοιχεί στα διανύσματα συντελεστών

$$C = \{(0, 0), (1, 1), (3, 0), (2, 2), (0, 3)\} \tag{5.11}$$

Χρησιμοποιώντας βασικές γνώσεις από θεωρία βελτιστοποίησης, είναι εύκολο να δείξουμε ότι για max-πολυώνυμα της μορφής (5.9), το πολύτοπο Newton υποδεικνύει έμμεσα και το πλήθος των γραμμικών περιοχών. Είναι εύκολο να δείξουμε το ακόλουθο:



Σχήμα 5.5: Newt $[\max(0, x + y, 3x, 2x + 2y, 3y)]$

Πρόταση 18. Ο αριθμός των γραμμικών περιοχών ενός max-πολυωνύμου p της μορφής 5.9 είναι ίσος με το πλήθος των κορυφών του πολυτόπου Newton, Newt(p).

Απόδειξη. Μια απόδειξη μπορεί να δοθεί με τη χρήση του Θεμελιώδους Θεωρήματος του Γραμμικού Προγραμματισμού [64, Theorem 3.4]. Θεωρούμε το γραμμικό πρόγραμμα:

$$\begin{aligned} &\text{Maximize } z = \mathbf{x}^T \boldsymbol{c} \\ &\text{s.t. } \boldsymbol{c} \in \text{Newt}(p) \end{aligned} \tag{5.12}$$

στο οποίο η μεταβλητή προς βελτιστοποίηση είναι το διάνυσμα των συντελεστών, $\boldsymbol{c}$. Από το θεμελιώδες θέωρημα γραμμικού προγραμματισμού, και επειδή το Newt(p) είναι φραγμένο, γνωρίζουμε ότι για κάθε επιλογή $\mathbf{x} \in \mathbb{R}^n$, το πρόβλημα (5.12) αποτελεί ένα γραμμικό πρόγραμμα το μέγιστο του οποίου δίνεται όταν το $\boldsymbol{c}$ είναι κάποια κορυφή του Newt(p). Επομένως, κάθε διάνυσμα $\boldsymbol{c}_j \notin \text{vert}(\text{Newt}(p))$ δεν μπορεί να ανήκει σε μεγιστικό όρο του max-πολυωνύμου. Συνεπώς, το πλήθος των γραμμικών περιοχών είναι ίσο με το πλήθος των κορυφών του Newt(p). □

Έτσι, στο παράδειγμα της Εικόνας 5.5, ο όρος $x + y$ του max-πολυωνύμου δεν "συνεισφέρει" καμία γραμμική περιοχή (δηλαδή δεν είναι ποτέ ο μεγαλύτερος όρος του πολυωνύμου) σύμφωνα με την πρόταση 18, κάτι το οποίο μπορεί να επαληθευθεί και αναλυτικά, επιλύοντας ανισώσεις μεταξύ των όρων του πολυωνύμου.

Η παραπάνω αντιμετώπιση δεν λαμβάνει υπ'όψιν τους σταθερούς όρους στους παράγοντες ενός maxout unit, ωστόσο μπορουμε να την επεκτείνουμε για αυτή την περίπτωση. Εάν ένα max-πολυώνυμο έχει τη μορφή

$$p(\mathbf{x}) = \max_{i \in I} \{c_{i1}x_1 + \cdots + c_{in}x_n + b_i\} = \max_i \{\boldsymbol{c}_i^T \mathbf{x} + b_i\}$$

Συμβολίζοντας με $\mathbf{x}'$ το διάνυσμα $[1, x_1, \dots x_n]^T$ και με με $\boldsymbol{c}_i'$ το διάνυσμα $[b_i, c_{i1}, \dots c_{in}]^T$, γράφουμε το παραπάνω στη μορφή

$$p(\mathbf{x}) = \max_i \{\boldsymbol{c}_i^T \mathbf{x} + b_i\} = \max_i \{(\boldsymbol{c}')^T \mathbf{x}'\}$$

Χρησιμοποιούμε το επεκτεταμένο πολύτοπο Newton (extended Newton Polytope), το οποίο ορίζεται ως

$$\text{Newt}(p) = \text{conv}\{(b_i, c_{i1}, \dots c_{in}) : i \in I\} \tag{5.13}$$

Παρόμοια με την απόδειξη της Πρότασης (18), μπορούμε να αντιστοιχίσουμε το γραμμικό πρόγραμμα

$$\begin{aligned} &\text{Maximize } z = (\mathbf{x}')^T \boldsymbol{c}' = b_i + \mathbf{x}^T \boldsymbol{c} \\ &\text{s.t. } \boldsymbol{c}' \in \text{Newt}(p) \end{aligned} \tag{5.14}$$

Οι βέλτιστες λύσεις αυτού του προγράμματος δίνονται από το λεγόμενο άνω κάλυμμα (*upper hull*) του πολυτόπου Newton. Για ένα κυρτό πολύτοπο, έστω $P \in \mathbb{R}^{d+1}$, το άνω κύτος ως προς την πρώτη συντεταγμένη ορίζεται ως

$$P^{\max} := \{(\lambda, \mathbf{x}) \in P : \forall (t, \mathbf{x}) \in P \Rightarrow \lambda > t\} \in \mathbb{R} \times \mathbb{R}^d \tag{5.15}$$

Πράγματι, έστω $\boldsymbol{c}_k' = (b_k, \boldsymbol{c}_k) \notin \text{Newt}^{\max}(p)$. Τότε, $\exists \boldsymbol{c}_l' = (b_l, \boldsymbol{c}_k) \in \text{Newt}^{\max}(p) \Rightarrow b_l > b_k$. Για την τιμή της αντικειμενικής συνάρτησης στα σημεία αυτά, όπως ορίζεται στο Πρόβλημα (5.14), ισχύει:

$$\begin{aligned} z_{\boldsymbol{c}_k'} &= b_k + \mathbf{x}^T \boldsymbol{c}_k \\ z_{\boldsymbol{c}_l'} &= b_l + \mathbf{x}^T \boldsymbol{c}_k \end{aligned}$$

για τα οποία προφανώς ισχύει $z_{\boldsymbol{c}_k'} < z_{\boldsymbol{c}_l'}$. Επομένως, οι όροι του extended newton polytope που δεν ανήκουν στο άνω κύτος του αντιστοιχούν σε μη-απαραίτητους όρους του max-πολυωνύμου. Συνάγουμε:

Πρόταση 19. Το πλήθος των γραμμικών περιοχών ενός maxout unit με τύπο

$$h(\mathbf{x}) = \bigvee_{i \in I} \{W_{i,\cdot}\mathbf{x} + b_i\}$$

είναι ίσο με το πλήθος των κορυφών του $\text{Newt}(h)$ που ανήκουν στο άνω κάλυμμα (upper hull) του, $\text{Newt}^{\max}(h)$.

Χρησιμοποιώντας αυτό το αποτέλεσμα, και παρατηρώντας ότι το σύνολο των διανυσμάτων

$$\{(1, 0, \dots, 0), (0, 1, \dots, 0) \dots, (0, 0, \dots, 1)\}$$

ανήκει στο αντίστοιχο upper hull, μπορούμε να "διορθώσουμε" την πρόταση 15 ως εξής:

Πρόταση 20. Για οποιαδήποτε επιλογή παραμέτρων w_i, ένα tropical max-perceptron έχει πάντα ακριβώς n γραμμικές περιοχές.

Με τη χρήση του πολύτοπου Newton, μπορούμε να εξάγουμε θεωρητικά αποτελέσματα και για ελαφρώς πιο σύνθετα μοντέλα, τα οποία μπορεί να εμφανίζονται σε δίκτυα από maxout νευρώνες. Συγκεκριμένα, μπορούμε αφενός να αποφανθούμε για το πλήθος των γραμμικών περιοχών σε μοντέλα που αποτελούνται από το supremum από maxout activations, δηλαδή περιγράφονται από τη μορφή

$$g_{\vee}(\mathbf{x}) = \bigvee_{i=1}^{m} h_i(\mathbf{x}) \tag{5.16}$$

όπου $h_i(\mathbf{x})$ είναι όροι maxout.
Μπορούμε, ακόμη, να εξετάσουμε και προσθετικά μοντέλα, της μορφής

$$g_{+}(\mathbf{x}) = \sum_{i=1}^{m} h_i(\mathbf{x}) \tag{5.17}$$

Το κλειδί στις παραπάνω περιπτώσεις είναι να παρατηρήσουμε ότι κάθε όρος $h_i(\mathbf{x})$ είναι συσχετισμένος με ένα πολύτοπο Newton $\mathrm{Newt}(h_i)$ και οι πράξεις $\vee, +$ καθορίζουν το πολύτοπο Newton που αντιστοιχεί στην προκύπτουσα συνάρτηση. Για την πρώτη περίπτωση, έχουμε το supremum από συναρτήσεις που είναι με τη σειρά τους supremum από αφφινικούς όρους, ενώ στη δεύτερη περίπτωση, αρκεί να χρησιμοποιούμε την ιδιότητα

$$a + (b \vee c) = (a + b) \vee (a + c)$$

η οποία γενικεύεται έτσι ώστε

$$(a_1 \vee \cdots \vee a_n) + (b_1 \vee \cdots \vee b_n) = (a_1 + b_1) \vee \cdots \vee (a_1 + b_n) \vee \cdots \vee (a_n + b_1) \vee \cdots \vee (a_n + b_n)$$

Έτσι καταλήγουμε στην ακόλουθη πρόταση:

Πρόταση 21. Έστω $H = \{h_1, \dots h_m\}$ ένα σύνολο από όρους maxout. Συμβολίζουμε $\mathrm{Newt}(h_i)$ το πολύτοπο Newton που αντιστοιχεί στον όρο h_i. Έστω οι συναρτήσεις

$$g_{\vee}(\mathbf{x}) = \bigvee_{i=1}^{m} h_i(\mathbf{x})$$

και

$$g_{+}(\mathbf{x}) = \sum_{i=1}^{m} h_i(\mathbf{x})$$

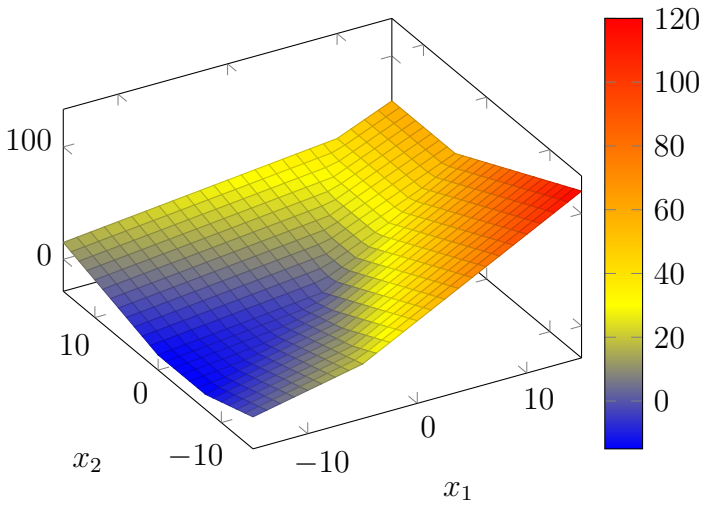
Τα πολύτοπα Newton που αντιστοιχούν στις παραπάνω συναρτήσεις είναι:

- $\mathrm{Newt}(g_{\vee}) = \mathrm{conv}(\mathrm{Newt}(h_1), \dots, \mathrm{Newt}(h_m))$
- $\mathrm{Newt}(g_{+}) = \mathrm{Newt}(h_1) \oplus \cdots \oplus \mathrm{Newt}(h_m)$ όπου $\oplus$ συμβολίζει το άθροισμα Minkowski των πολυτόπων.

Έτσι, μπορούμε αυτόματα να αποφανθούμε για το πλήθος των γραμμικών περιοχών για αθροιστικά ή piecewise-max μοντέλα που περιλαμβάνουν όρους maxout. Σε αυτά μπορούμε να συμπεριλάβουμε και την περίπτωση του γενικευμένου αθροίσματος

$$g_+(\mathbf{x}) = \sum_{i=1}^{m} w_i \cdot h_i(\mathbf{x}), \quad \boldsymbol{w} \geq 0 \tag{5.18}$$

για το οποίο ισχύει η ίδια πρόταση, εφόσον παρατηρήσουμε ότι το πολύτοπο Newton που αντιστοιχεί στην έκφραση $w_i h_i(\mathbf{x})$ είναι το $\mathrm{Newt}(h_i)$ κλιμακούμενο κατά w_i. Στην εικόνα 5.6 φαίνεται η γραφική



Σχήμα 5.6: $g(x, y)$

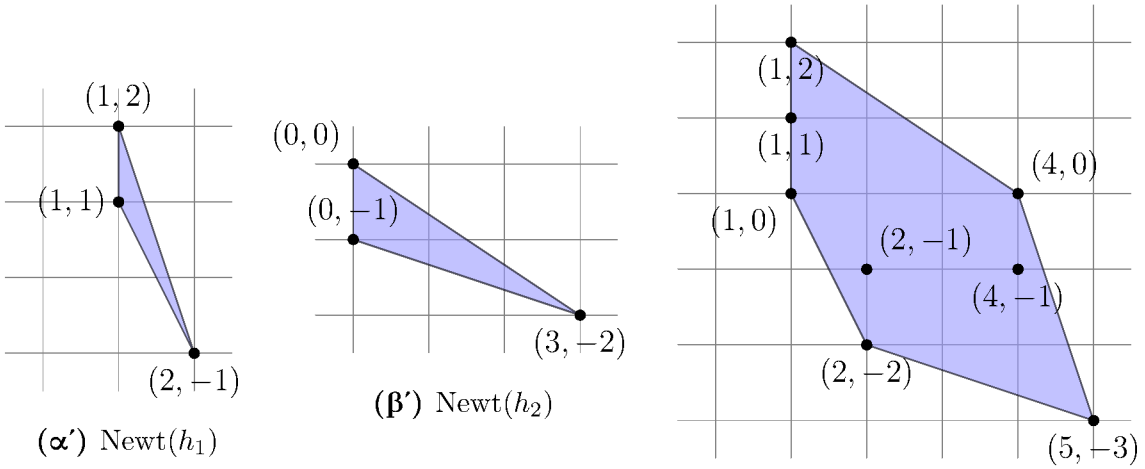
της συνάρτησης

$$g(x, y) = \max(x + y, 2x - y, x + 2y) + \max(0, -y, 3x - 2y)$$

η οποία αποτελείται από 5 γραμμικές περιοχές. Αντίστοιχα, σχεδιάζοντας τα πολύγωνα Newton των 2 max-όρων από τους οποίους αποτελείται

$$h_1(\mathbf{x}) = \max(x + y, 2x - y, x + 2y)$$
$$h_2(\mathbf{x}) = \max(0, -y, 3x - 2y)$$

προκύπτει η εικόνα 5.7, στην οποία φαίνεται και το πολύγωνο Newton που προκύπτει ως το άθροισμα minkowski των 2 επιμέρους. Πράγματι, οι κορυφές του τελευταίου πολυγώνου Newton είναι 5, σε 1-1 αντιστοιχία με το πλήθος των γραμμικών περιοχών της συνάρτησης $g(x, y)$.
Η χρησιμότητα του πολυτόπου Newton για την κατανόηση της πολυπλοκότητας διαφόρων μοντέλων μάθησης έχει απασχολήσει και άλλους ερευνητές στο παρελθόν. Για παράδειγμα, στο [49], οι συγγραφείς δείχνουν πως το πλήθος των συναρτήσεων συμπερασμού (inference function) ενός γραφικού μοντέλου f είναι σε 1-1 αντιστοιχία με το πλήθος των κορυφών του πολυτόπου Newton της απεικόνισης f.



(α′) Newt(h_1) **(β′)** Newt(h_2) **(γ′)** Newt(h_1) + Newt(h_2)

Σχήμα 5.7: Πολύγωνα Newton για τους όρους της $g(x, y)$

5.1.5 Απλοποίηση τροπικών πολυωνύμων

Παρατηρούμε ότι μια ευχάριστη συνέπεια της Πρότασης 19 είναι το γεγονός ότι μας προσφέρει μια φυσική μέθοδο να "ελαχιστοποιούμε" τροπικά πολυώνυμα σε αυθαίρετο πλήθος διαστάσεων. Με τον όρο ελαχιστοποίηση, εννοούμε την αφαίρεση των παραγόντων ενός maxπολυωνύμου που δεν συνεισφέρουν ποτέ στο αποτέλεσμα, δηλαδή τους όρους του πολυωνύμου $p(\mathbf{x})$ που δεν το μεταβάλλουν ως συνάρτηση. Στο βιβλίο του Butkovič [9], περιγράφεται ένας αλγόριθμος με χρόνο εκτέλεσης $\mathcal{O}(n)$, όπου n το πλήθος των παραγόντων, για την περίπτωση των τροπικών πολυωνύμων μιας μεταβλητής. Ο αλγόριθμος του Butkovič στηρίζεται στην παραγοντοποίηση του πολυωνύμου ως ενδιάμεσο βήμα, ωστόσο γνωρίζουμε ότι για 2 και άνω μεταβλητές αυτό αποτελεί δυσεπίλυτο (NP-complete) πρόβλημα, όπως περιγράφεται στο [28].

Περιγράφουμε τον αλγόριθμό μας παρακάτω, ο οποίος ουσιαστικά υπολογίζει τα σημεία του άνω καλύμματος του εκτεταμένου πολυτόπου Newton, χρησιμοποιώντας 2 υπολογισμούς κυρτών καλυμμάτων. Διαισθητικά, αυτό που κάνει είναι να υπολογίζει ένα υποσύνολο του καλύμματος

$$\text{Conv}\left(\{-\infty\} \cup \text{Newt}(p)\right) \tag{5.19}$$

του πολυωνύμου $p(\mathbf{x})$.

Algorithm 7 Απομάκρυνση μη-απαραίτητων όρων από τροπικά πολυώνυμα

function KEEPESSENTIAL($p : \mathbb{R}^d \to \mathbb{R}, p(\mathbf{x}, \boldsymbol{W}, \boldsymbol{b}) = \bigvee_{i \in K} \boldsymbol{W}_{i,\cdot}\mathbf{x} + b_i$)
  Find $\text{Newt}(p) = \text{conv}((b_i, \boldsymbol{W}_{i,\cdot}) : i \in K)$
  $\mathcal{U} \leftarrow \emptyset$
  for $v_i = (b_i, \boldsymbol{W}_{i,\cdot}) \in \text{Newt}(p)$ **do**
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{(0, \boldsymbol{W}_{i,\cdot})\}$
  end for
  Find $\mathcal{V} = \text{conv}(\mathcal{U})$ ▷ Project to $n-1$ dimensions
  $b_{\min} = \min\{b_i\}$
  for $v_i \in \text{vert}(\mathcal{V})$ **do**
    Set $v_i \leftarrow (b_{\min} - \epsilon, \boldsymbol{W}_{i,\cdot})$
  end for
  Find $\mathcal{L} = \text{conv}(\mathcal{V} \cup \text{Newt}(p))$ ▷ Computer lower envelope
  return $\text{vert}(\mathcal{L}) \cap \text{vert}(\text{Newt}(p))$
end function

Η εφαρμογή του Αλγορίθμου 7 μπορεί να φανεί γεωμετρικά ανά στάδιο για ένα πολυώνυμο μιας μεταβλητής στο Σχήμα 5.8.

Όπως φαίνεται, ο αλγόριθμος απλοποιεί το πολυώνυμο $p(x)$ στην ελάχιστη μορφή

$$p_s(x) = \max\{2, 2x + 1, -2x + 0.6\} \tag{5.20}$$

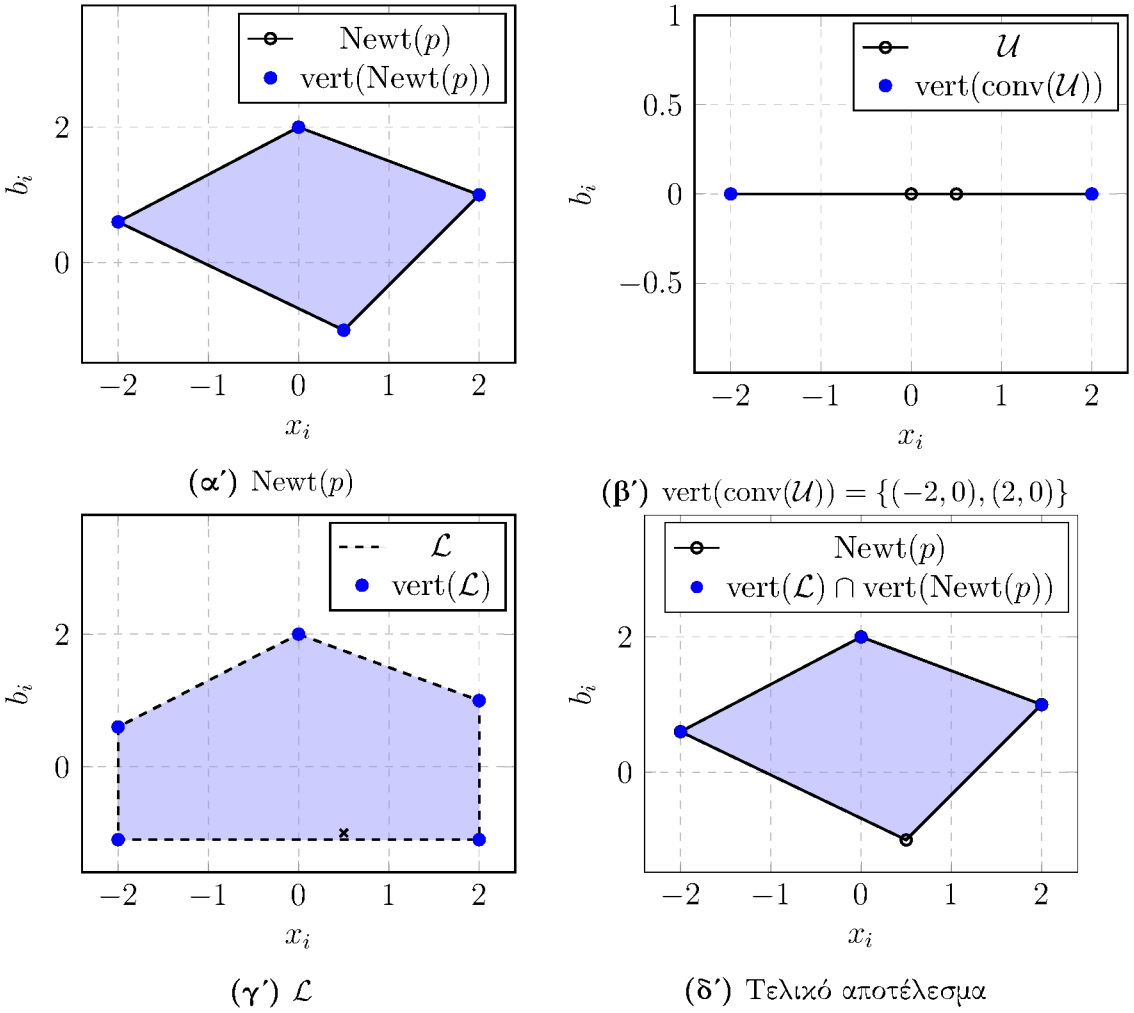
την οποία μπορούμε να επαληθεύσουμε ως προς την ορθότητα χρησιμοποιώντας π.χ. τον αλγόριθμο [9, Algorithm 5.1.11]. Στην Εικόνα 5.9 φαίνεται ότι πράγματι ο όρος $0.5x - 1$ δε συνεισφέρει σε καμία

περίπτωση στη συνάρτηση που ορίζει το πολυώνυμο $p(x)$. Η πολυπλοκότητα του Αλγορίθμου 7 είναι στενά συνδεδεμένη με την πολυπλοκότητα των αλγορίθμων εύρεσης του convex hull ενός συνόλου σημείων στον $\mathbb{R}^n$. Για παράδειγμα, στην περίπτωση $n = 1$, πρέπει να βρούμε το upper hull ενός συνόλου σημείων στο $\mathbb{R}^2$, αφού αφιερώνουμε μια επιπλέον διάσταση στις σταθερές b_i. Γενικά, ισχύει:

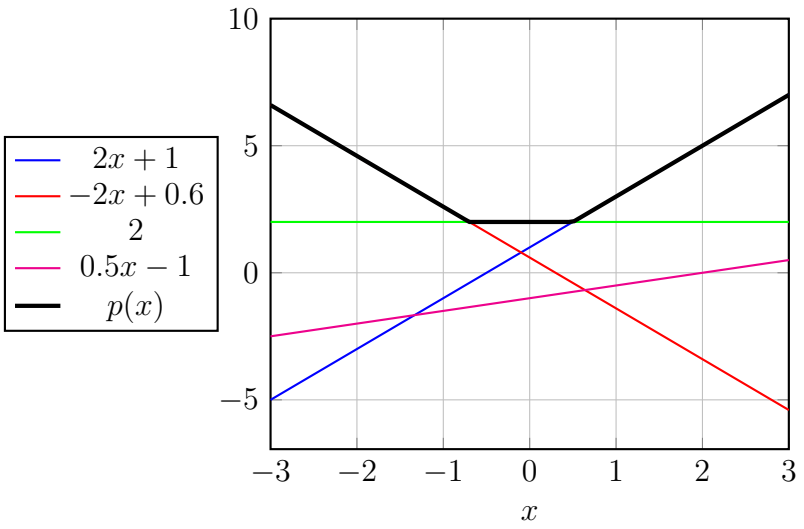
Πρόταση 22. Έστω $p : \mathbb{R}^d \to \mathbb{R}$ ένα max-πολυώνυμο με n όρους. Ο χρόνος εκτέλεσης του Αλγορίθμου 7 με είσοδο το p είναι:

$$T(d,n) = \begin{cases} \mathcal{O}(n \log n) & d \in \{1,2\} \\ \mathcal{O}(n \log n + n^{\lfloor d/2 \rfloor}) & d \geq 3 \end{cases} \tag{5.21}$$

Για σημεία στις 2 και 3 διαστάσεις, οι βέλτιστοι αλγόριθμοι εύρεσης convex hull έχουν χρόνο εκτέλεσης $\mathcal{O}(n \log k)$, όπου k το πλήθος των κορυφών του convex hull. Για μεγαλύτερες διαστάσεις d, οι ταχύτεροι αλγόριθμοι έχουν ασυμπτωτικό χρόνο εκτέλεσης $\mathcal{O}(n \log n + n^{\lfloor d/2 \rfloor})$. Στην πραγματικότητα, η εύρεση convex hull στις d διαστάσεις είναι ταυτόχρονα κάτω φραγμένη, αφού το convex hull μπορεί να έχει μέχρι και $\mathcal{O}(n^{\lfloor d/2 \rfloor})$ όψεις [6]. Ο Αλγόριθμος 7 πραγματοποιεί 3 ευρέσεις convex hull, εκ των οποίων οι 2 γίνονται για διάσταση $d+1$ και η μια για διάσταση d, επομένως καταλήγουμε στο χρόνο εκτέλεσης της Πρότασης 22.



Σχήμα 5.8: Στάδια εφαρμογής του Αλγορίθμου 7 για το πολυώνυμο $p(x) = \max\{2, 2x + 1, -2x + 0.6, 0.5x - 1\}$



Σχήμα 5.9: Το πολυώνυμο $p(x)$ και οι παράγοντές του

5.2 Εκπαίδευση

5.2.1 Gradient Descent

Η εκπαίδευση του perceptron του σχήματος 5.1 μπορεί να πραγματοποιηθεί με μια απλή διαδικασία κατάβασης κλίσεων (*gradient descent*), με στόχο την ελαχιστοποίηση κάποιας συνάρτησης κόστους, όπως είναι η τετραγωνική:

$$J(\mathbf{x}_{1\dots k}, \boldsymbol{w}, \boldsymbol{m}) = \frac{1}{2}\sum_k (\mathbf{y}(\mathbf{x}_k) - d_k)^2 \tag{5.22}$$

ή η αρνητική λογαριθμική πιθανοφάνεια (*negative log-likelihood*):

$$J(\mathbf{x}_{1\dots k}, \boldsymbol{w}, \boldsymbol{m}) = -\sum_k \log P(\mathbf{y}(\mathbf{x}_k) = d_k | \boldsymbol{w}, \boldsymbol{m}) \tag{5.23}$$

όπου d_k είναι η επιθυμητή έξοδος του δικτύου για την είσοδο $\mathbf{x}_k$. Η κλίση ως προς κάθε παράμετρο του δικτύου μπορεί να βρεθεί γράφοντας αναλυτικά την παράγωγο, ∂J. Παρακάτω φαίνεται η διαδικασία για την περίπτωση της τετραγωνικής συνάρτησης κόστους:

$$\begin{aligned}
\frac{\partial J}{\partial \delta} &= \sum_k (y(\mathbf{x}_k) - d_k) \cdot \frac{\partial y(\mathbf{x}_k)}{\partial \delta} \\
&= \sum_k (y(\mathbf{x}_k) - d_k) \cdot \lambda \qquad (5.24) \\
\frac{\partial J}{\partial \epsilon} &= \sum_k (y(\mathbf{x}_k) - d_k) \cdot \frac{\partial y(\mathbf{x}_k)}{\partial \epsilon} \\
&= \sum_k (y(\mathbf{x}_k) - d_k) \cdot (1 - \lambda) \qquad (5.25) \\
\frac{\partial J}{\partial w_i} &= \sum_k (y(\mathbf{x}_k) - d_k) \cdot \frac{\partial y(\mathbf{x}_k)}{\partial w_i} \\
\frac{\partial J}{\partial m_i} &= \sum_k (y(\mathbf{x}_k) - d_k) \cdot \frac{\partial y(\mathbf{x}_k)}{\partial m_i} \\
\frac{\partial y(\mathbf{x}_k)}{\partial w_i} &= \lambda \cdot \frac{\partial \delta(\mathbf{x}_k)}{\partial w_i} \\
\frac{\partial \epsilon(\mathbf{x}_k)}{\partial m_i} & \\
&= \lambda \cdot \begin{cases} 1 & \arg\max_l \{w_l + x_l\} = i \\ 0 & \text{αλλιώς} \end{cases} \qquad (5.26) \\
\frac{\partial y(\mathbf{x}_k)}{\partial m_i} &= (1 - \lambda) \cdot \frac{\partial \epsilon(\mathbf{x}_k)}{\partial m_i} \\
&= (1 - \lambda) \cdot \begin{cases} 1 & \arg\min_l \{m_l + x_l\} = i \\ 0 & \text{αλλιώς} \end{cases} \qquad (5.27)
\end{aligned}$$

Στις παραπάνω εξισώσεις, οι μερικές παράγωγοι $\frac{\theta\delta(\mathbf{x})}{\theta w_i}$ μπορούν να προσαρμοστούν με βάση την ιδέα του subdifferential έτσι ώστε

$$\frac{\partial\delta(\mathbf{x}_k)}{\partial w_i} = \begin{cases} \frac{1}{|\{j : \arg\max_l w_l + x_l = j\}|} & \arg\max_l\{w_l + x_l\} = i \\ 0 & \text{αλλιώς} \end{cases} \tag{5.28}$$

δηλαδή διαιρούμε τη σταθερά της παραγώγου με το πλήθος των παραγόντων που μεγιστοποιούν (ή ελαχιστοποιούν αντίστοιχα για τις m_i) την έξοδο, αφού το subdifferential στα σημεία ασυνέχειας είναι $\partial\delta = [0, 1]$.
Στην περίπτωση που περιλαμβάνονται οι "εκθέτες" c_i, d_i όπως στις εξισώσεις (5.4), (5.5), με παρόμοιο τρόπο ισχύει ότι

$$\frac{\partial\delta(\mathbf{x}_k)}{\partial c_i} = \begin{cases} \frac{\mathbf{x}_{k,i}}{|\{j : \arg\max_l w_l + x_l c_l = j\}|} & \arg\max_l w_l + x_l c_l = i \\ 0 & \text{αλλιώς} \end{cases}$$

$$\frac{\partial\epsilon(\mathbf{x}_k)}{\partial d_i} = \begin{cases} \frac{\mathbf{x}_{k,i}}{|\{j : \arg\min_l m_l + x_l d_l = j\}|} & \arg\min_l m_l + x_l d_l = i \\ 0 & \text{αλλιώς} \end{cases}$$

Η ανανέωση των παραμέτρων ακολουθεί το γνωστό κανόνα της κατάβασης κλίσεων, δηλαδή

$$w_i^{t+1} = w_i^t - \alpha_t \frac{\partial J}{\partial w_i^t}$$
$$m_i^{t+1} = m_i^t - \alpha_t \frac{\partial J}{\partial m_i^t}$$
$$c_i^{t+1} = c_i^t - \alpha_t \frac{\partial J}{\partial c_i^t}$$
$$d_i^{t+1} = d_i^t - \alpha_t \frac{\partial J}{\partial d_i^t}$$

όπου το βήμα α_t (ρυθμός μάθησης) μπορεί να είναι είτε σταθερό είτε κάποιο μεταβαλλόμενο της μορφής $f(t)$, όπως

$$\alpha_t = \frac{\alpha_0 t}{1 + t}$$

Επίσης, αρκετές από τις υπάρχουσες ευριστικές μεθόδους (όπως κατάβαση κλίσεων με όρο ορμής, προσαρμοστικοί ρυθμοί μάθησης, weight decay κλπ.) μπορούν να εφαρμοστούν στην παραπάνω διαδικασία.

5.2.2 Convex-Concave Programming

Οι επαναληπτικές διαδικασίες όπως το gradient descent χρησιμοποιούνται πολύ συχνά για την εκπαίδευση νευρωνικών δικτύων στην πράξη, ωστόσο στην περίπτωση του max-min TP έχουμε στη

διάθεσή μας ακόμα ένα εργαλείο το οποίο στηρίζεται στην θεωρία βελτιστοποίησης.
Όταν έχουμε να κάνουμε με ένα γραμμικό νευρώνα, του οποίου η συνάρτηση πεδίου είναι στη μορφή

$$f(\mathbf{x}) = b + \sum_{i=1}^{n} w_i x_i$$

και έχει συνάρτηση ενεργοποίησης κάποια $\phi(v)$, μπορούμε να εκφράσουμε το πρόβλημα της δυαδικής ταξινόμησης ως εξής:

$$\begin{aligned} &\text{Minimize } J(\boldsymbol{w}, \mathbf{x}^{(k)}) \\ &\text{s.t. } b + \sum_{i=1}^{n} w_i x_i^k \geq \theta && \mathbf{x}^{(k)} \in \mathcal{C}_1 \\ &\quad b + \sum_{i=1}^{n} w_i x_i^k < \theta && \mathbf{x}^{(k)} \in \mathcal{C}_0 \end{aligned}$$

όπου J είναι κάποιο κριτήριο κόστους εξαρτώμενο από τα βάρη $\boldsymbol{w}$ προς προσδιορισμό και τα πρότυπα εκπαίδευσης. Μια γνωστή περίπτωση προβλήματος βελτιστοποίησης που εφαρμόζεται στην αναγνώριση προτύπων είναι αυτή της γραμμικής μηχανής διανυσμάτων υποστήριξης (Linear Support Vector Machine), στην οποία τα πρότυπα που ανήκουν στην κλάση $\mathcal{C}_1$ έχουν επιθυμητή έξοδο $y^k = 1$ και τα πρότυπα στην κλάση $\mathcal{C}_0$ έχουν $y^k = -1$. Επιλέγοντας ως $\phi(v) = \mathbf{sign}(v)$, συνήθως καταλήγουμε στο εξής πρόβλημα προς βελτιστοποίηση:

$$\begin{aligned} &\text{Minimize } ||\boldsymbol{w}||_2^2 + C \sum_k \xi_k \\ &\text{s.t. } y^{(k)} \cdot f\left(\mathbf{x}^{(k)}\right) \geq 1 - \xi_k \\ &\quad \xi_k \geq 0 \end{aligned} \tag{5.29}$$

Οι μεταβλητές ξ_k (γνωστές ως μεταβλητές χαλάρωσης - slack variables) έχουν εισαχθεί στους περιορισμούς έτσι ώστε οι ανισότητες να είναι ικανοποιήσιμες ακόμα και για μη-γραμμικά διαχωρίσιμα πρότυπα, εφόσον το ξ_k τεθεί αρκετά μεγάλο. Η C είναι μια παράμετρος που ελέγχει το "tradeoff" μεταξύ της ικανοποίησης των περιορισμών και του κόστους $||\boldsymbol{w}||_2^2$. Για $C \uparrow$, δίνεται έμφαση στο να ταξινομηθούν σωστά περισσότερα πρότυπα, ενώ για $C \downarrow$ στο να βρεθεί μια "καλή" διαχωριστική επιφάνεια (πιθανώς επιτρέποντας κάποια misclassifications). Για λεπτομέρειες σχετικά με τα SVMs ο αναγνώστης παραπέμπεται στα [11, 29].
Το πρόβλημα προς βελτιστοποίηση (5.29) είναι ένα πρόβλημα ελαχιστοποίησης μιας τετραγωνικής (quadratic) συνάρτησης με γραμμικούς περιορισμούς ως προς $\boldsymbol{w}$, δηλαδή με κυρτή περιοχή λύσεων (feasible region). Επομένως, για δεδομένη τιμή της παραμέτρου C αποτελεί πρόβλημα κυρτής βελτιστοποίησης με μοναδική βέλτιστη λύση. Στην περίπτωση του max-min TP, γνωρίζουμε ήδη πως οι εκφράσεις

$$\delta(\mathbf{x}) = \bigvee_{i=1}^{n} (w_i + x_i)$$

$$\varepsilon(\mathbf{x}) = \bigwedge_{i=1}^{n}(m_i + x_i)$$

είναι μη-γραμμικές ως προς $\boldsymbol{w}, \boldsymbol{m}$ και επομένως ανισότητες της μορφής

$$\lambda\delta(\mathbf{x}) + (1-\lambda)\varepsilon(\mathbf{x}) \geq \alpha$$
$$\lambda\delta(\mathbf{x}) + (1-\lambda)\varepsilon(\mathbf{x}) \leq \beta$$

δεν αποτελούν περιορισμούς με κυρτό feasible region, οπότε δεν μπορούμε να ελπίζουμε να καταλήξουμε μέσω αυτών σε κάποιο πρόβλημα κυρτής βελτιστοποίησης με μοναδικη λύση, όπως στην περίπτωση του linear SVM. Ωστόσο, μπορούμε να εκμεταλλευτούμε και πάλι την τεχνική του convex-concave programming, καταλήγοντας στο πρόβλημα προς βελτιστοποίηση που περιγράφεται από την εξίσωση (5.30), καθώς η έξοδος ενός max-min TP αποτελείται από το τη διαφορά 2 κυρτών συναρτήσεων και άρα εμπίπτει στην οικογένεια προβλημάτων που χειρίζεται η τεχνική του CCP.

$$\begin{aligned} \text{Minimize } & \sum_k \nu_k \cdot \max(0, \xi_k) && && (5.30)\\ \text{s.t. } & \lambda \cdot \delta(\mathbf{x}) \geq (\lambda - 1) \cdot \varepsilon(\mathbf{x}) - \xi_k && \mathbf{x}^{(k)} \in \mathcal{C}_1 \\ & \lambda \cdot \delta(\mathbf{x}) \leq (\lambda - 1) \cdot \varepsilon(\mathbf{x}) + \xi_k && \mathbf{x}^{(k)} \in \mathcal{C}_0 \end{aligned}$$

Στο (5.30), οι συντελεστές ν_k είναι συντελεστές ομαλοποίησης, όπως έχουν περιγραφεί στην ενότητα 4.1.4 για την περίπτωση γραμμικώς μη-διαχωρίσιμων κλάσεων κατά την τροπική έννοια. Η μορφή των ανισοτήτων, στις οποίες οι 2 κυρτές συναρτήσεις διαχωρίζονται στα 2 μέλη της ανισότητας, είναι απαραίτητη για την εκμετάλλευση του solver που παρέχουν οι συγγραφείς του [59], ο οποίος χρησιμοποιεί ανάλυση καμπυλότητας (curvature) καθώς στηρίζεται στην τεχνική DCP για την επίλυση του προβλήματος.

Αντίστοιχα μπορούμε να αντιμετωπίσουμε και μια περίπτωση που αποτελεί γενίκευση του max-min perceptron, χρησιμοποιώντας αυτή τη φορά maxout units. Συγκεκριμένα ορίζουμε το ακόλουθο μοντέλο:

$$g(\boldsymbol{x}) = \lambda\left(\bigvee_{i=1}^{K_{\max}} c_i + \boldsymbol{W}_i^T\boldsymbol{x}\right) + (1-\lambda)\left(\bigwedge_{i=1}^{K_{\min}} d_i + \boldsymbol{M}_i^T\boldsymbol{x}\right), \quad \lambda \in [0,1] \tag{5.31}$$

Στην περίπτωση του (5.31), ο αλγόριθμος εκπαίδευσης και πάλι μπορεί να χρησιμοποιεί κάποια μέθοδο κατάβασης κλίσεων, ωστόσο παρατηρώντας ότι το πρόβλημα παραμένει ένα στιγμιότυπο από Difference-of-Convex (DC) programming, μπορεί να εφαρμοστεί ακριβώς η τεχνική του CCP εκπεφρασμένη όπως στο πρόβλημα (5.30), με τα δ, ε να είναι πλέον:

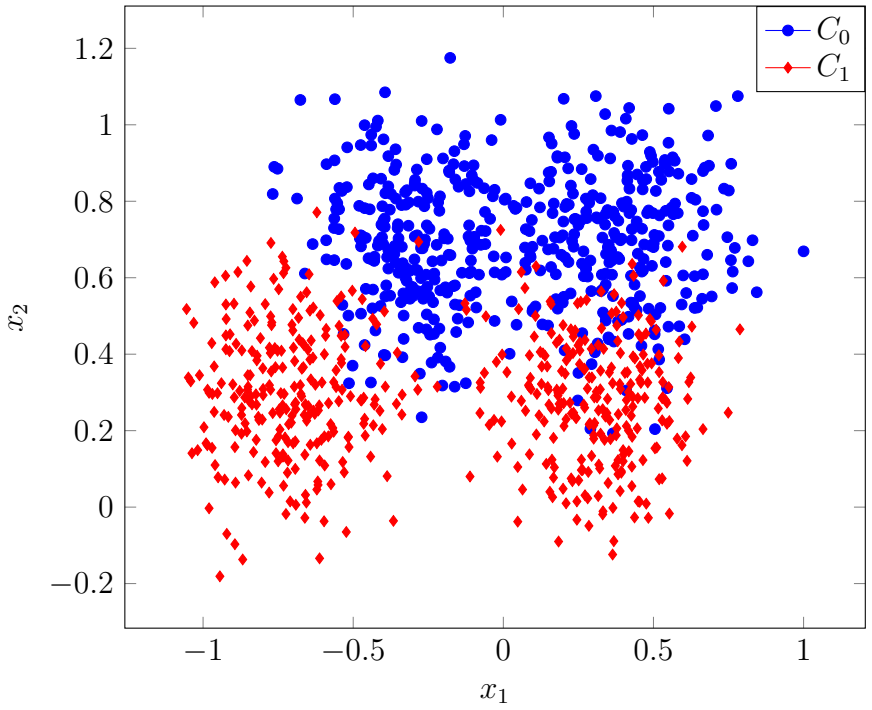
$$\delta(\boldsymbol{x}) = \bigvee_{i=1}^{K_{\max}} c_i + \boldsymbol{W}_i^T\boldsymbol{x}$$
$$\varepsilon(\boldsymbol{x}) = \bigwedge_{i=1}^{K_{\min}} d_i + \boldsymbol{M}_i^T\boldsymbol{x}$$

5.2.3 Πειραματισμός σε Datasets

Στη συνέχεια, παρουσιάζουμε τα αποτελέσματα της εφαρμογής του tropical max-min perceptron σε κάποια γνωστά datasets που προορίζονται για binary classification.

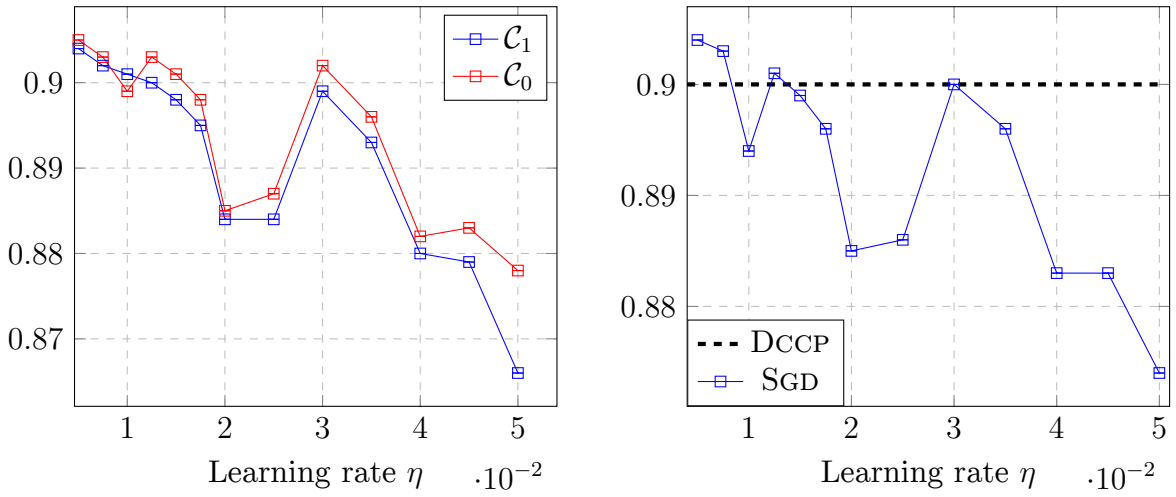
Ripley's synthetic dataset

Το συνθετικό dataset που είναι γνωστό με τον άνωθι τίτλο εμφανίστηκε στο [52] και αποτελεί ένα σύνολο δεδομένων για την αξιολόγηση γραμμικών ταξινομητών. Αποτελείται από ένα σύνολο προτύπων τα οποία χωρίζονται σε 2 κλάσεις, καθεμία εκ των οποίων έχει παραχθεί με τυχαία δειγματοληψία από ένα 2-part gaussian mixture. Χωρίζεται σε 250 training και 1000 testing πρότυπα, τα οποία *δεν είναι γραμμικώς διαχωρίσιμα*. Μια γραφική απεικόνιση ολόκληρου του dataset φαίνεται στην Εικόνα 5.10.



Σχήμα 5.10: Ripley's synthetic dataset

Τα αποτελέσματα του min-max perceptron για $k = 100$ δοκιμές με τυχαία αρχικοποίηση φαίνονται στην Εικόνα 5.11. Χρησιμοποιούμε stochastic gradient descent και $N = 100$ εποχές εκπαίδευσης, ενώ χρησιμοποιούμε τον ήδη υπάρχοντα εντός του dataset διαχωρισμό σε training και testing δεδομένα. Τέλος, η διακεκομμένη μαύρη γραμμή δείχνει τη μέση ακρίβεια με τη χρήση Convex-Concave Programming, στον οποία ο ρυθμός μάθησης είναι εσωτερική παράμετρος.
Το εύρος τιμών της παραμέτρου μάθησης για την περίπτωση του gradient descent επιλέχθηκε με βάση τυπικά εύρη τιμών που επιλέγονται για εκπαίδευση με κατάβαση κλίσεων σε αντίστοιχα γραμμικά (single ή multilayer) perceptrons. Παρατηρούμε ότι η απόδοση του tropical perceptron φαίνεται να



(α′) F1 Score **(β′)** Accuracy

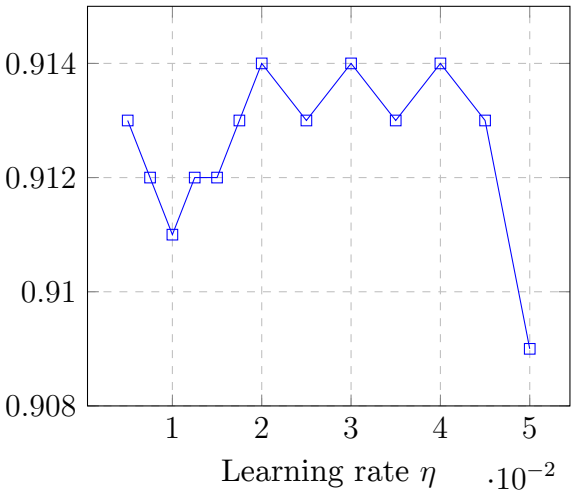
Σχήμα 5.11: `Ripleys` classification results

μην επηρεάζεται ιδιαίτερα από το ρυθμό μάθησης, ωστόσο παρατηρείται μια πτωτική τάση όταν αυτός κινείται προς υψηλότερες τιμές. Ακόμη, η τυπική απόκλιση των παραπάνω αποτελεσμάτων βρίσκεται σε επίπεδα χαμηλότερα του 10^{-3}. Για την εκπαίδευση με Disciplined Convex-Concave Programming, επιλέξαμε $\tau_{\max} = 0.01$ και κριτήριο τερματισμού $\varepsilon \leq 10^{-3}$ για τη διαφορά κόστους μεταξύ 2 διαδοχικών επαναλήψεων.

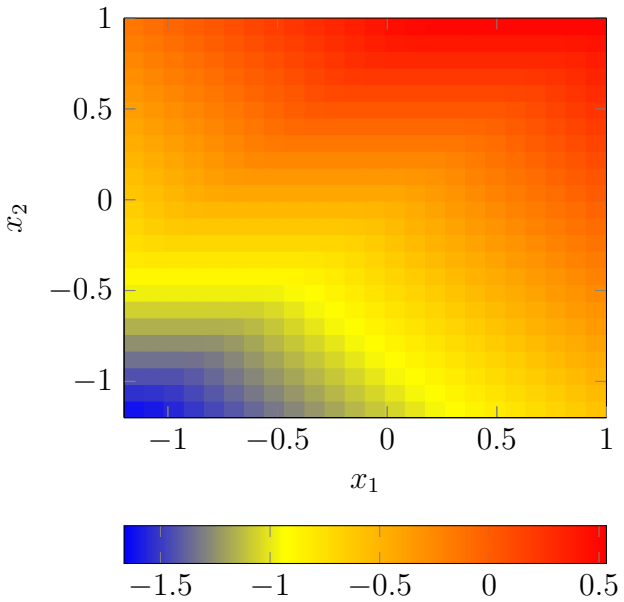
Όταν $\alpha_t \simeq 0.1$, η συμπεριφορά του perceptron αρχίζει να εμφανίζει σημάδια εγκλωβισμού σε τοπικά ελάχιστα, επομένως συμπεραίνουμε ότι το διάστημα $\alpha_t \in [0.005, 0.05]$ ενδείκνυται ως επιλογή αρχικής τιμής παραμέτρου μάθησης σε περίπτωση που η χρήση κάποιας adaptive εκδοχής του gradient descent είναι επιθυμητή.

Επιπροσθέτως, στην Εικόνα 5.12 φαίνεται η καλύτερη ακρίβεια που επιτεύχθηκε για κάθε επιλογή του α_t. Σημειώνεται ότι, θεωρώντας την περίπτωση της εκπαίδευσης με gradient descent, η απόδοση του Tropical Max-Min Perceptron για το Ripley's dataset ανταγωνίζεται (και σε περιπτώσεις ενδεχομένως να υπερτερεί σχετικά με) διάφορα μοντέλα που έχουν προταθεί στη βιβλιογραφία. Για παράδειγμα, *με βάση τα αποτελέσματα που αναφέρονται στο [3]*, υπερτερεί του SLMP που προτείνεται στο [53], καθώς και του morphological perceptron με ανταγωνιστική μάθηση που παρουσιάζεται στο [62]. Υπερτερεί ακόμα σε σχέση με τεχνικές που χρησιμοποιούν decision trees ή classifiers που στηρίζονται σε fuzzy lattices [51, 33]. Φαίνεται να υστερεί μόνο σε σχέση με υλοποιήσεις που στηρίζονται σε Support Vector Machines (SVM-L, SVM-RBF).

Η εκπαίδευση με disciplined convex-concave programming δίνει σχεδόν σταθερή ακρίβεια που κυμαίνεται στο $\simeq 90\%$, ενώ σε καμία περίπτωση το πλήθος εποχών εκπαίδευσης δεν ξεπέρασε τις 10. Επομένως, φαίνεται πως η χρήση convex-concave programming αποτελεί μια καλή εναλλακτική για νευρωνικά δίκτυα αυτής της μορφής, αφού με ελάχιστες επαναλήψεις συγκριτικά με τη μέθοδο κατάβασης κλίσεων η ακρίβεια κατηγοριοποίησης είναι συγκρίσιμη με τις καλύτερες επιδόσεις της μεθόδου.



Σχήμα 5.12: Best result on accuracy

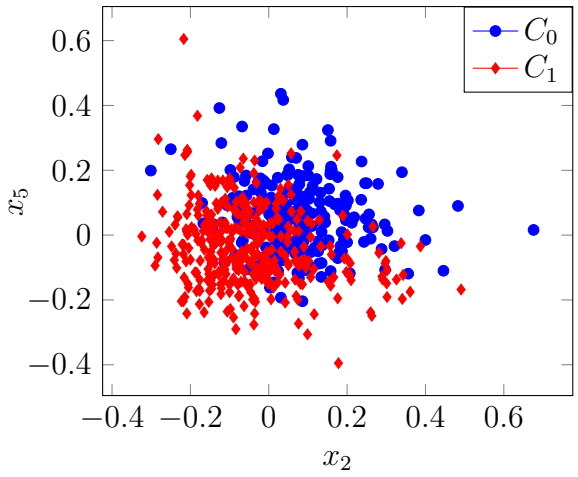


Σχήμα 5.13: DCCP output, `Ripleys` dataset

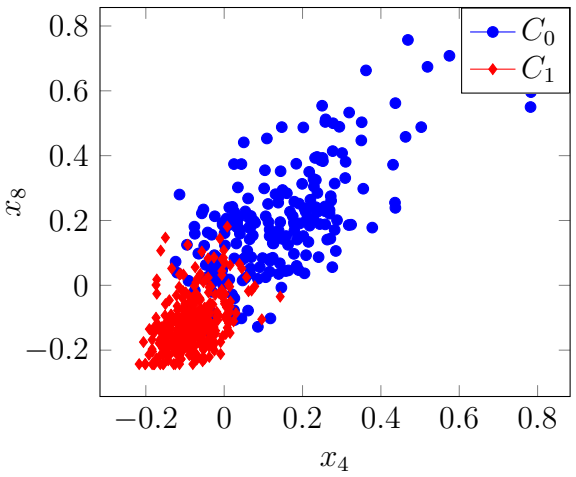
Wisconsin breast cancer dataset

Το Winsconsin breast cancer dataset είναι ένα σύνολο από 569 πρότυπα τα οποία περιέχουν features που έχουν εξαχθεί από εικόνες που χρησιμοποιούνται για τη διάγνωση καρκίνου του μαστού. Έχουν χωριστεί σε 2 κλάσεις που αντιστοιχούν σε καλοήθεις και κακοήθεις περιπτώσεις αντίστοιχα, ενώ κάθε πρότυπο περιγράφεται από 30 features. Στη βιβλιογραφία συνήθως χρησιμοποιούνται τα πρώτα 10 εξ'αυτών, οπότε ακολουθούμε αυτή τη σύμβαση και στο δικό μας πείραμα. Σημειώνεται ότι οι 2 κλάσεις *δεν είναι γραμμικώς διαχωρίσιμες*. Στην Εικόνα 5.14, φαίνονται τα διαγράμματα 2 τυχαίων επιλογών από συνδυασμούς features για το WDBC dataset.

Για την αξιολόγηση της απόδοσης του max-min perceptron, χωρίζουμε κάθε κλάση τυχαία σε 70% training - 30% testing δεδομένα. Στην εκπαίδευση, χρησιμοποιούμε stochastic gradient descent, δηλαδή κατάβαση κλίσεων όπου τα πρότυπα παρουσιάζονται με τυχαία σειρά στο perceptron και οι παράμετροι αναπροσαρμόζονται μετά από κάθε πρότυπο (σε αντίθεση με τη batch μέθοδο, όπου οι παράμετροι αναπροσαρμόζονται μετά από την παρουσίαση ενός συνόλου προτύπων). Στην εικόνα 5.15α΄ φαίνεται η μέση ακρίβεια κατηγοριοποίησης (μαζί με την τυπική απόκλιση) για ένα εύρος τιμών της παραμέτρου μάθησης.
Η διακεκομμένη γραμμή στην Εικόνα 5.15α΄ δείχνει την μέση ακρίβεια που επιτεύχθηκε με τη χρήση του Convex-Concave programming ως μεθόδου εκπαίδευσης (με σχεδόν μηδενική τυπική απόκλιση). Και σε αυτή την περίπτωση, όπως και στην περίπτωση του `Ripleys` dataset, ο απαιτούμενος αριθμός επαναλήψεων δεν υπερέβη ποτέ τις 10, ενώ παρατηρούμε ξανά πως η ακρίβεια είναι συγκρίσιμη με τα καλύτερα αποτελέσματα της gradient descent μεθόδου.

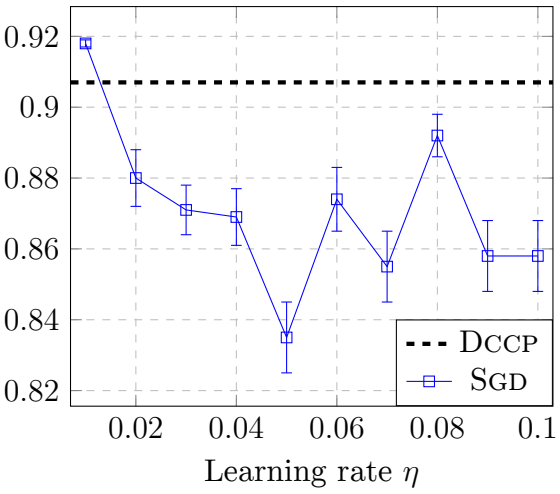


(α′) WDBC data - features 2 and 5

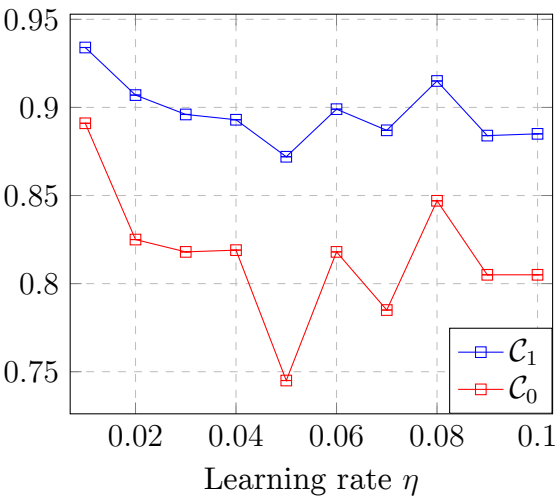


(β′) WDBC data - features 4 and 8

Σχήμα 5.14: WDBC dataset, two choices of feature plots



(α′) WDBC classification accuracy

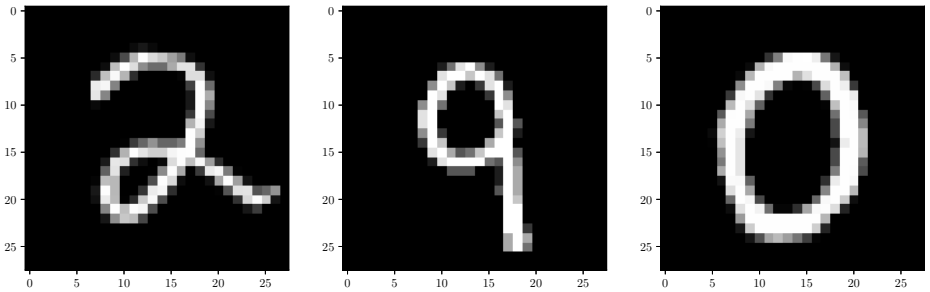


(β′) WDBC Classification F1 Score

Σχήμα 5.15: `WDBC` classification results

MNIST dataset

Στη συνέχεια, χρησιμοποιήσαμε ένα μορφολογικό perceptron για την αναγνώριση και ταξινόμηση ψηφίων από το σύνολο δεδομένων γνωστό ως MNIST [36] dataset, το οποίο χρησιμοποιήθηκε για ως benchmark στη δημοσίευση των LeCun et. al [35], η οποία εισήγαγε το πρώτο μοντέλο συνελικτικού νευρωνικού δικτύου. Το dataset αυτό αποτελείται από 70000 εικόνες μεγέθους 28×28 οι οποίες απεικονίζουν χειρόγραφα ψηφία από το 0 έως και το 9, και χρησιμοποιείται πάρα πολύ συχνά ως benchmark για μεθόδους μηχανικής μάθησης, ιδιαίτερα αυτές που εμπλέκουν βαθείς αρχιτεκτονικές νευρωνικών δικτύων (deep neural nets). Στην Εικόνα 5.16 φαίνονται 3 παραδείγματα από πρότυπα του υποσυνόλου αξιολόγησης (testing) του `MNIST` dataset.

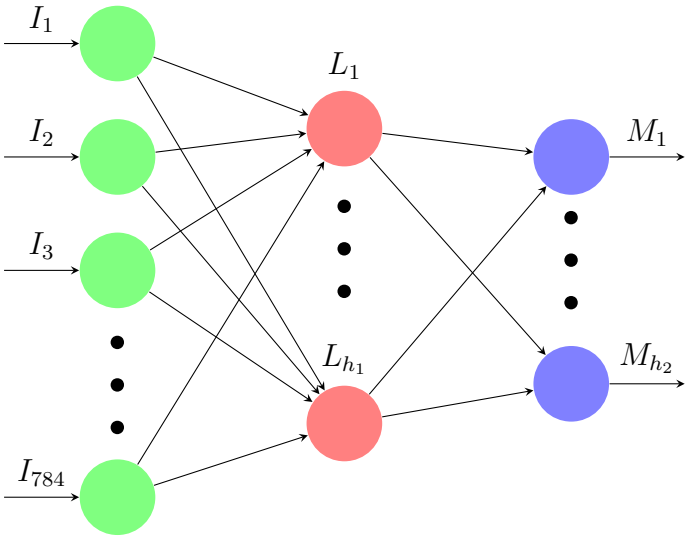


Σχήμα 5.16: Παραδείγματα εικόνων ψηφίων από το `MNIST` dataset

Η αρχιτεκτονική που χρησιμοποιήσαμε αποτελείται από ένα πλήρως συνδεδεμένο κρυφό επίπεδο από γραμμικούς νευρώνες χωρίς κάποια μη-γραμμική συνάρτηση ενεργοποίησης, και ένα μορφολογικό επίπεδο πριν την έξοδο. Καθώς στο σύνολο MNIST ζητούμενο είναι η ανάθεση ενός class label σε κάθε ψηφίο, από ένα σύνολο $\mathcal{C} = \{0, 1, \dots 9\}$, ενδείκνυται να χρησιμοποιήσουμε τη συνάρτηση ενεργοποίησης softmax στο επίπεδο εξόδου, η οποία ορίστηκε στην Ενότητα 2.2.1. Σαν συνάρτηση κόστους για την εκπαίδευση, ενδείκνυται να χρησιμοποιήσουμε την διασταυρωμένη εντροπία (*cross-entropy*), η οποία αναφέρεται συχνά στη βιβλιογραφία ως η καλύτερη επιλογή για softmax μοντέλα [20]. Τέλος, προκειμένου να αποκτήσουμε μια εποπτεία σχετικά με το είδος των χαρακτηριστικών που θα οδηγηθεί να ανακαλύψει ένα μορφολογικό μοντέλο, επιλέξαμε συνειδητά να χρησιμοποιήσουμε μονάχα dilation ή erosion στο μορφολογικό επίπεδο, δηλαδή $\lambda \in \{0, 1\}$. Η συνολική αρχιτεκτονική φαίνεται στο επόμενο σχήμα: Όπως φαίνεται, το πρώτο επίπεδο αποτελείται από h_1 γραμμικούς νευρώνες οι οποίοι καταλήγουν σε h_2 μορφολογικούς νευρώνες με έξοδο τύπου softmax. Έτσι, ο νευρώνας M_i του μορφολογικού επιπέδου υπολογίζει την παράσταση

$$h_{M_i}(\mathbf{x}) = \text{softmax}\left((\boldsymbol{w}^i)^T \boxplus \boldsymbol{L}\right) = \text{softmax}\left(\bigvee_{j=1}^{h_1} w_j^i + L_j(\mathbf{x})\right) \tag{5.32}$$

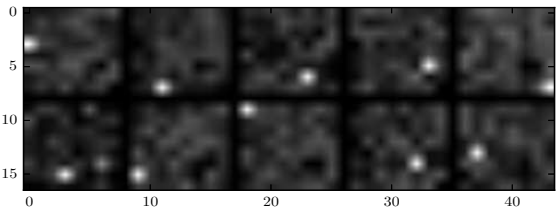
όπου w_j^i το j-οστό στοιχείο του διανύσματος παραμέτρων $\boldsymbol{w}^i$ που αντιστοιχεί στο νευρώνα M_i, και $L_j(\vec{x}) \in \mathbb{R}$ συμβολίζει την απόκριση του j-οστού γραμμικού νευρώνα του πρώτου επιπέδου. Ο κώδικας των πειραμάτων είναι ελεύθερα διαθέσιμος και στηρίζεται σε μοντέλα που έχουν κατασκευαστεί με τη βοήθεια της βιβλιοθήκης `Theano` [2], η οποία είναι μια βιβλιοθήκη symbolic computation της `Python`.



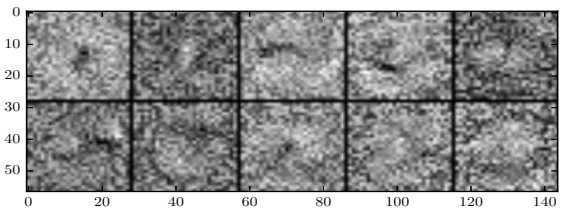
Σχήμα 5.17: Αρχιτεκτονική δικτύου για το σύνολο χειρόγραφων ψηφίων `MNIST`

Dilation Layer - $\lambda = 1$

Εκπαιδεύοντας το νευρωνικό για $n = 100$ εποχές στο πλήρες dataset των 50000 ψηφίων, και χωρίς να χρησιμοποιήσουμε κάποια μέθοδο validation (όπως π.χ. k-way folding ή early stopping), η καλύτερη ακρίβεια ταξινόμησης που επιτυγχάνουμε είναι 92.1%, για επιλογή παραμέτρων $n_1 = 64, n_2 = 10$. Στην εικόνα 5.18, φαίνονται τα διανύσματα παραμέτρων $\boldsymbol{w}^{(i)}, i \in 1 \dots n_2$ που αντιστοιχούν στους μορφολογικούς νευρώνες. Το πλέγμα που απεικονίζεται δείχνει τα 10 μορφολογικά



Σχήμα 5.18: Απεικόνιση των παραμέτρων του Dilation layer



Σχήμα 5.19: Απεικόνιση των L_i που αντιστοιχούν στις μέγιστες ενεργοποιήσεις

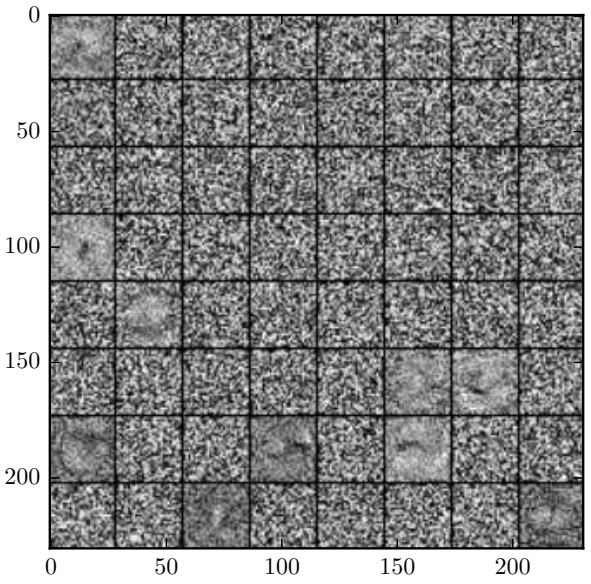
φίλτρα, για τα οποία το διάνυσμα βαρών $\boldsymbol{w} \in \mathbb{R}^{1\times 64}$ έχει μετασχηματιστεί στην grayscale εικόνα $I_{\boldsymbol{w}} \in \mathbb{R}^{8\times 8}$ κανονικοποιώντας τα βάρη στο διάστημα $[0, 1]$. Παρατηρούμε άμεσα ότι κάθε φίλτρο φαίνεται να επικεντρώνει μονάχα σε ένα χαρακτηριστικό του προηγούμενου επιπέδου (το οποίο αντιστοιχεί στο έντονα φωτεινό pixel κάθε εικόνας). Σε μια προσπάθεια να αποκτήσουμε περαιτέρω διαίσθηση, απεικονίζουμε τα γραμμικά φίλτρα στο πρώτο επίπεδο, με το αποτέλεσμα να φαίνεται στην εικόνα 5.20. Αν και εκ πρώτης όψεως δε μπορούμε να βγάλουμε συγκεκριμένα συμπεράσματα, επικεντρώνουμε την προσοχή μας το γεγονός ότι οι περισσότερες εικόνες εμφανίζουν μεγάλη τυχαιότητα ως προς τις τιμές τους, ομοιάζοντας με θόρυβο, εκτός από ελάχιστες. Έτσι, απομονώνουμε

τα φίλτρα του πρώτου επιπέδου, η έξοδος των οποίων αντιστοιχεί σε κάποιον από τους maximal συντελεστές που φαίνονται στην εικόνα 5.18, τα οποία φαίνονται στην εικόνα 5.19. Η εικόνα 5.19 αποτελείται από 10 ξεχωριστά φίλτρα $L_{\max}$ (δηλαδή σε κάθε μορφολογικό φίλτρο αντιστοιχεί ένα ξεχωριστό μεγιστικό feature από το προηγούμενο επίπεδο), και μάλιστα κανένα από αυτά τα φίλτρα δεν επιδεικνύει την noise-like εικόνα των περισσότερων φίλτρων της εικόνας 5.20. Η διαίσθησή μας μας λέει πως ίσως να μπορούμε να "ξεφορτωθούμε" πλήρως τα υπόλοιπα φίλτρα $L_i \setminus L_{\max}$. Θέτοντας $W_{L_i} = 0, b_{L_i} = -\infty$ για κάθε τέτοιο φίλτρο, ώστε η έξοδός τους να μην λαμβάνεται καθόλου υπόψιν στο μορφολογικό επίπεδο του νευρωνικού, επιβεβαιώνουμε τη διαίσθησή μας (βλ. πίνακα 5.1) - η περίπτωση που περιγράψαμε μόλις αντιστοιχεί στον τύπο `Sparse`. Για διαφορετικές περιπτώσεις n_1, το

n_1	Accuracy	Accuracy (`Sparse`)	# Active Filters
24	84.29%	84.28%	17
32	84.84%	84.85%	15
48	84.63%	84.61%	18
64	92.10%	92.07%	10

Πίνακας 5.1: Συγκριτικά αποτελέσματα των 2 δικτύων, $\lambda = 1$

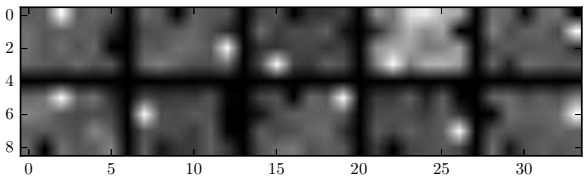
επίπεδο του dilation φάνηκε να λαμβάνει υπόψιν 1 ή 2 γραμμικά φίλτρα για κάθε ψηφίο. Η τεχνική που ακολουθήσαμε για να εξάγουμε τα αποτελέσματα του Πίνακα 5.1 ήταν να κρατάμε, για κάθε ψηφίο, έναν σταθερό αριθμό γραμμικών φίλτρων στο προηγούμενο επίπεδο "ενεργό". Πρακτικά, κρατώντας μόλις τα 2 φίλτρα με το ισχυρότερο βάρος, επιτυγχάνουμε σχεδόν την ίδια ακρίβεια ταξινόμησης.



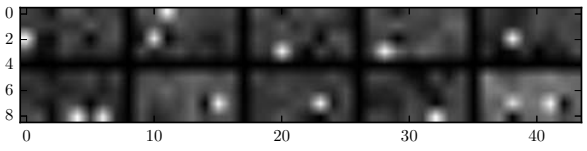
Σχήμα 5.20: Απεικόνιση βαρών για τα φίλτρα $L_i, i \in 1 \dots n_1$ της Εικόνας 5.17

Erosion Layer - $\lambda = 0$

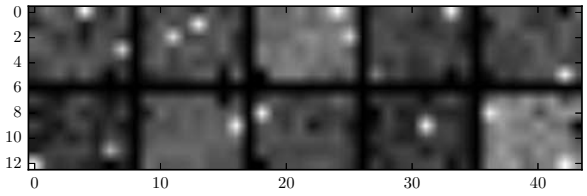
Επαναλάβαμε το πείραμα χρησιμοποιώντας την αρχιτεκτονική που φαίνεται στην Εικόνα 5.17, χρησιμοποιώντας πλέον $(\min, +)$ perceptrons στο μορφολογικό επίπεδο. Το πλήθος επαναλήψεων διατηρείται σταθερό $n = 100$ χωρίς early stopping ή cross validation. Τα διανύσματα των φίλτρων φαίνονται στην Εικόνα 5.21 για διάφορες επιλογές του n_1. Και σε αυτή την περίπτωση, παρατηρούμε



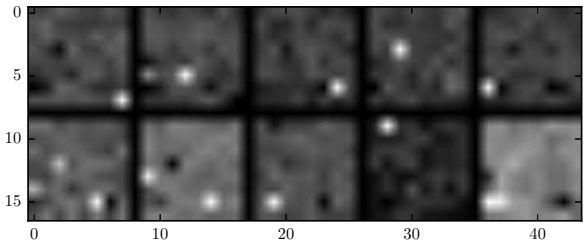
(α′) Erosion Layer, $n_1 = 24$



(β′) Erosion Layer, $n_1 = 32$



(γ′) Erosion Layer, $n_1 = 48$



(δ′) Erosion Layer, $n_1 = 64$

Σχήμα 5.21: Erosion Layers

πως για όλες τις επιλογές του n_1, το μορφολογικό φίλτρο κάθε ψηφίου και πάλι εμφανίζει δυσανάλογα ενισχυμένα 1 ή 2 βάρη. Επαναλαμβάνοντας τη διαδικασία της απομάκρυνσης των φίλτρων του 1ου επιπέδου που δεν αντιστοιχούν στους μη-μέγιστους συντελεστές, λαμβάνουμε τα αποτελέσματα του Πίνακα 5.2. Αν και θα μπορούσαμε να παρατηρήσουμε ότι το Erosion Layer δίνει γενικά πιο συνεπή

Πίνακας 5.2: Συγκριτικά αποτελέσματα των 2 δικτύων, $\lambda = 0$

n_1	Accuracy	Accuracy (`Sparse`)	# Active Filters
24	86.82%	86.82%	9
32	92.03%	92.05%	14
48	90.16%	90.11%	12
64	90.65%	90.66%	13

ακρίβεια από το Dilation Layer, τα αποτελέσματα των Πινάκων (5.1, 5.2) δεν αποτελούν benchmarks, αλλά είναι ενδεικτικά ώστε να φανεί η τάση των μορφολογικών επιπέδων για αραιές ενεργοποιήσεις του χώρου χαρακτηριστικών.

Κεφάλαιο 6

Επίλογος

6.1 Ερευνητική Συμβολή

Η ολοκλήρωση της παρούσας εργασίας δεν συμπίπτει σε καμία περίπτωση με την εξάντληση του ερευνητικού θέματος με το οποίο καταπιάστηκε, αλλά θέτει ένα νοητό όριο σε έναν κύκλο πρώιμης ενασχόλησης με ζητήματα σχετικά με μορφολογικές μηχανές μάθησης, τη θεωρία που τις διέπει αλλά και πρακτικούς αλγορίθμους που μας επιτρέπουν να τις χρησιμοποιούμε ως ένα ακόμη βέλος στη φαρέτρα της μηχανικής μάθησης. Κλείνοντας τον προλεγόμενο κύκλο, μπορούμε να εντοπίσουμε την ερευνητική συμβολή αυτής της εργασίας γύρω από 3 βασικούς άξονες:

- Δεδομένου ενός μορφολογικού ταξινομητή, η χρήση ενός αλγορίθμου εκπαίδευσης που επιλύει το πρόβλημα (4.17), το οποίο αποτελεί ένα στιγμιότυπο ενός δομημένου μη-κυρτού προβλήματος βελτιστοποίησης, προσφέρει μια χρήσιμη εναλλακτική στις κατασκευαστικές μεθόδους που κυριαρχούν στη σχετική βιβλιογραφία. Τέτοιες μέθοδοι είναι πιθανό να οδηγήσουν στο σχηματισμό εξαιρετικά πολύπλοκων επιφανειών απόφασης, οι οποίες είτε είναι περιττές για το μεγαλύτερο κομμάτι του προβλήματος είτε έχουν σχηματιστεί υπό την επιρροή outliers, κάτι το οποίο ο προτεινόμενος αλγόριθμος φαίνεται να αποφεύγει σε μεγάλο βαθμό.

  Ακόμη, δεδομένου ενός συνόλου προτύπων που προορίζονται για δυαδική ταξινόμηση, ο Αλγόριθμος 6 μας επιτρέπει να αποφανθούμε σε χρόνο πολυωνυμικό ως προς το πλήθος των προτύπων εάν αυτά μπορούν να διαχωριστούν με τη χρήση του αντίστοιχου $(\max, +)$ perceptron. Όπως σχολιάζουμε και στο κυρίως κείμενο, ο αλγόριθμος μπορεί να προκύψει με φυσικό τρόπο χρησιμοποιώντας τη θεωρία της minimax άλγεβρας.

- Ένα μεγάλο κομμάτι της εργασίας επικεντρώνει στην ανάδειξη της σχέσης μεταξύ διαφόρων μοντέλων ταξινομητών και του πεδίου της τροπικής γεωμετρίας. Η ανάδειξη αυτή είναι χρήσιμη για 2 λόγους: αφενός, μέχρι ένα βαθμό ενοποιεί το θεωρητικό υπόβαθρο των μορφολογικών ταξινομητών, αφού η γεωμετρία που διέπει τις περιοχές απόφασης, τα σύνολα εφικτών αναθέσεων παραμέτρων κλπ. μέχρι σήμερα εξεταζόταν κυρίως στο βαθμό που εξυπηρετούσε την εκάστοτε πρακτική εφαρμογή. Αφετέρου, ακόμη και από μια πιο πρακτική σκοπιά, πρέπει να σχολιάσουμε ότι η ερευνητική κοινότητα ήδη χρησιμοποιεί μορφολογικά μοντέλα ή γενικεύσεις αυτών. Η ενοποίηση που προσπαθούμε να επιτύχουμε προσφέρει νέες μεθοδολογικές προσεγγίσεις σε προβλήματα που αφορούν "διαδεδομένες" οικογένειες ταξινομητών (π.χ. RelU ή Maxout επίπεδα νευρωνικών δικτύων), όπως το πρόβλημα της διαμέρισης του χώρου χαρακτηριστικών ενός πολύπλοκου ταξινομητή σε "απλές" (γραμμικές) περιοχές. Η Πρόταση 19 αποτελεί ένα πρωτότυπο ερευνητικό αποτέλεσμα σε αυτή την κατεύθυνση, η οποία ταυτόχρονα προσφέρει και έναν αλγόριθμο για την απλοποίηση τροπικών πολυωνύμων πολλών μεταβλητών.

- Τελικά, η γενικότερη ενασχόλησή μας γύρω από τα προβλήματα βελτιστοποίησης που συνοδεύουν την εκπαίδευση ταξινομητών, μας ώθησε στην εξέταση μιας πρόσφατα προτεινόμενης μεθόδου που προσανατολίζεται σε βελτιστοποίηση μη-κυρτών συναρτήσεων μέσω της αντικατάστασης της συνάρτησης κόστους με μια εξομαλυμένη εκδοχή της. Καθώς η εκδοχή αυτή δεν είναι πάντα εφικτό να υπολογιστεί αναλυτικά, δείχνουμε πως μέσω δειγματοληψίας (και με σχετικά μη-περιοριστικές υποθέσεις για τη συνάρτηση προς βελτιστοποίηση) αυτή μπορεί να προσεγγισθεί εντός μιας επιθυμητής ακρίβειας με αυθαίρετα μεγάλη πιθανότητα, εφόσον έχει επιλεχθεί ο κατάλληλος αριθμός δειγμάτων. Το βασικό αποτέλεσμα στο οποίο στηριζόμαστε είναι η Πρόταση 7.

Είναι σημαντικό να αναφέρουμε πως ένα κομμάτι της εργασίας, το οποίο σχετίζεται κυρίως με τους πρώτους 2 άξονες, θα παρουσιαστεί ως paper στο International Symposium of Mathematical Morphology (ISMM) το Μάιο του 2017.

6.2 Μελλοντικές επεκτάσεις

Όπως έχει αναφερθεί, υπάρχουν αρκετά περιθώρια για περαιτέρων ανάπτυξη των ιδεών και της προσέγγισης που ακολουθείται σε αυτή τη διπλωματική. Ομοίως με πριν, χωρίζουμε τα πιθανά επόμενα βήματα σε 3 κύριους πυλώνες:

- Η εκμετάλλευση της σχέσης μεταξύ τροπικής γεωμετρίας και στοιχειωδών ιδιοτήτων ταξινομητών που χρησιμοποιούνται ήδη ευρέως στη βιβλιογραφία, ενδέχεται να οδηγήσει σε νέα αποτελέσματα στο κομμάτι αυτό. Θα ήταν ενδιαφέρον να διερευνηθεί κατά πόσο είναι δυνατόν να αναπαραχθούν τα αποτελέσματα που εμφανίζονται στο [46] με μεθόδους που στηρίζονται καθαρά στη θεωρία της τροπικής γεωμετρίας. Η προσέγγιση αυτή ακολουθεί την "παράδοση" των Pachter & Sturmfels [49], Cueto & Sturmfels [12], όπου τα αναλυτικά εργαλεία της τροπικής γεωμετρίας χρησιμοποιούνται για να εξάγουν αποτελέσματα για τη διάσταση και την πολυπλοκότητα διαφόρων στατιστικών μοντέλων.

- Η καταλληλότητα των μορφολογικών ταξινομητών ως κομμάτια βαθύτερων ή εν γένει πιο πολύπλοκων ταξινομητών δεν έχει διερευνηθεί επαρκώς. Μια κατεύθυνση στην οποία ήδη εργαζόμαστε είναι η αντικατάσταση των εγγενώς μη-γραμμικών επιπέδων των συνελικτικών νευρωνικών δικτύων, τα οποία αποτελούν το state of the art σε διάφορα προβλήματα μηχανικής μάθησης, από μορφολογικά επίπεδα αποτελούμενα από $(\max, +)$ ή $(\min, +)$ φίλτρα. Για παράδειγμα, η αντικατάσταση ενός max-pooling φίλτρου από ένα φίλτρο dilation θα μπορούσε να οδηγήσει σε ένα συνελικτικό νευρωνικό δίκτυο το οποίο "προσαρμόζει" τον τρόπο που πραγματοποιεί το pooling σε διάφορες περιοχές της ενδιάμεσης αναπαράστασης.

- Όσον αφορά την βελτιστοποίηση μέσω ισοτροπικής διάχυσης, η χρησιμότητα της Πρότασης 7 είναι ακόμη περιορισμένη. Συγκεκριμένα, η Πρόταση μας παρέχει ένα κάτω φράγμα στο πλήθος των απαιτούμενων δειγμάτων για την επίτευξη της ζητούμενης ακρίβειας προσέγγισης του κόστους, ωστόσο το φράγμα αυτό μπορεί να είναι ιδιαίτερα μεγάλο για μεγάλες τιμές της παραμέτρου κλίμακας σ, που αναγκαστικά μας οδηγεί στην αναζήτηση ευριστικών ή ακριβών κανόνων για την επιλογή των παραμέτρων του αλγορίθμου προσέγγισης. Επιπλέον, όπως αναφέρεται και στο κυρίως κείμενο, ενώ η διάσταση d των δεδομένων φαινομενικά δεν επηρεάζει το πλήθος των δειγμάτων, συχνά συνοδεύεται από μεγαλύτερες σταθερές Lipschitz. Για τη θεραπεία αυτού του προβλήματος, μια υποσχόμενη οδός ενδέχεται να είναι η χρήση μεθόδων βελτιστοποίησης που διαμερίζουν το διάνυσμα χαρακτηριστικών $\mathbf{x} \in \mathbb{R}^d$ σε (συνήθως ξένα μεταξύ τους) σύνολα χαμηλότερης διάστασης, γνωστές στη βιβλιογραφία ως block coordinate minimization schemes. Μια πρόσφατα προτεινόμενη μέθοδος [16] η οποία πραγματοποιεί *ασύγχρονες* ανανεώσεις των παραμέτρων προς βελτιστοποίηση σε συνδυασμό με την ισοτροπική διάχυση πιθανώς να οδηγούσε σε ικανοποιητικούς χρόνους εκτέλεσης στην πράξη.

Appendix

Οργάνωση Κώδικα Αριθμητικών Πειραμάτων

Για τις ανάγκες της παρούσας εργασίας, αναπτύχθηκε στη γλώσσα `Python` μια βιβλιοθήκη που περιέχει τις υλοποιήσεις των μορφολογικών μοντέλων που παρουσιάζονται στα προηγούμενα κεφάλαια. Η βιβλιοθήκη χρησιμοποιεί το `Theano` [2] framework για συμβολικούς υπολογισμούς και είναι ελεύθερα διαθέσιμη στο `Github`[1]. Στο σημείο αυτό, παρουσιάζουμε συνοπτικά την οργάνωσή της και τη λειτουργικότητα που παρέχει.

Δομή Βιβλιοθήκης

Η δομή της βιβλιοθήκης φαίνεται στο παρακάτω διάγραμμα:

- `algorithms/`
  - `mcmc_diffusion.py`
- `models/`
  - `linear.py`
  - `tropical.py`
- `compose.py`
- `mpsvm.py`
- `pymp.py`
- `train_utils.py`
- `tropical_svm.py`
- `utils.py`

Η λειτουργικότητα της βιβλιοθήκης μοιράζεται ως εξής:

- `mpsvm.py`: Περιέχει τις υλοποιήσεις μορφολογικών perceptrons οι οποίες χρησιμοποιούν την Convex-Concave Procedure για την εκπαίδευση των μοντέλων.
- `pymp.py`: Περιέχει ορισμένες fallback υλοποιήσεις των μοντέλων που παρουσιάζονται στο Κεφάλαιο 4 οι οποίες χρησιμοποιούν gradient descent για την εκπαίδευση των μοντέλων.
- `tropical_svm.py`: Περιέχει την υλοποίηση του Tropical (Affine) SVM, όπως περιγράφεται στην τελευταία ενότητα του Κεφαλαίου 4, η οποία χρησιμοποιεί τη βιβλιοθήκη CvxPy [18] για εκπαίδευση.

[1] VHarisop/PyMP

- `utils.py`: Περιέχει διάφορες συναρτήσεις γενικού σκοπού, όπως για παράδειγμα συναρτήσεις για normalization/scaling των δεδομένων. Ακόμη, περιέχει την υλοποίηση του Αλγορίθμου 7 για την απομάκρυνση των μη-απαραίτητων όρων από max-πολυώνυμα.

- `models`: Περιέχει 2 αρχεία με υλοποιήσεις νευρώνων, οι οποίες βρίσκονται στα `linear.py`, `tropical.py`. Το πρώτο περιέχει γνωστές "γραμμικές" υλοποιήσεις, όπως perceptron με σιγμοειδή ενεργοποίηση κλπ., ενώ το δεύτερο περιέχει όσα μορφολογικά μοντέλα έχουν αναφερθεί στην εργασία. Καθένα εξ'αυτών έχει υλοποιηθεί ως κλάση η οποία κάνει expose τις εξής μεθόδους:

  1. `error(Y)`: μέθοδος που επιστρέφει (σε συμβολική μορφή) το σφάλμα της εξόδου του νευρώνα σχετικά με ένα διάνυσμα/πίνακα επιθυμητών εξόδων `Y`. Ο τύπος του σφάλματος ποικίλλει, ανάλογα με το μοντέλο που έχει υλοποιηθεί.
  2. `output_at(Xs)`: μέθοδος που επιστρέφει την απόκριση του νευρώνα σε ένα διάνυσμα εισόδων `Xs`.
  3. `validation_error(Y)`: μέθοδος που επιστρέφει (σε συμβολική μορφή) το σφάλμα κατηγοριοποίησης (0-1 loss) του νευρώνα σχετικά με ένα διάνυσμα/πίνακα επιθυμητών εξόδων. *Η υλοποίηση αυτής της μεθόδου είναι προαιρετική* και δεν υπάρχει σε όλα τα μοντέλα.

- `compose.py`: Στο αρχείο αυτό περιέχεται η κλάση `DeepMP`, η οποία υλοποιεί ένα νευρωνικό δίκτυο πολλαπλών επιπέδων. Δέχεται ως ορίσματα μια λίστα με μοντέλα νευρώνων τα οποία πρέπει να κάνουν expose τις 2 πρώτες από τις 3 μεθόδους που αναφέρθηκαν παραπάνω, καθώς και μια λίστα από ακέραιους αριθμούς που συμβολίζουν τη διάσταση κάθε επιπέδου. Η κλάση αυτή περιέχει μια σειρά από μεθόδους εκπαίδευσης (`train, train_nesterov, train_alternating`), οι οποίες στηρίζονται στις βοηθητικές συναρτήσεις στο αρχείο `train_utils.py`. Ο χρήστης μπορεί, περνώντας κατάλληλα ορίσματα, να επιλέξει ανάμεσα σε batch, minibatch, και fully stochastic εκπαίδευση.

Τονίζουμε πως οι υλοποιήσεις που παρέχονται δεν φιλοδοξούν σε καμία περίπτωση να είναι βέλτιστες, αλλά παρέχονται για λόγους αναπαραγωγής των πειραμάτων της εργασίας και με την ελπίδα να χρησιμοποιήσουν ως σημείο αναφοράς για οποιονδήποτε θελήσει να επεκτείνει τα αποτελέσματά της. Με εξαίρεση ζητήματα αποδοτικότητας, έχει γίνει προσπάθεια να ακολουθηθεί η φιλοσοφία της Pylearn2 [25] ως προς την ευκολία της επιλογής και σύνδεσης των επιμέρους μοντέλων που παρέχονται για σκοπούς prototyping.

Βιβλιογραφία

[1] Akian, M., Gaubert, S., and Guterman, A. Tropical polyhedra are equivalent to mean payoff games. *International Journal of Algebra and Computation 22*, 01 (2012), 1250001.

[2] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Bleecher Snyder, J., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., de Brébisson, A., Breuleux, O., Carrier, P.-L., Cho, K., Chorowski, J., Christiano, P., Cooijmans, T., Côté, M.-A., Côté, M., Courville, A., Dauphin, Y. N., Delalleau, O., Demouth, J., Desjardins, G., Dieleman, S., Dinh, L., Ducoffe, M., Dumoulin, V., Ebrahimi Kahou, S., Erhan, D., Fan, Z., Firat, O., Germain, M., Glorot, X., Goodfellow, I., Graham, M., Gulcehre, C., Hamel, P., Harlouchet, I., Heng, J.-P., Hidasi, B., Honari, S., Jain, A., Jean, S., Jia, K., Korobov, M., Kulkarni, V., Lamb, A., Lamblin, P., Larsen, E., Laurent, C., Lee, S., Lefrancois, S., Lemieux, S., Léonard, N., Lin, Z., Livezey, J. A., Lorenz, C., Lowin, J., Ma, Q., Manzagol, P.-A., Mastropietro, O., McGibbon, R. T., Memisevic, R., van Merriënboer, B., Michalski, V., Mirza, M., Orlandi, A., Pal, C., Pascanu, R., Pezeshki, M., Raffel, C., Renshaw, D., Rocklin, M., Romero, A., Roth, M., Sadowski, P., Salvatier, J., Savard, F., Schlüter, J., Schulman, J., Schwartz, G., Serban, I. V., Serdyuk, D., Shabanian, S., Simon, E., Spieckermann, S., Subramanyam, S. R., Sygnowski, J., Tanguay, J., van Tulder, G., Turian, J., Urban, S., Vincent, P., Visin, F., de Vries, H., Warde-Farley, D., Webb, D. J., Willson, M., Xu, K., Xue, L., Yao, L., Zhang, S., and Zhang, Y. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints abs/1605.02688* (May 2016).

[3] Araújo, R. d. A., Oliveira, A. L., and Meira, S. R. A hybrid neuron with gradient-based learning for binary classification problems. *Encontro Nacional de Inteligência Artificial-ENIA* (2012).

[4] Bishop, C. M. *Pattern recognition and machine learning*. Springer Science & Business Media, 2006.

[5] Björklund, H., Svensson, O., and Vorobyov, S. Controlled linear programming for infinite games. Tech. rep., Technical Report DIMACS-2005-13, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, 2005.

[6] Boissonnat, J.-D., and Yvinec, M. *Algorithmic geometry*. Cambridge university press, 1998.

[7] Borwein, J., and Lewis, A. S. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[8] Boyd, S., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

[9] Butkovič, P. *Max-linear systems: theory and algorithms*. Springer Science & Business Media, 2010.

[10] Calafiore, G. C. Parallel block coordinate minimization with application to group regularized regression. *Optimization and Engineering 17*, 4 (2016), 941–964.

[11] Cortes, C., and Vapnik, V. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[12] Cueto, M. A., Morton, J., and Sturmfels, B. Geometry of the restricted boltzmann machine. *Algebraic Methods in Statistics and Probability,(eds. M. Viana and H. Wynn), AMS, Contemporary Mathematics 516* (2010), 135–153.

[13] Cuninghame-Green, R. A. Projections in minimax algebra. *Mathematical Programming 10*, 1 (1976), 111–123.

[14] Cuninghame-Green, R. A. *Minimax Algebra, vol. 166 of Lecture Notes in Economics and Mathematical Systems*. Berlin, Germany: Springer-Verlag, 1979.

[15] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems* (2014), pp. 2933–2941.

[16] Davis, D. The asynchronous PALM algorithm for nonsmooth nonconvex problems. *arXiv preprint arXiv:1604.00526* (2016).

[17] Devroye, L., and Lugosi, G. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2012.

[18] Diamond, S., and Boyd, S. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research 17*, 83 (2016), 1–5.

[19] Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems* (2001), 472–478.

[20] Dunne, R. A., and Campbell, N. A. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 181* (1997), vol. 185, Citeseer.

[21] Freund, Y., and Schapire, R. E. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory* (1995), Springer, pp. 23–37.

[22] Gärtner, B., and Jaggi, M. Tropical support vector machines. Tech. Rep. ACS-TR-362502-01, 2008.

[23] Gaubert, S., and Katz, R. D. Minimal half-spaces and external representation of tropical polyhedra. *Journal of Algebraic Combinatorics 33*, 3 (2011), 325–348.

[24] Gondran, M., and Minoux, M. *Graphs, dioids and semirings: new models and algorithms*, vol. 41. Springer Science & Business Media, 2008.

[25] Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., and Bengio, Y. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214* (2013).

[26] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. C., and Bengio, Y. Maxout networks. *ICML (3) 28* (2013), 1319–1327.

[27] Grant, M., Boyd, S., and Ye, Y. Disciplined convex programming. In *Global optimization*. Springer, 2006, pp. 155–210.

[28] Grigg, N. B. *Factorization of Tropical Polynomials*. PhD thesis, Brigham Young University, 2007.

[29] Haykin, S. S. *Neural networks and learning machines*, vol. 3. Pearson Upper Saddle River, NJ, USA, 2009.

[30] Hazan, E., Levy, K. Y., and Shalev-Shwartz, S. On graduated optimization for stochastic non-convex problems. *arXiv preprint arXiv:1503.03712* (2015).

[31] Huber, P. J., et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics 35*, 1 (1964), 73–101.

[32] Izhakian, Z. Tropical arithmetic and matrix algebra. *Communications in Algebra® 37*, 4 (2009), 1445–1468.

[33] Kaburlasos, V. G., Athanasiadis, I. N., and Mitkas, P. A. Fuzzy lattice reasoning (flr) classifier and its application for ambient ozone estimation. *International Journal of Approximate Reasoning 45*, 1 (2007), 152–188.

[34] Lange, K., Hunter, D. R., and Yang, I. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics 9*, 1 (2000), 1–20.

[35] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[36] LeCun, Y., Cortes, C., and Burges, C. J. The MNIST database of handwritten digits, 1998.

[37] Lipp, T., and Boyd, S. Variations and extension of the convex–concave procedure. *Optimization and Engineering 17*, 2 (2016), 263–287.

[38] Litvinov, G. L. Maslov dequantization, idempotent and tropical mathematics: A brief introduction. *Journal of Mathematical Sciences 140*, 3 (2007), 426–444.

[39] Maclagan, D., and Sturmfels, B. *Introduction to tropical geometry*, vol. 161. American Mathematical Soc., 2015.

[40] Maragos, P. Morphological filtering for image enhancement and feature detection. In *The Image and Video Processing Handbook*, A. C. Bovik, Ed., 2 ed. Elsevier Acad. Press, 2005, pp. 135–156.

[41] Maragos, P. Dynamical systems on weighted lattices: General theory. *arXiv preprint arXiv:1606.07347* (2016).

[42] Maragos, P., and Koutras, P. Max-product dynamical systems and applications to audio-visual salient event detection in videos. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (2015), IEEE, pp. 2284–2288.

[43] McCulloch, W. S., and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*, 4 (1943), 115–133.

[44] Mobahi, H. Training recurrent neural networks by diffusion. *arXiv preprint arXiv:1601.04114* (2016).

[45] Mobahi, H., and Fisher III, J. W. On the link between gaussian homotopy continuation and convex envelopes. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (2015), Springer, pp. 43–56.

[46] Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems* (2014), pp. 2924–2932.

[47] Nair, V., and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010), pp. 807–814.

[48] Owen, A. B. *Monte Carlo theory, methods and examples.* 2013.

[49] Pachter, L., and Sturmfels, B. Tropical geometry of statistical models. *Proceedings of the National Academy of Sciences of the United States of America 101*, 46 (2004), 16132–16137.

[50] Pascanu, R., Dauphin, Y. N., Ganguli, S., and Bengio, Y. On the saddle point problem for non-convex optimization. *arXiv preprint arXiv:1405.4604* (2014).

[51] Petridis, V., and Kaburlasos, V. G. Fuzzy lattice neural network (flnn): a hybrid model for learning. *IEEE Transactions on Neural Networks 9*, 5 (1998), 877–890.

[52] Ripley, B. D. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[53] Ritter, G. X., and Urcid, G. Lattice algebra approach to single-neuron computation. *IEEE Transactions on Neural Networks 14*, 2 (2003), 282–295.

[54] Rivasplata, O. Subgaussian random variables: An expository note.

[55] Rockafellar, T. R., and Wetts, R. J. B. *Variational Analysis*, vol. 317 of *A Series of Comprehensive Studies in Mathematics*. Springer Science & Business Media, 1998.

[56] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*, 6 (1958), 386.

[57] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature 323* (1986), 533–536.

[58] Serra, J. *Image Analysis and Mathematical Morphology*, vol. 1. Academic press, 1982.

[59] Shen, X., Diamond, S., Gu, Y., and Boyd, S. Disciplined convex-concave programming. *arXiv preprint arXiv:1604.02639* (2016).

[60] Soille, P. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[61] Sussner, P. Morphological perceptron learning. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings* (1998), IEEE, pp. 477–482.

[62] Sussner, P., and Esmi, E. L. Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm. *Information Sciences 181*, 10 (2011), 1929–1950.

[63] Sussner, P., and Valle, M. E. Gray-scale morphological associative memories. *IEEE Transactions on Neural Networks 17*, 3 (2006), 559–570.

[64] Vanderbei, R. J., et al. *Linear programming*. Springer, 2015.

[65] Vese, L. A method to convexify functions via curve evolution. *Communications in partial differential equations 24*, 9-10 (1999), 1573–1591.

[66] Wainwright, M. *High Dimensional Statistics - A Non-Asymptotic Viewpoint*, draft ed. 2015.

[67] Wang, S. General constructive representations for continuous piecewise-linear functions. *IEEE Transactions on Circuits and Systems I: Regular Papers 51*, 9 (2004), 1889–1896.

[68] Xu, L., Crammer, K., and Schuurmans, D. Robust support vector machine training via convex outlier ablation. In *AAAI* (2006), vol. 6, pp. 536–542.

[69] Yuille, A. L., and Rangarajan, A. The concave-convex procedure. *Neural computation 15*, 4 (2003), 915–936.

[70] Ziegler, G. M. *Lectures on polytopes*, vol. 152. Springer Science & Business Media, 1995.