Τεχνικές βελτιστοποίησης με εφαρμογές στη Μηχανική Μάθηση

Κωνσταντίνα-Μαρία Αργυροπούλου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
Τμήμα Μαθηματικών και Εφαρμοσμένων Μαθηματικών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
Τμήμα Μαθηματικών και Εφαρμοσμένων Μαθηματικών

Τεχνικές βελτιστοποίησης με εφαρμογές στη Μηχανική Μάθηση

Κωνσταντίνα-Μαρία Αργυροπούλου

Διπλωματική εργασία υποβληθείσα προς μερική εκπλήρωση
των απαραίτητων προϋποθέσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης
στα Εφαρμοσμένα Μαθηματικά

20 Οκτωβρίου 2022

Επιβλέπων:

Μιχάλης Πλεξουσάκης

Μέλη επιτροπής:

Θεόδωρος Κατσαούνης

Παναγιώτης Χατζηπαντελίδης

Ημερομηνία εξέτασης: 20 Οκτωβρίου 2022

Ευχαριστίες

Ένα μεγάλο ευχαριστώ στον κύριο Μιχάλη Πλεξουσάκη για τις συμβουλές, την καθοδήγηση και την βοήθεια που μου προσέφερε από την έναρξη έως την περάτωση της εργασίας!

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Περίληψη

Πολλά προβλήματα στην επιστήμη και την τεχνολογία οδηγούν στη διατύπωση και επίλυση προβλημάτων βελτιστοποίησης σε διακριτές ή συνεχείς μεταβλητές. Λόγω της αυξημένης χρήσης τεχνικών μηχανικής μάθησης για την επίλυση προβλημάτων στην τεχνολογία και για τη λύση αποφάσεων, η αποτελεσματική επίλυση προβλημάτων βελτιστοποίησης έχει αποκτήσει ένα νέο θεωρητικό και πρακτικό ενδιαφέρον. Στην εργασία αυτή γίνεται μια ανασκόπηση των θεμελιωδών αλγορίθμων βελτιστοποίησης και της επίλυσης του προβλήματος βελτιστοποίησης που ανακύπτει κατά την εκπαίδευση νευρωνικών δικτύων.

Abstract

Many problems in science and technology lead to the formulation and solution of optimization problems in discrete or continuous variables. Due to the increased use of machine learning techniques to solve problems in technology and for the decision making problem, the effective optimization problem solving has gained a new theoretical and practical interest. This work reviews the fundamental optimization algorithms and the solution of the optimization problem that arises during the training of neural networks.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Λόγω της ευρείας (και αυξανόμενης) χρήσης αλγορίθμων βελτιστοποίησης στην επιστήμη, τη μηχανική, την οικονομία και τη βιομηχανία το αντικείμενο αυτό έχει βρει ανανεωμένο ενδιαφέρον. Η παρούσα εργασία έχει σκοπό να παρουσιάσει μια περιγραφή των πιο ισχυρών τεχνικών για την επίλυση των προβλημάτων βελτιστοποίησης. Στην πράξη, η βελτιστοποίηση εξαρτάται όχι μόνο από αποτελεσματικούς και ισχυρούς αλγόριθμους, αλλά και από καλές τεχνικές μοντελοποίησης και προσεκτική ερμηνεία των αποτελεσμάτων. Στην εργασία αυτή προσπαθούμε επίσης να αναδείξουμε αυτές τις πτυχές, δηλαδή, τη μοντελοποίηση, συνθήκες βελτιστοποίησης (optimality conditions), την υλοποίηση αλγορίθμων και την ερμηνεία των αποτελεσμάτων.

Τα υπόλοιπα κεφάλαια της εργασίας είναι διαρθρωμένα ως εξής: στο κεφάλαιο 2 παρουσιάζεται η μέθοδος της καθόδου της μέγιστης κλίσης και παραλλαγές της, οι ο οποίες σε συνδυασμό με τη μέθοδο του Newton συνθέτουν ένα πλαίσιο μεθόδων που συγκλίνουν ολικά. Με τον όρο *ολική σύγκλιση* εννοούμε, βέβαια, τη σύγκλιση μιας μεθόδου σε ένα τοπικό ελάχιστο από σχεδόν κάθε αρχική προσεγγιστική τιμή.

Το κεφάλαιο 3 παρουσιάζει τη μέθοδο των Nelder–Mead, ως αντιπρόσωπο των μεθόδων βελτιστοποίησης που απαιτούν μόνο τη διαθεσιμότητα τιμών της αντικειμενικής συνάρτησης. Τροφοδοτούμενη από έναν αυξανόμενο αριθμό εφαρμογών στην επιστήμη και τη μηχανική, η ανάπτυξη αλγορίθμων βελτιστοποίησης χωρίς παραγώγους έχει μελετηθεί διεξοδικά και έχει βρει ανανεωμένο ενδιαφέρον τον τελευταίο καιρό. Ο ενδιαφερόμενος αναγνώστης παραπέμπεται στην αναφορά [11] για περισσότερες λεπτομέρειες.

Το κεφάλαιο 4 αποτελεί μια μικρή εισαγωγή στα τεχνητά νευρωνικά δίκτυα, αλγόριθμοι εκμάθησης εμπνευσμένοι από την δομή και την λειτουργία των βιολογικών νευρωνικών δικτύων, με χρήση του παραδείγματος των Higham και Higham [2]. Η βελτιστοποίηση είναι ένα από τα κρίσιμα στοιχεία αλγορίθμων μηχανικής μάθησης. Οι αλγόριθμοι μηχανικής μάθησης δημιουργούν ένα μοντέλο βελτιστοποίησης το οποίο επιχειρεί να "μάθει" τις παραμέτρους της αντικειμενικής συνάρτησης. Με την έκρηξη του όγκου των δεδομένων, η αποδοτικότητα και αποτελεσματικότητα των αριθμητικών αλγορίθμων βελτιστοποίησης επηρεάζει καίρια την εφαρμογή αυτών των αλγορίθμων.

Το λογισμικό που αναπτύχθηκε για τις ανάγκες αυτής της εργασίας παρουσιάζεται στο Κεφάλαιο 5. Περιλαμβάνει υλοποιήσεις, σε Python, της μεθόδου της καθόδου μέγιστης κλίσης, της μεθόδου της καθόδου μέγιστης κλίσης με γραμμική αναζήτηση και επιλογής του βήματος με backtracking, της μεθόδου του Broyden για τη λύση μη γραμμικών εξισώσεων και του προβλήματος της βελτιστοποίησης, και της μεθόδου των Nelder–Mead.

Τέλος, στο Παράρτημα Α παρουσιάζεται η έννοια της παραγωγισιμότητας κατά Fréchet διανυσματικών συναρτήσεων $F : \mathbb{R}^n \to \mathbb{R}^n$ και διάφορα τύπου θεωρήματα "μέσης τιμής".

ΚΕΦΑΛΑΙΟ 2

ΜΕΘΟΔΟΙ ΚΑΘΟΔΟΥ ΠΡΩΤΗΣ ΤΑΞΕΩΣ

2.1 Μέθοδος Απότομης Καθόδου (Steepest Descent Method)

Η μέθοδος απότομης καθόδου είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης πρώτης τάξεως που χρησιμοποιείται για την εύρεση ενός τοπικού ελαχίστου μιας διαφορίσιμης συνάρτησης. Δοθείσας μιας διαφορίσιμης συνάρτησης $f : \mathbb{R}^n \to \mathbb{R}$, η κατεύθυνση της *μέγιστης καθόδου* (steepest descent) είναι η $-\nabla f(x_0)$, όπου το x_0 είναι το σημείο εκκίνησης.

Πράγματι, ορίζοντας τη συνάρτηση $\varphi(t) = f(x_0+tu)$, όπου $u \in \mathbb{R}^n$ με $||u|| = 1$, και εφαρμόζοντας τον κανόνα της αλυσίδας έχουμε:

$$\begin{aligned}\varphi'(t) &= \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \cdots + \frac{\partial f}{\partial x_n}\frac{\partial x_n}{\partial t} \\ &= \frac{\partial f}{\partial x_1}u_1 + \cdots + \frac{\partial f}{\partial x_n}u_n \\ &= \nabla f(x_0 + tu) \cdot u.\end{aligned}$$

Θέτοντας $t = 0$ παίρνουμε

$$\varphi'(0) = \nabla f(x_0) \cdot u = ||\nabla f(x_0)|| \cos(\theta),$$

όπου θ είναι η γωνία μεταξύ των διανυσμάτων $\nabla f(x_0)$ και u. Από τη σχέση αυτή έπεται ότι η $\varphi'(0)$ ελαχιστοποιείται όταν $\cos(\theta) = -1$, δηλαδή όταν $\theta = \pi$. Συνεπώς,

$$u = -\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}, \qquad \varphi'(0) = -\|\nabla f(x_0)\|.$$

Επομένως, έχουμε αναγάγει το πρόβλημα ελαχιστοποίησης της συνάρτησης πολλών μεταβλητών f στο πρόβλημα ελαχιστοποίησης μιας συνάρτησης μιας μεταβλητής φ, για την επιλογή u που αναφέρουμε παραπάνω. Ψάχνουμε πλέον την τιμή της μεταβλητής t, για $t > 0$, που ελαχιστοποιεί την

$$\varphi_0(t) = f(x_0 - t\nabla f(x_0)).$$

Αν η φ ελαχιστοποιείται στο σημείο t_0, υπολογίζουμε το επόμενο σημείο της μεθόδου απότομης κλίσης ως

$$x_1 = x_0 - t_0\nabla f(x_0).$$

Συνεχίζουμε την διαδικασία αναζητώντας από το x_1 στην κατεύθυνση της $-\nabla f(x_1)$, το σημείο το x_2 που ελαχιστοποιεί την $\varphi_1(t) = f(x_1 - t\nabla f(x_1))$.

Συμπερασματικά, η μέθοδος απότομης καθόδου ξεκινώντας από ένα αρχικό σημείο x_0, υπολογίζει μια ακολουθία επαναλήψεων (x_k), όπου για $k \geq 0$

$$x_{k+1} = x_k - t_k \nabla f(x_k),$$

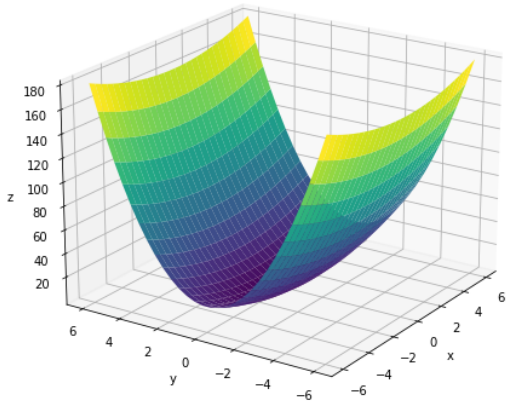
με το $t_k > 0$ τέτοιο ώστε να ελαχιστοποιεί την συνάρτηση

$$\varphi_k(t) = f(x_k - t_k \nabla f(x_k)). \tag{2.1}$$

Παράδειγμα. Θα εφαρμόσουμε τη μέθοδο απότομης καθόδου στη συνάρτηση

$$f(x, y) = \frac{1}{2}x^2 + \frac{9}{2}y^2,$$

η οποία προφανώς έχει ελάχιστο στο σημείο $(0, 0)$, δείτε το Σχήμα 2.1 Επιλέγουμε το αρχικό σημείο εκκίνησης της μεθόδου απότομης κλίσης ως το $x_0 = (9, 1)$.



Σχήμα 2.1: Η γραφική παράσταση της συνάρτησης $f(x) = \frac{1}{2}x^2 + \frac{9}{2}y^2$.

Επειδή $\nabla f(x, y) = (x, 9y)$, υπολογίζουμε πρώτα την κατεύθυνση της απότομης καθόδου

$$\nabla f(x_0) = (9, 9),$$

και ελαχιστοποιούμε τη συνάρτηση

$$\varphi(t) = f((x, y) - t(x, 9y)) = f(x(1 - t), y(1 - 9t)),$$

δηλαδή την

$$\varphi(t) = \frac{1}{2}x^2(1 - t)^2 + \frac{9}{2}y^2(1 - 9t)^2.$$

Στην συνέχεια, βρίσκουμε τα κρίσιμα σημεία της φ

$$\varphi'(t) = 0 \Rightarrow -x^2(1 - t) - 81y^2(1 - 9t) = 0,$$

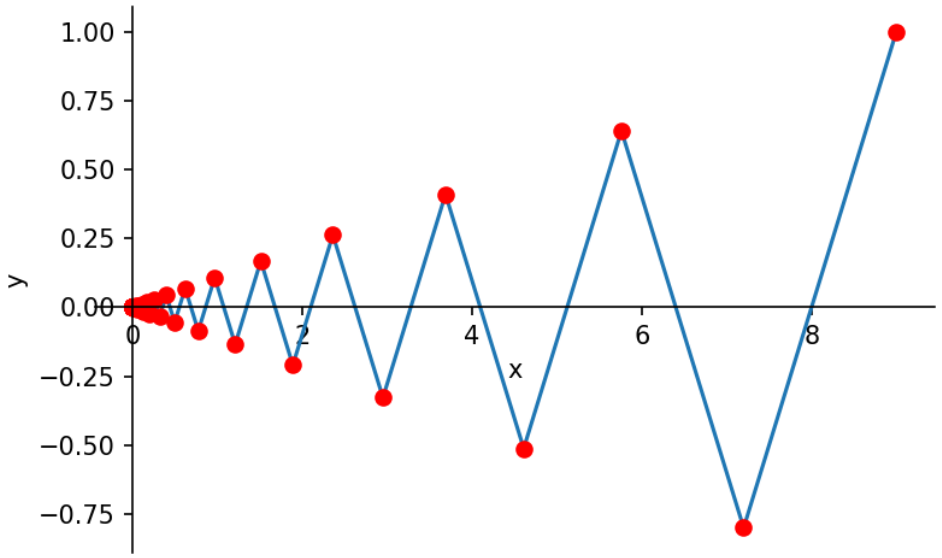
όπου λύνοντας ως προς t προκύπτει ότι

$$t = \frac{x^2 + 82y^2}{x^2 + 729y^2}. \tag{2.2}$$

Το επόμενο σημείο της ακολουθίας (x_k) προκύπτει, όπως έχουμε ήδη δει, από τη σχέση

$$x_{k+1} = x_k - t_k \nabla f(x_k). \tag{2.3}$$

Στο Σχήμα 2.2 απεικονίζεται η ακολουθία επαναλήψεων (x_k) που παράγει η μέθοδος για την συνάρτηση $f(x, y) = \frac{1}{2}x^2 + \frac{9}{2}y^2$, η οποία φαίνεται να συγκλίνει στο ελάχιστο της $f(x, y)$, δηλαδή το $x^* = (0, 0)$.



Σχήμα 2.2: Μέθοδος Απότομης Καθόδου για τη συνάρτηση $f(x) = \frac{1}{2}x^2 + \frac{9}{2}y^2$.

2.2 Γραμμική Αναζήτηση με Οπισθοχώρηση - Backtracking Line Search

Συχνά είναι μόνο δυνατό, και υπολογιστικά πιο αποδοτικό, να χρησιμοποιούμε *προσεγγίσεις* του ελαχίστου της t_k της συνάρτησης (2.1) από το να το υπολογίζουμε ακριβώς σε κάθε επανάληψη, ειδικά όταν οι υπολογισμοί της συνάρτησης που ελαχιστοποιούμε και της πρώτης παραγώγου της είναι υπολογιστικά ακριβοί.
Έτσι, έχουν δημιουργηθεί διάφορες μέθοδοι που χρησιμοποιούν σε κάθε επανάληψη μια *προσέγγιση* του ελαχίστου t_k. Μια από αυτές τις μεθόδους είναι η μέθοδος της *γραμμικής αναζήτησης με οπισθοδρόμηση* (backtracking line search). Δοθείσης μιας συνάρτησης $f : \mathbb{R}^n \to \mathbb{R}$ η οποία είναι διαφορίσιμη, γνωρίζουμε ότι η κατεύθυνση της απότομης καθόδου στο k βήμα του αλγορίθμου, είναι η $p_k = -\nabla f(x_k)$, όπου το x_k είναι η k-στη προσέγγιση μιας επαναληπτικής διαδικασίας ελαχιστοποίησης, όπως η μέθοδος της απότομης καθόδου. Θα λέμε ότι το βήμα t_k στη σχέση (2.1) είναι *αποδεκτό* αν ισχύουν τα παρακάτω:

- για κάποια σταθερά $\alpha \in (0, 1)$

$$f(x_k + t \cdot p_k) \leq f(x_k) + \alpha t \nabla f(x_k)^T \cdot p_k, \tag{2.4}$$

- για κάποια σταθερά $\beta \in (\alpha, 1)$

$$\beta \cdot \nabla f(x_k)^T \leq \nabla f(x_k + t p_k)^T \cdot p_k. \tag{2.5}$$

Οι παραπάνω σχέσεις είναι γνωστές ως συνθήκες Wolfe. Επιλέγουμε αρχικά $t = 1$ και στη συνέχεια, όσο δεν ικανοποιείται η σχέση (2.4) οπισθοχωρούμε, δηλαδή, μειώνουμε το βήμα t κατά μια σταθερά $\rho \in (0, 1)$. Κατόπιν, και εφόσον η συνθήκη (2.4) ικανοποιείται, υπολογίζουμε τον επόμενο όρο της ακολουθίας προσεγγίσεων x_{k+1} από τη σχέση

$$x_{k+1} = x_k + tp_k.$$

Στην πράξη, επιλέγουμε $\rho \in [\frac{1}{10}, 1)$. Έστω

$$\varphi(t) = f(x_k + tp_k), \quad t > 0. \tag{2.6}$$

Στην πρώτη οπισθοδρόμηση κατασκευάζουμε μια προσέγγιση της φ χρησιμοποιώντας ένα πολυώνυμο $q \in \mathbb{P}_2$ τέτοιο ώστε

$$q(0) = \varphi(0), \quad q(1) = \varphi(1), \quad q'(0) = \varphi'(0).$$

Αν το $f(x_k + p_k)$ δεν είναι αποδεκτό (δηλαδή δεν ικανοποιείται η πρώτη συνθήκη Wolfe), τότε

$$\varphi(1) > \varphi(0) + \alpha\varphi'(0).$$

Εύκολα βλέπουμε ότι

$$q(t) = [\varphi(1) - \varphi(0) - \varphi'(0)]t^2 + \varphi'(0) \cdot t + \varphi(0),$$

και η προηγούμενη σχέση δείχνει ότι το πολυώνυμο q είναι όντως δευτέρου βαθμού. Μηδενίζοντας την πρώτη παράγωγο καταλήγουμε στην επιλογή του t

$$t^* = \frac{\varphi'(0)}{2[\varphi(1) - \varphi(0) - \varphi'(0)]}. \tag{2.7}$$

Το γεγονός ότι $q''(t^*) = \varphi(1) - \varphi(0) - \varphi'(0) > 0$, δείχνει ότι το q έχει όντως ελάχιστο στο t^*.

Παρατήρηση. Στην περίπτωση που χρειαστεί να οπισθοχωρήσουμε δυο φορές, δηλαδή η πρώτη συνθήκη Wolfe εξακολουθεί να μην ισχύει με την επιλογή του βήματος t^* από την σχέση (2.7), τότε προσεγγίζουμε τη συνάρτηση (2.1) σε μια γειτονιά του x_k από ένα πολυώνυμο $p \in \mathbb{P}_3$ τέτοιο ώστε

$$p(0) = \varphi(0), \quad p'(0) = \varphi'(0) = \nabla f(x_k) \cdot p_k, \quad p(t_1) = \varphi(t_1), \quad p(t_2) = \varphi(t_2).$$

Όπως και πριν, εύκολα βλέπουμε ότι οι παραπάνω σχέσεις δίνουν

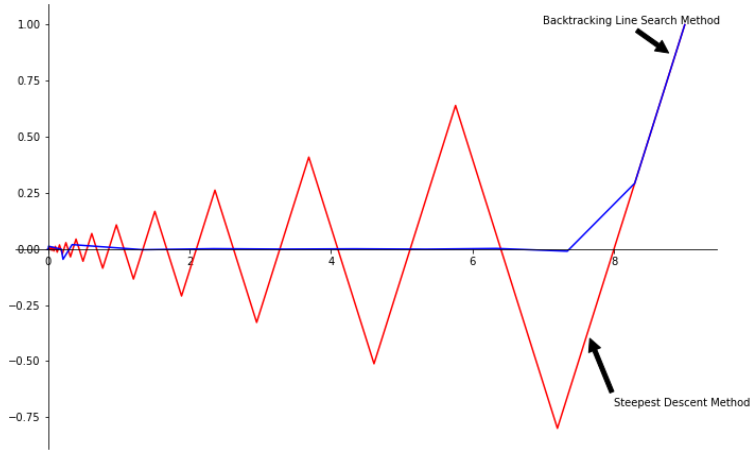
$$p(\lambda) = \alpha t^3 + \beta t^2 + \varphi'(0)t + \varphi(0),$$

όπου οι σταθερές α, β ορίζονται από τις σχέσεις

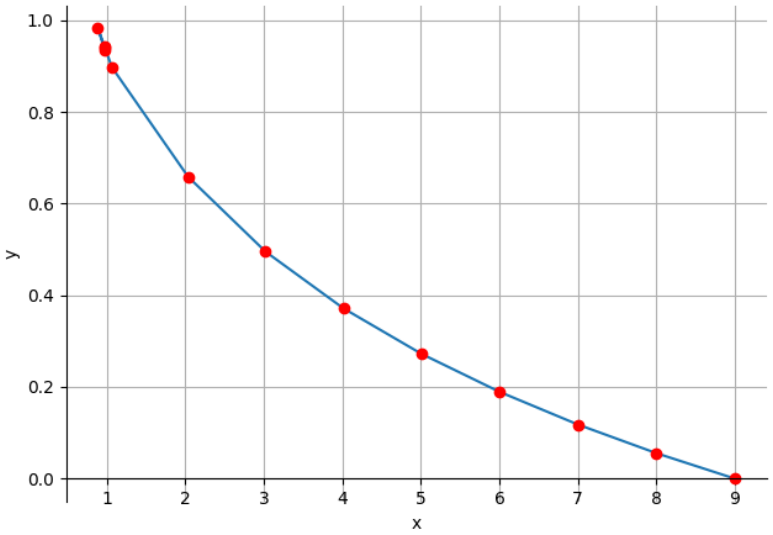
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{1}{t_1 - t_2} \cdot \begin{bmatrix} 1/t_1^2 & -1/t_2^2 \\ -t_2/\lambda_1^2 & t_1/t_2^2 \end{bmatrix} \cdot \begin{bmatrix} \varphi(t_1) - \varphi(0) - \varphi'(0)t_1 \\ \varphi(t_2) - \varphi(0) - \varphi'(0)t_2 \end{bmatrix}$$

Παρατήρηση. Η μέθοδος της γραμμικής αναζήτησης με οπισθοδρόμηση, όπως παρουσιάστηκε εδώ, δεν χρησιμοποιεί τη δεύτερη συνθήκη Wolfe (2.5). Είναι προφανές ότι θα μπορούσαμε να βασίσουμε την επιλογή του βήματος t απαιτώντας την ικανοποίηση και των *δύο* συνθηκών Wolfe κατά την οπισθοδρόμηση.

Παράδειγμα. Εφαρμόζουμε τη μέθοδο Γραμμικής Αναζήτησης με Οπισθοδρόμηση στη συνάρτηση $f(x, y) = \frac{1}{2}x^2 + \frac{9}{2}y^2$ με αρχικό σημείο $x_0 = (9, 1)$ και συγκρίνουμε με την μέθοδο Απότομης Καθόδου. Οι ακολουθίες προσεγγίσεων που παράγουν οι δύο μέθοδοι φαίνονται στο Σχήμα 2.3. Είναι φανερό ότι μέθοδος της γραμμικής αναζήτησης με οπισθοδρόμηση συγκλίνει προς το ελάχιστο $(0, 0)$ γρηγορότερα.



Σχήμα 2.3: Σύγκριση των μεθόδων γραμμικής αναζήτησης (μπλε) και απότομης καθόδου (κόκκινο) για την ελαχιστοποίηση της $f(x, y) = \frac{1}{2}x^2 + \frac{9}{2}y^2$.



Σχήμα 2.4: Ακολουθία προσεγγίσεων της μεθόδου της γραμμικής αναζήτησης για την συνάρτηση Rosenbrock για $n = 2$

Παράδειγμα. Εφαρμόζοντας τη μέθοδο της γραμμικής αναζήτησης με οπισθοδρόμηση στη συνάρτηση Rosenbrock με $n = 2$, δηλαδή την

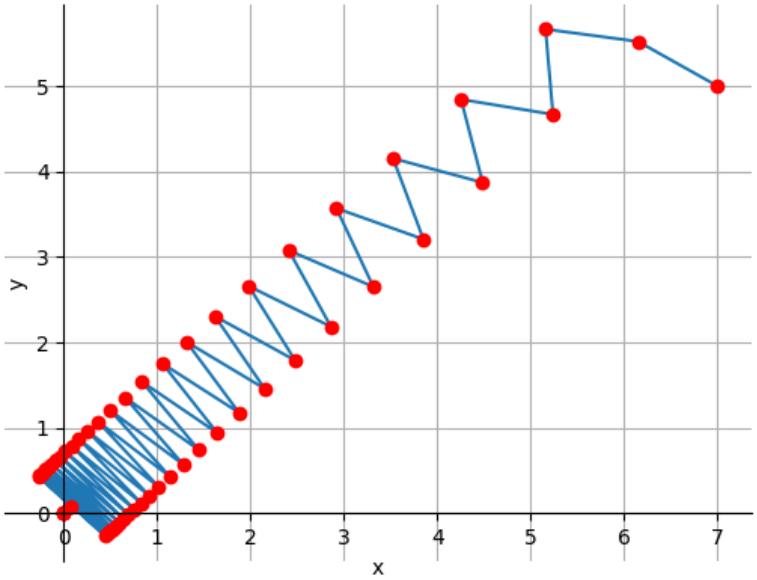
$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2,$$

με αρχικό σημείο $x_0 = (9, 0)$, λαμβάνουμε την ακολουθία προσεγγίσεων του ελαχίστου $(1, 1)$ που φαίνεται στο Σχήμα 2.4. Συγκεκριμένα, βλέπουμε ότι ξεκινώντας από το $x_0 = (9, 0)$, η πρώτη επανάληψη δίνει $x_1 = (8.00153984, 0.05547343)$. Ύστερα από 12 επαναλήψεις καταλήγουμε στο σημείο $x_{13} = (0.96983032, 0.94031536)$, το οποίο είναι αρκετά κοντά στο ολικό ελάχιστο $x^* = (1, 1)$. Η συμπεριφορά της μεθόδου για διαφορετικά σημεία εκκίνησης είναι εντελώς ανάλογη.

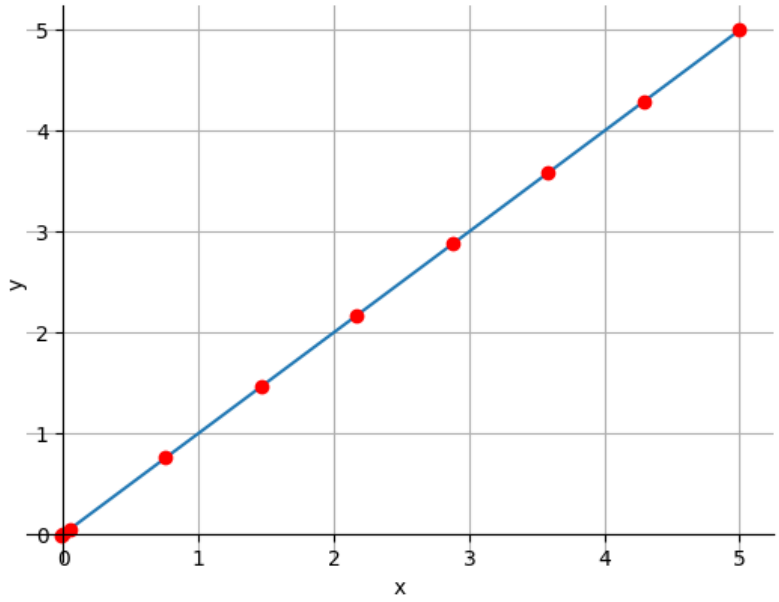
Παράδειγμα. Θεωρούμε τώρα τη συνάρτηση Matyas

$$f(x) = 0.26(x^2 + y^2) - 0.48xy,$$

με αρχικό σημείο για τη μέθοδο της γραμμικής αναζήτησης το $x_0 = (7, 5)$. Η ακολουθία προσεγγίσεων που παράγει η μέθοδος της γραμμικής αναζήτησης με οπισθοδρόμηση φαίνεται στο Σχήμα 2.5, ενώ αν επιλέξουμε ως αρχικό σημείο το $x_0 = (5, 5)$ η μέθοδος παράγει την ακολουθία που απεικονίζεται στο Σχήμα 2.6.



Σχήμα 2.5: Μέθοδος της γραμμικής αναζήτησης με οπισθοδρόμηση για τη συνάρτηση Matyas με αρχικό σημείο $x_0 = (7, 5)$.



Σχήμα 2.6: Μέθοδος της γραμμικής αναζήτησης με οπισθοδρόμηση για την συνάρτηση Matyas με αρχικό σημείο $x_0 = (5, 5)$.

2.3 Η μέθοδος του Broyden

Δοθείσας μιας διανυσματικής συνάρτησης $F : \mathbb{R}^n \to \mathbb{R}^n$, αναζητούμε $x^* \in \mathbb{R}^n$ τέτοιο ώστε $F(x^*) = 0$. Όπως είναι γνωστό, το βήμα της μεθόδου του Newton, δεδομένης κάποιας αρχικής προσέγγισης x_c είναι

$$x_+ = x_c - J(x_c)^{-1} F(x_c),$$

όπου $J(x_c)$ είναι ο Ιακωβιανός πίνακας της F στο x_c (Λεπτομέρειες για την έννοια της παραγώγου διανυσματικών συναρτήσεων και διάφορα θεωρήματα τύπου "μέσης τιμής" για διανυσματικές συναρτήσεις περιέχονται στο Παράρτημα Α). Η μέθοδος του Newton συγκλίνει γρήγορα (τετραγωνικά) αν η προσέγγιση x_c βρεθεί αρκετά κοντά στη ρίζα x^*, αλλά δεν συγκλίνει απαραίτητα από οποιαδήποτε αρχική τιμή.

Αν η προσέγγιση x_c δεν είναι κοντά στη ρίζα, ένας τρόπος να αποδεχτούμε ή να απορρίψουμε το βήμα x_+ της μεθόδου του Newton θα μπορούσε να είναι ο έλεγχος κατά πόσο $\|F(x_+)\| < \|F(x_c)\|$, όπου $\|\cdot\|$ κάποια νόρμα στον $\mathbb{R}^n$, για παράδειγμα, η Ευκλείδεια νόρμα. Θεωρούμε λοιπόν το πρόβλημα ελαχιστοποίησης

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \frac{1}{2} F(x)^T F(x), \tag{2.8}$$

το οποίο δικαιολογείται από το γεγονός ότι $\|F(x)\|_2^2 = F(x)^T F(x)$. Ο όρος $1/2$ υπάρχει για την ευκολία των υπολογισμών. Προφανώς, κάθε λύση του προβλήματος $F(x) = 0$ είναι λύση του προβλήματος (2.8) αλλά μπορεί να υπάρχουν τοπικά ελάχιστα του προβλήματος (2.8) τα οποία *δεν* ικανοποιούν το πρόβλημα $F(x) = 0$.

Στη μία διάσταση θεωρούμε το γραμμικό μοντέλο

$$M_+(x) = f(x_+) + a_+(x - x_+), \quad a_+ \in \mathbb{R},$$

για την προσέγγιση της συνάρτησης $f(x)$ σε μια γειτονιά του σημείου x_+. Το μοντέλο αυτό ικανοποιεί $M_+(x_+) = f(x_+)$, και δίνει τη μέθοδο του Newton, αν επιλέξουμε $a_+ = f'(x_+)$. Αν η τιμή $f'(x_+)$

δεν είναι διαθέσιμη , η απαίτηση $M_+(x_c) = f(x_c)$, δηλαδή,

$$f(x_c) = f(x_+) + a_+(x_c - x_+),$$

δίνει τη *μέθοδο της τέμνουσας*, δηλαδή, την προσέγγιση της $f'(x_+)$ από το πηλίκο

$$a_+ = \frac{f(x_+) - f(x_c)}{x_+ - x_c}.$$

Σε περισσότερες διαστάσεις το ανάλογο γραμμικό μοντέλο είναι το

$$M_+(x) = F(x_+) + A_+(x - x_+), \quad x, x_+ \in \mathbb{R}^n, \quad A_+ \in \mathbb{R}^{n\times n},$$

το οποίο επίσης ικανοποιεί $M_+(x_+) = F(x_+)$ και, όπως προηγουμένως, η επιλογή $A_+ = J(x_+)$ δίνει τη μέθοδο του Newton. Αν η Ιακωβιανή $J(x_+)$ δεν είναι διαθέσιμη, η απαίτηση $M_+(x_c) = F(x_c)$ δίνει την προσέγγιση

$$F(x_c) = F(x_+) + A_+(x_c - x_+) \Rightarrow A_+(x_+ - x_c) = F(x_+) - F(x_c).$$

Θα γράφουμε $s_c = x_+ - x_c$ για το βήμα της μεθόδου, και $y_c = F(x_+) - F(x_c)$, για την απόδοση του τρέχοντος βήματος, έτσι ώστε

$$A_+ s_c = y_c. \tag{2.9}$$

Παρατηρούμε πως ο πίνακας A_+ έχει n^2 αγνώστους και το διάνυσμα s_c ανήκει στο $\mathbb{R}^n$. Επομένως, δεν μπορούμε να υπολογίσουμε τον πίνακα A_+ μόνο από τη σχέση (2.9).

Η ιδέα που εισήγαγε ο **Broyden** το 1965 για την επιλογή του πίνακα A_+ είναι η εξής: θα επιλέξουμε τον A_+ έτσι ώστε να ελαχιστοποιήσουμε τη διαφορά των μοντέλων $M_+(x)$ και $M_c(x)$ διατηρώντας παράλληλα την ισχύ της (2.9). Παρατηρούμε ότι για $x \in \mathbb{R}^n$ έχουμε, χρησιμοποιώντας την (2.9), ότι

$$\begin{aligned} M_+(x) - M_c(x) &= F(x_+) + A_+(x - x_+) - F(x_c) - A_c(x - x_c) \\ &= F(x_+) - F(x_c) - A_+(x_+ - x_c) + (A_+ - A_c)(x - x_c) \\ &= (A_+ - A_c)(x - x_c). \end{aligned}$$

Αν γράψουμε $x - x_c = \alpha s_c + t = \alpha(x_+ - x_c) + t$, για κάποιο $\alpha \in \mathbb{R}$ και $t \in \mathbb{R}^n$, με $t^T s_c = 0$, έχουμε

$$M_+(x) - M_c(x) = \alpha(A_+ - A_c)s_c + (A_+ - A_c)t.$$

Για τον πρώτο όρο στο δεξί μέλος της παραπάνω σχέσης έχουμε

$$(A_+ - A_c)s_c = A_+ s_c - A_c s_c = y_c - A_c s_c,$$

αλλά μπορούμε να μηδενίσουμε τον δεύτερο όρο επιλέγοντας τον πίνακα A_+ έτσι ώστε $(A_+ - A_c)t = 0$, για κάθε διάνυσμα t κάθετο στο βήμα s_c. Προς αυτό τον σκοπό, επιλέγουμε $A_+ - A_c$ ως τον τάξης ένα πίνακα us_c^T, για κάποιο $u \in \mathbb{R}^n$. Αφού πρέπει, επιπλέον, $A_+ s_c = y_c$, προκύπτει ότι

$$u = \frac{y_c - A_c s_c}{s_c^T s_c},$$

και, βέβαια,

$$A_+ = A_c + \frac{(y_c - A_c s_c)s_c^T}{s_c^T s_c}$$

Λήμμα 2.1. *Έστω $A \in \mathbb{R}^{n\times n}$ και $s, y \in \mathbb{R}^n$ με $s \neq 0$. Η λύση του προβλήματος*

$$\min_{\substack{B\in\mathbb{R}^{n\times n} \\ Bs=y}} \|B - A\|_2$$

είναι η

$$A_+ = A + \frac{(y - As)s^T}{s^T s}.$$

Απόδειξη. Έστω $B \in \mathbb{R}^{n\times n}$ τέτοιος ώστε $Bs = y$. Τότε

$$\|A_+ - A\|_2 = \left\| \frac{(y - As)\, s^T}{s^T s} \right\| = \left\| \frac{(B - A)\, ss^T}{ss^T} \right\|_2 \leq \|B - A\|_2 \left\| \frac{ss^T}{s^T s} \right\|_2 = \|B - A\|_2,$$

□

Παράδειγμα. Η εξίσωση $F(x) = 0$, όπου

$$F(x) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix}, \tag{2.10}$$

έχει ρίζες $(0, 3)^T$ και $(3, 0)^T$. Αν πάρουμε $x_0 = (1, 5)^T$ και εφαρμόσουμε τον παραπάνω αλγόριθμο με

$$A_0 = J(x_0) = \begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix},$$

παίρνουμε

$$F(x_0) = \begin{bmatrix} 3 \\ 17 \end{bmatrix}, \quad s_0 = -A_0 F(x_0) = \begin{bmatrix} -1.625 \\ -1.375 \end{bmatrix}.$$

Άρα, υπολογίζουμε

$$x_1 = x_0 + s_0 = \begin{bmatrix} -0.625 \\ 3.625 \end{bmatrix},$$

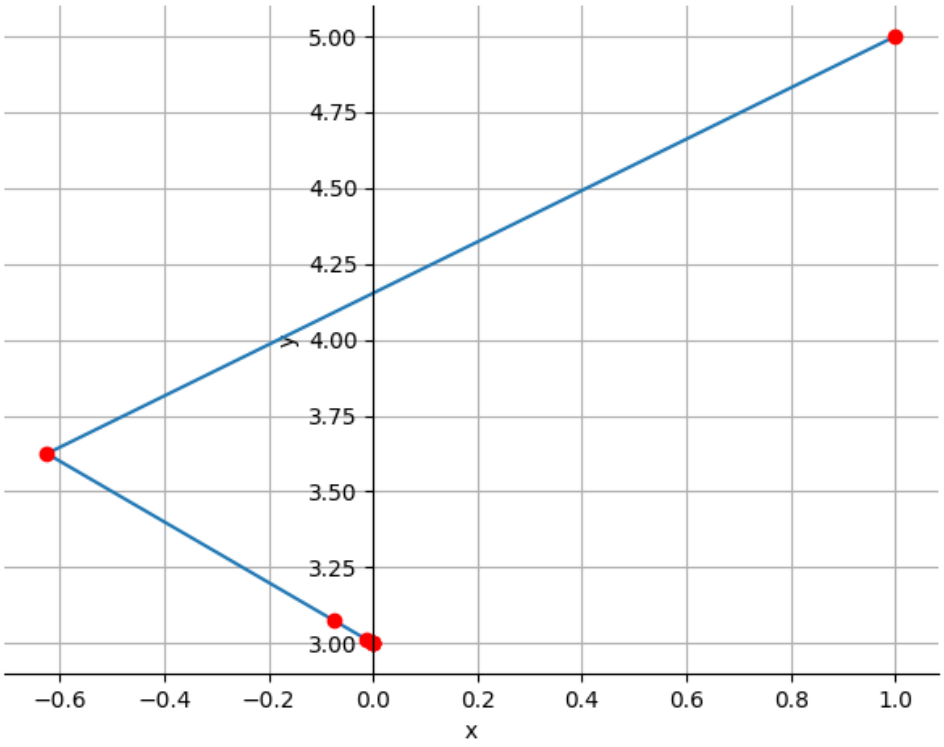
και

$$F(x_1) = \begin{bmatrix} 0 \\ 4.53125 \end{bmatrix}, \quad y_0 = \begin{bmatrix} -3 \\ -12.46875 \end{bmatrix}.$$

Επομένως, έχουμε

$$A_1 = A_0 + \begin{bmatrix} & \\ -1.625 & -1.375 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0.375 & 8.625 \end{bmatrix}.$$

Προκύπτει έτσι πως το επόμενο σημείο είναι το $x_2 = (-0.625, 3.625)$. Εύκολα κανείς επιβεβαιώνει πως η μέθοδος του Broyden συγκλίνει αρκετά γρήγορα, και μάλιστα καταλήγει στο σημείο $x_7 = (6.92880959 \times 10^{-17}, 3.00000000)$, το οποίο είναι πολύ κοντά στο σημείο (0,3), το οποίο είναι μία από τις δυο ρίζες της εξίσωσης $F(x) = 0$. Η ακολουθία προσεγγίσεων που παράγει η μέθοδος του Broyden φαίνεται στο Σχήμα 2.7.



Σχήμα 2.7: Η ακολουθία προσεγγίσεων της μεθόδου του Broyden για την εξίσωση (2.10) με $x_0 = (1, 5)$.

ΚΕΦΑΛΑΙΟ 3

ΜΕΘΟΔΟΙ ΚΑΘΟΔΟΥ ΔΕΥΤΕΡΗΣ ΤΑΞΕΩΣ

3.1 Μέθοδος Nelder-Mead

Η μέθοδος Nelder–Mead είναι μια αριθμητική μέθοδος που χρησιμοποιείται για την εύρεση του ελάχιστου (ή μέγιστου) μιας συνάρτησης σε έναν πολυδιάστατο χώρο. Είναι μια μέθοδος άμεσης αναζήτησης και εφαρμόζεται συχνά σε προβλήματα μη γραμμικής βελτιστοποίησης για τα οποία ενδεχομένως να μην είναι γνωστές οι παράγωγοι της αντικειμενικής συνάρτησης. Προτάθηκε από τους John Nelder και Roger Mead το 1965 και είναι αρκετά απλή στην κατανόηση. Η ιδέα της μεθόδου είναι ο υπολογισμός της αντικειμενικής συνάρτησης στις κορυφές ενός πολύτοπου (simplex) και η αντικατάσταση της κορυφής με την μεγαλύτερη τιμή από ένα άλλο σημείο, συνηθέστερα το συμμετρικό του ως προς την πλευρά με κορυφές τα σημεία όπου η συνάρτηση λαμβάνει τις μικρότερες τιμές. Υπενθυμίζουμε ότι ένα simplex είναι η γενίκευση της έννοιας του τριγώνου ή του τετράεδρου σε μεγαλύτερες διαστάσεις. Συγκεκριμένα, ένα n-simplex είναι η κυρτή θήκη $n+1$ σημείων στον $\mathbb{R}^n$. Η μέθοδος Nelder–Mead είναι πολύ δημοφιλής σε πολλούς τομείς της επιστήμης και της τεχνολογίας, ειδικά στη χημεία και την ιατρική.

Έστω συνάρτηση $f : \mathbb{R}^n \to \mathbb{R}$ και $p_0, p_1, ..., p_n$ τα (n+1) σημεία στον n-διάστατο χώρο, κορυφές ενός αρχικού simplex. Θα συμβολίζουμε με y_i για την τιμή της συνάρτησης f στο σημείο p_i. Ορίζουμε επίσης

$$p_h = \arg\max_i f(p_i), \quad p_l = \arg\min_i f(p_i), \qquad p_l \le p_m \le p_h, \quad m \ne l, h.$$

Έστω $\bar{p}$ το **κέντρο βάρους** των σημείων $p_i, i \ne h$. Ορίζουμε το σημείο

$$p^* = (1+\alpha)\bar{p} - \alpha p_h, \quad \alpha > 0$$

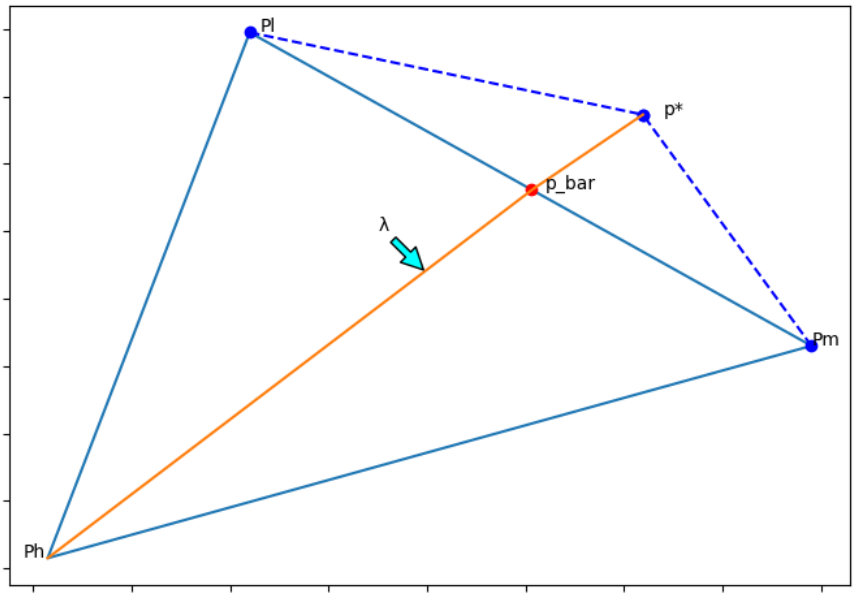
και θέτουμε $y^* = f(p^*)$, δείτε το Σχήμα 3.1. Το p^* είναι η ανάκλαση του p_h και βρίσκεται στην ευθεία που ενώνει το p_h και το $\bar{p}$, δηλαδή την

$$\lambda : p_h + t(\bar{p} - p_h), \quad t \in \mathbb{R}.$$

Τα βήματα της μεθόδου των Nelder–Mead είναι τα ακόλουθα:

- Εάν το y^* βρίσκεται μεταξύ των y_l και y_h τότε το p_h αντικαθίσταται από το p_* και επαναλαμβάνουμε την ίδια διαδικασία.
- Εάν $y^* < y_l$, τότε υπολογίζουμε το

$$p^{**} = \gamma p^* + (1-\gamma)\bar{p}, \quad \gamma > 1.$$



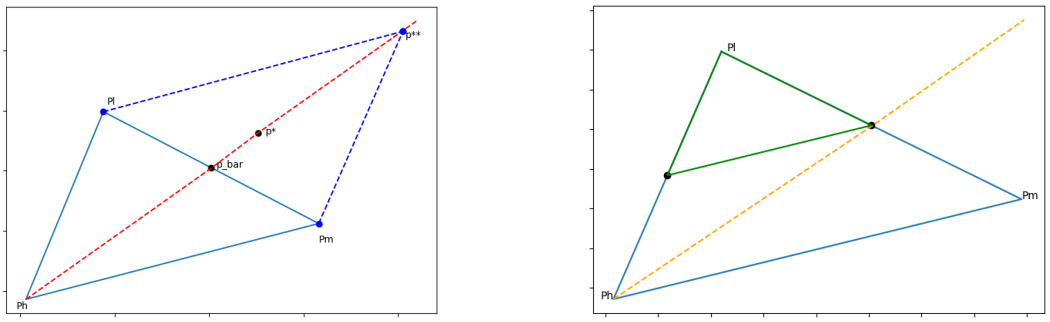
Σχήμα 3.1: Ανάκλαση του p_h ως προς την ευθεία λ με κάποιο συντελεστή $\alpha > 0$.

- Εάν $y^{**} < y_l$ αντικαθιστούμε το p_h απο το p^{**} και επαναλαμβάνουμε την ίδια διαδικασία.
- Διαφορετικά, αν $y^{**} > y_l$ αντικαθιστούμε το p_h απο το p^* και επαναλαμβάνουμε την ίδια διαδικασία.
- Εάν $y^* > y_i, \quad i \neq h$, τότε θέτουμε $p_h = \min(p_h, p^*)$ και υπολογίζουμε ξανά το

$$p^{**} = \beta p_h + (1-\beta)p^*, \quad \beta \in (0,1).$$

- Εάν $y^{**} > \min(y_h, y^*)$, τότε θέτουμε $p_i \leftarrow (p_i + p_l)/2$ και επαναλαμβάνουμε την ίδια διαδικασία.

Οι φάσεις της συστολής και διαστολής του simplex απεικονίζονται στο Σχήμα 3.2.

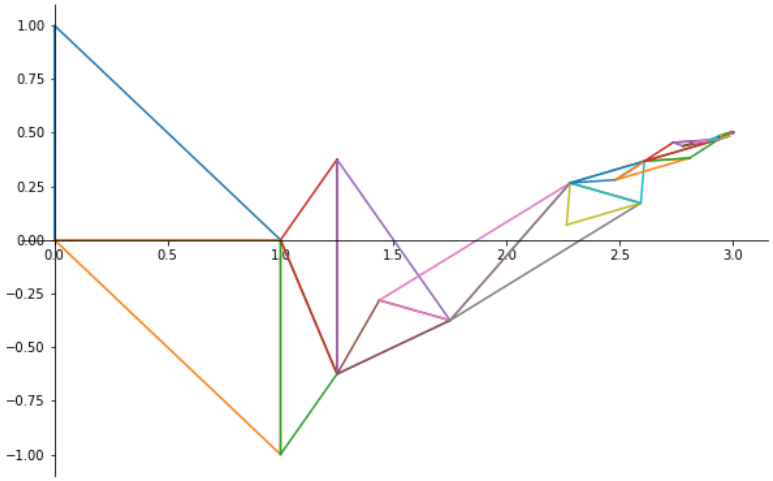


Σχήμα 3.2: Τα στάδια της διαστολής (αριστερά) και συστολής στη μέθοδο Nelder-Mead.

Παράδειγμα. Για την συνάρτηση Beale

$$f(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2,$$

η μέθοδος Nelder-Mead, ξεκινώντας από το σημείο $x_0 = (1, 4)$ καταλήγει στο σημείο με συντεταγμένες $x = (3.00002838, 0.50000758)$, πολύ κοντά, δηλαδή, στο ολικό ελάχιστο της συνάρτησης που λαμβάνεται στο $x^* = (3, 0.5)$. Το Σχήμα 3.3 απεικονίζει τους μετασχηματισμούς του αρχικού simplex με κορυφές $(0, 0)$, $(1, 0)$ και $(0, 1)$. Είναι φανερό ότι το στο πρώτο βήμα της μεθόδου το simplex ανακλάται ως προς τον οριζόντιο άξονα και στο δεύτερο συρρικνώνεται. Το βαρύκεντρο του τελικού simplex έχει συντεταγμένες $(3.0000000000000107, 0.5000000000000023)$.



Σχήμα 3.3: Οι μετασχηματισμοί του αρχικού simplex από τη μέθοδο Nelder-Mead για τη συνάρτηση Beale.

Παράδειγμα. Για τη συνάρτηση Rosenbrock

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2,$$

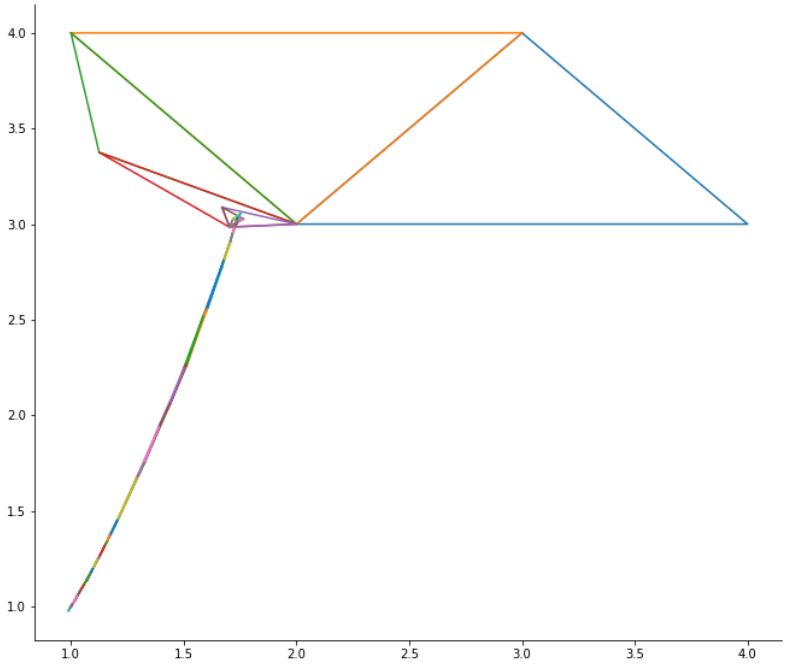
ξεκινώντας από το σημείο $x_0 = (9, 0)$ καταλήγουμε στο $x = (1.00001198, 1.0000268)$, το οποίο προσεγγίζει το ολικό ελάχιστο της συνάρτησης στο σημείο $x^* = (1, 1)$. Οι μετασχηματισμοί του αρχικού simplex με κορυφές $(3, 4)$, $(4, 3)$ και $(2, 3)$ απεικονίζονται στο Σχήμα 3.4. Το βαρύκεντρο του τελικού simplex έχει συντεταγμένες $(1.0000000000027787, 1.0000000000050666)$.

Ιδιαίτερο ενδιάφερον για τη μέθοδο Nelder-Mead παρουσιάζουν συναρτήσεις με πολλά τοπικά ελάχιστα, καθώς υπάρχει το ενδεχόμενο η μέθοδος να εγκλωβιστεί σε κάποιο από αυτά. Ένα τέτοιο παράδειγμα αποτελεί η συνάρτηση Ackley.

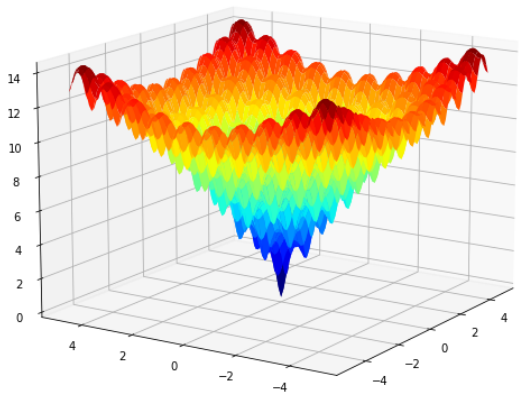
Παράδειγμα. Η συνάρτηση Ackley (στις δυο διαστάσεις) δίνεται από τη σχέση

$$f(x, y) = -20e^{0.2\sqrt{\frac{1}{2}(x^2+y^2)}} - e^{\frac{1}{2}(\cos(2\pi x)+(\cos 2\pi y))} + e + 20,$$

και παρουσιάζει ολικό ελάχιστο στο σημείο $(0, 0)$, δείτε το Σχήμα 3.5 . Έχει όμως, πλήθος τοπικών ελαχίστων και μεγίστων. Για παράδειγμα, η μέθοδος Nelder Mead μετασχηματίζει ένα αρχικό simplex με κορυφές τα σημεία $(-1, 0)$, $(0, 0)$ και $(0, 1)$ σε ένα simplex στο οποίο η συνάρτηση Ackley έχει ελάχιστο στην κορυφή με συντεταγμένες $x = (-0.96851623, -0.96850533)$.



Σχήμα 3.4: Οι μετασχηματισμοί του αρχικού simplex από τη μέθοδο Nelder-Mead για τη συνάρτηση Rosenbrock.



Σχήμα 3.5: Η γραφική παράσταση της συνάρτησης Ackley.

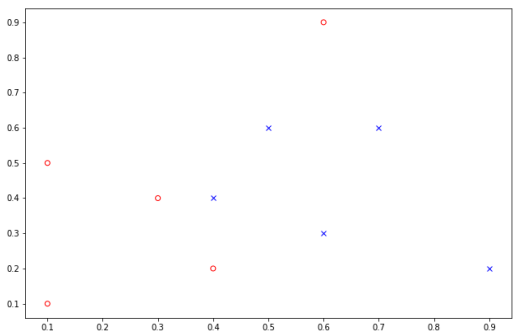
ΚΕΦΑΛΑΙΟ 4

Μέθοδοι βελτιστοποίησης στη Μηχανική Μάθηση

Η μηχανική μάθηση έχει ως αντικείμενο την κατασκευή και μελέτη αλγορίθμων οι οποίοι χρησιμοποιούν δεδομένα με σκοπό να ανακαλύψουν τάσεις και σχέσεις ώστε να κάνουν προβλέψεις και να πάρουν αποφάσεις. Αν και αναφορές στη μηχανική μάθηση υπάρχουν από το 1960, μόνο σχετικά πρόσφατα, με την έκρηξη του πλήθους των δεδομένων, την ανάπτυξη τεχνικών εξόρυξης δεδομένων και την αύξηση στην ταχύτητα των υπολογισμών έχει γίνει σημαντικό εργαλείο σε τομείς της επιστήμης, της τεχνολογίας, της βιομηχανίας και της επιχειρηματικότητας. Εισάγουμε τις βασικές ιδέες χρησιμοποιώντας ένα απλό παράδειγμα, από την εργασία [Higham-Higham] στο οποίο κατασκευάζουμε και εκπαιδεύουμε ένα νευρωνικό δίκτυο. Η εκπαίδευση του απαιτεί τη λύση ενός προβλήματος βελτιστοποίησης σε πολλές διαστάσεις, το οποίο είναι σημαντική υπολογιστική πρόκληση.

4.1 Νευρωνικά Δίκτυα

Έστω ένα σύνολο σημείων, όπως στο σχήμα (4.1), απο τα οποία κάποια είναι κατανεμημένα στην κατηγορία A (συμβολίζονται με κύκλο), ενώ τα υπόλοιπα στην κατηγορία B (σημειώνονται με x). Για παράδειγμα, τα δεδομένα σημεία μπορεί να υποδεικνύουν θέσεις γεώτρησης πετρελαίου σε έναν χάρτη, με την κατηγορία A να δηλώνει επιτυχές αποτέλεσμα. Θέλουμε να κατασκευάσουμε μια απεικόνιση η οποία να λαμβάνει ως είσοδο ένα σημείο στον $\mathbb{R}^2$ και να επιστρέφει είτε κύκλο είτε "x". Μια τέτοια απεικόνιση μπορεί να δημιουργηθεί μέσω τεχνητών νευρωνικών δικτύων, που στην ουσία τους, αποτελούν την επαναλαμβανόμενη εφαρμογή μιας απλής, μη γραμμικής συνάρτησης. Θα χρησιμοποιήσουμε

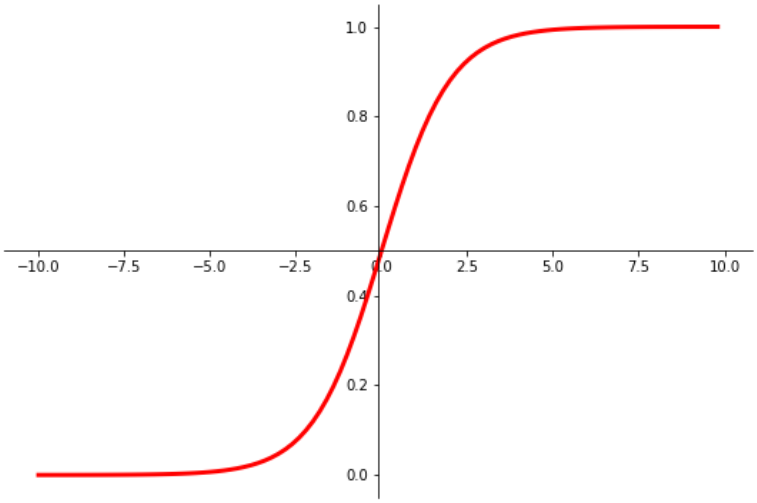


Σχήμα 4.1: Κατηγοριοποίηση σημείων του $\mathbb{R}^2$. Οι κύκλοι δηλώνουν σημεία κατηγορίας A και οι ετικέτες "x" σημεία κατηγορίας B.

τη *σιγμοειδή συνάρτηση*

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{4.1}$$

το γράφημα της οποίας φαίνεται στο Σχήμα 4.2 ως μια ομαλή εκδοχή μιας συνάρτησης σκαλοπατιού (step



Σχήμα 4.2: Η σιγμοειδής συνάρτηση (4.1).

function) η οποία μιμείται την συμπεριφορά ενός νευρώνα στον εγκέφαλο. Η σιγμοειδής συνάρτηση έχει τη χρήσιμη ιδιότητα

$$\sigma'(x) = \sigma(x)(1-\sigma(x)).$$

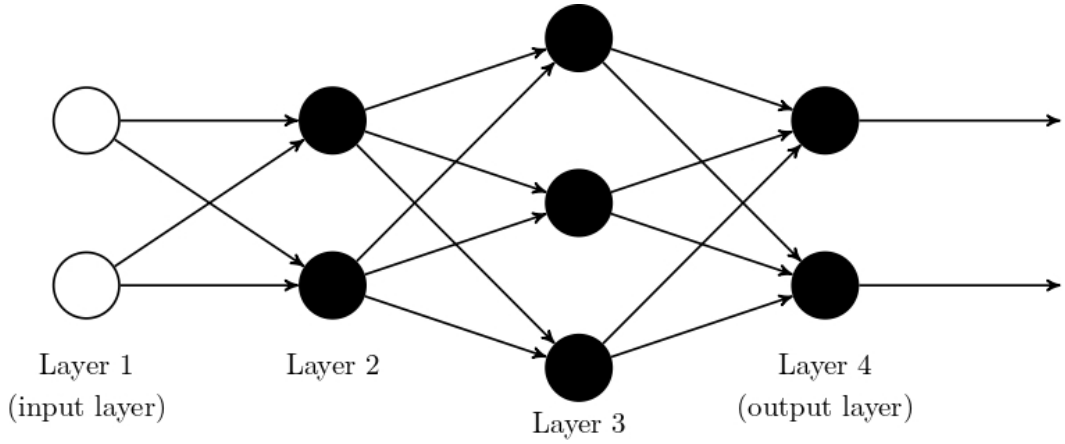
Για την απλούστευση του συμβολισμού, αν $z \in \mathbb{R}^m$ τότε ορίζουμε $\sigma(z) \in \mathbb{R}^m$ ως το διάνυσμα με συνιστώσες $(\sigma(z))_i = \sigma(z_i)$, $i = 1, \ldots, m$. Σε κάθε επίπεδο (layer) κάθε νευρώνας λαμβάνει ως είσοδο μια πραγματική τιμή απο κάθε νευρώνα του προηγούμενου layer και παράγει μια πραγματική τιμή, την οποία μεταδίδει σε κάθε νευρώνα του επόμενου layer. Στο επίπεδο εισόδου (input layer) δεν υπάρχει "προηγούμενο layer" και κάθε νευρώνας λαμβάνει το διάνυσμα εισόδου (input vector). Στο επίπεδο εξόδου (output layer) δεν υπάρχει επόμενο layer και αυτοί οι νευρώνες παρέχουν την συνολική έξοδο (overall output). Τα layers μεταξύ αυτών των δυο ονομάζονται *κρυμμένα* ή *υπολογιστικά* επίπεδα (hidden layers). Αν a είναι το διάνυσμα των τιμών που παράγουν οι νευρώνες σε ένα επίπεδο, τότε το διάνυσμα των τιμών που παράγουν οι νευρώνες στο επόμενο επίπεδο έχει τη μορφή

$$\sigma(Wa + b), \tag{4.2}$$

όπου W είναι ο πίνακας των *βαρών* (weights) and b το διάνυσμα των *μεροληψιών* (biases). Φυσικά, ο αριθμός των στηλών του W ισούται με τον αριθμό των νευρώνων του προηγούμενο επιπέδου και ο αριθμός των γραμμών ισούται με τον αριθμό των νευρώνων στο επόμενο επίπεδο.

Υποθέτουμε πως το δίκτυο έχει L layers, με 1 και L να είναι τα input και output layers, αντίστοιχα. Για το layer l, $l = 1, 2, \ldots, L$, σημειώνουμε με n_l τον αριθμό των νευρώνων του. Έτσι, n_1 είναι η διάσταση των δεδομένων εισόδου. Συνολικά, το δίκτυο μετασχηματίζει δεδομένα από τον $\mathbb{R}^{n_1}$ στον χώρο $\mathbb{R}^{n_L}$. Ο $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$ συμβολίζει τον πίνακα των βαρών του layer l. Πιο αναλυτικά $w^{[l]}_{jk}$ είναι το βάρος του νευρώνα j στο layer l που εφαρμόζεται στην έξοδο απο τον νευρώνα k στο layer $l-1$. Ομοίως, $b^{[l]} \in \mathbb{R}^{n_l}$ το διάνυσμα μεροληψιών του layer l, έτσι ώστε ο νευρώνας j στο layer l χρησιμοποιεί τη μεροληψία $b^{[l]}_j$.

Παράδειγμα. Στο Σχήμα 4.3 έχουμε ένα τεχνητό νευρωνικό δίκτυο με 4 layers. Το πρώτο (input) layer αναπαρίσταται απο δυο κύκλους, επειδή τα σημεία εισόδου (input data points) έχουν δυο συνιστώσες. Το δεύτερο layer έχει δυο κύκλους υποδεικνύοντας ότι δυο νευρώνες έχουν τεθεί σε λειτουργία. Τα βέλη απο το layer 1 στο layer 2 υποδεικνύουν πως και οι δυο συνιστώσες διατίθενται στους δυο νευρώνες του layer 2. Τα weights και οι biases του layer 2 μπορούν να αναπαρασταθούν απο έναν πίνακα $W^{[2]} \in \mathbb{R}^{2\times 2}$



Σχήμα 4.3: Ένα δίκτυο με 4 layers.

και ένα διάνυσμα $b^{[2]} \in \mathbb{R}^2$, αντίστοιχα, εφόσον το σημείο εισόδου (input data) έχει την μορφή $x \in \mathbb{R}^2$. Το output απο το layer 2, εφαρμόζοντας την σιγμοειδή συνάρτηση, έχει την μορφή

$$\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) \in \mathbb{R}^3.$$

Τέλος, το τέταρτο (output) layer έχει δυο νευρώνες, όπου ο καθένας λαμβάνει ως είσοδο ένα διάνυσμα του $\mathbb{R}^3$, επομένως τα weights και οι biases αυτού του layer μπορούν να αναπαρασταθούν από έναν πίνακα $W^{[4]} \in \mathbb{R}^{2\times 3}$ και ενός διανύσματος $b^{[4]} \in \mathbb{R}^2$, αντίστοιχα. Το output του layer 4, και επομένως όλου του δικτύου, έχει την μορφή

$$F(x) = \sigma(W^{[4]}\sigma(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}) + b^{[4]}) \in \mathbb{R}^2,$$

η οποία ορίζει μια συνάρτηση $F : \mathbb{R}^2 \to \mathbb{R}^2$ ως προς τις 23 παραμέτρους της— τα στοιχεία των πινάκων βαρών και τα στοιχεία των διανυσμάτων μεροληψιών. Στόχος μας είναι να παράγουμε έναν *κατηγοριοποιητή* (classifier) που βασίζεται στα δεδομένα του Σχήματος 4.1. Αυτό επιτυγχάνεται βελτιστοποιώντας ως προς τις παραμέτρους. Απαιτούμε η $F(x)$ να είναι κοντά στο $[1, 0]^T$ για δεδομένα της κατηγορίας A και κοντά στο $[0, 1]$ για δεδομένα της κατηγορίας B. Τότε, ένα νέο σημείο $x \in \mathbb{R}^2$ θα ήταν λογικό να το κατατάξουμε σύμφωνα με την μεγαλύτερη συνιστώσα της $F(x)$: στην κατηγορία A εάν $F_1(x) > F_2(x)$ και στην κατηγορία B εάν $F_1(x) < F_2(x)$ με κάποιες επιπλέον κανόνες στην περίπτωση που οι συνιστώσες είναι περίπου ίσες. Αυτή η απαίτηση στην F μπορεί να καθοριστεί μέσω μιας *συνάρτησης κόστους* (cost function). Συμβολίζοντας τα δέκα σημεία του Σχήματος 4.1 με $\left(x^{\{i\}}\right)_{i=1}^{10}$ (training points), θέτουμε $y(x^{\{i\}})$ την αναμενόμενη έξοδο, δήλαδή,

$$y(x^{\{i\}}) = \begin{cases} [1, 0], & \text{αν το } x^{\{i\}} \text{ είναι στην κατηγορία A.} \\ [0, 1], & \text{αν το } x^{\{i\}} \text{ είναι στην κατηγορία B.} \end{cases}$$

Τότε η συνάρτηση κόστους θα μπορούσε να έχει την μορφή

$$Cost(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]}) = \frac{1}{10}\sum_{i=1}^{10}\frac{1}{2}||y(x^{\{i\}}) - F(x^{\{i\}})||_2^2. \tag{4.3}$$

Ο όρος $\frac{1}{2}$ χρησιμοποιείται για ευκολία στην παραγώγιση. Η συνάρτηση κόστους εξαρτάται, φυσικά, από τα βάρη και τις μεροληψίες. Η συγκεκριμένη μορφή με την οποία δίνεται η cost function (4.3), όπου η απόκλιση (discrepancy) από το τις επθυμητές τιμές μετράται απο τον μέσο όρο της τετραγωνισμένης Ευκλείδειας νόρμας στα δεδομένα, αναφέρεται συχνά ως *τετραγωνική συνάρτηση κόστους*. Στη γλώσσα της θεωρίας βελτιστοποίησης, η συνάρτηση κόστους αναφέρεται ως *αντικειμενική συνάρτηση* (objective function).

Η επιλογή των βαρών και μεροληψιών με τρόπο που να ελαχιστοποιεί τη συνάρτηση κόστους αναφέρεται ως *εκπαίδευση του δικτύου* (training thenetwork). Αξίζει να σημειωθεί πως η ανακλιμάκωση της objective function δεν αλλάζει το πρόβλημα βελτιστοποίησης, δηλαδή θα πρέπει να καταλήξουμε στον ίδιο ελαχιστοποιητή αν αλλάξουμε τη συνάρτηση κόστους σε αC, για οποιαδήποτε μη μηδενική τιμή της παραμέτρου α.

Δεδομένου ενός διανύσματος εισόδου $x \in \mathbb{R}^{n_1}$ μπορούμε να συνοψίσουμε την δράση (action) του δικτύου ορίζοντας το $a_j^{[l]}$ ως την έξοδο απο τον νευρώνα j στο layer l. Επομένως έχουμε

$$a^{[1]} = x \in \mathbb{R}^{n_1}, \tag{4.4}$$

$$a^{[l]} = \sigma(W^{[k]}a^{[l-1]} + b^{[l]}) \in \mathbb{R}^{n_l}, \quad l = 0, 1, \ldots, L. \tag{4.5}$$

Ας υποθέσουμε ότι έχουμε N training points στον $\mathbb{R}^{n_1}$, $\{x^{\{i\}}\}_{i=1}^N$, για τα οποία ορίζονται επιθυμητές τιμές εξόδου $\{y(x^{\{i\}})\}_{i=1}^N$ στον $\mathbb{R}^{n_L}$. Γενικεύοντας την (4.3) η τετραγωνική συνάρτηση κόστους που επιθυμούμε να ελαχιστοποιήσουμε έχει τη μορφή

$$\text{Cost} = \frac{1}{10}\sum_{i=1}^{N}\frac{1}{2}||y(x^{\{i\}}) - a^{[L]}(x^{\{i\}})||_2^2, \tag{4.6}$$

όπου, για να διατηρήσουμε τον συμβολισμό υπό έλεγχο, δεν έχουμε δηλώσει ρητά ότι το κόστος είναι συνάρτηση όλων των βαρών και μεροληψιών.

4.2 Στοχαστική Μέθοδος Απότομης Καθόδου (Stochastic Gradient Descent)

Για να εκπαιδεύσουμε ένα δίκτυο αρκεί να επιλέξουμε τις παραμέτρους του, δηλαδή τα βάρη και τις μεροληψίες, που ελαχιστοποιούν τη συνάρτηση κόστους. Τα βάρη και οι μεροληψίες είναι, βέβαια, πίνακες και διανύσματα, αντίστοιχα, αλλά προς το παρόν είναι πιο χρήσιμο να σκεφτούμε τα στοιχεία τους ως συνιστώσες ενός μοναδικού διανύσματος, το οποίο θα συμβολίζουμε με p. Στο παράδειγμα του Σχήματος 4.3 το δίκτυο έχει συνολικά 23 βάρη και μεροληψίες. Έτσι, $p \in \mathbb{R}^{23}$. Γενικά, υποθέτουμε ότι $p \in \mathbb{R}^s$ και γράφουμε την συνάρτηση κόστους (4.3) ως $\text{Cost}(p)$ έτσι ώστε να δώσουμε έμφαση στην εξάρτησή της απο τις παραμέτρους. Επομένως, $\text{Cost} : \mathbb{R}^s \to \mathbb{R}$.

Για το πρόβλημα της ελαχιστοποίησης της συνάρτησης κόστους θα χρησιμοποιήσουμε τη μέθοδο της απότομης καθόδου (steepest descent) που είδαμε ήδη στο Κεφάλαιο 2. Θυμόμαστε ότι η μέθοδος αυτή υπολογίζει μια ακολουθία διανυσμάτων στον $\mathbb{R}^s$ που, υπό προϋποθέσεις, συγκλίνει στον ελαχιστοποιητή της αντικειμενικής συνάρτησης, στην προκειμένη περίπτωση της συνάρτησης κόστους. Έστω p ένα στοιχείο αυτής της ακολουθίας. Πώς θα μπορούσαμε να επιλέξουμε τη διαταραχή (perturbation), Δp, έτσι ώστε το διάνυσμα $p + \Delta p$ να αντιπροσωπεύει μια βελτίωση; Αν το Δp είναι μικρό, τότε αγνοώντας τους όρους τάξης $||\Delta p||^2$, το ανάπτυγμα Taylor μας δίνει

$$\text{Cost}(p + \Delta p) \approx \text{Cost}(p) + \sum_{r=1}^{s}\frac{\partial \text{Cost}(p)}{\partial p_r}\Delta p_r \tag{4.7}$$

Εδώ, $\partial\text{Cost}(p)/\partial p_r$ συμβολίζει τη μερική παράγωγο της συνάρτησης κόστους ως προς την r συνιστώσα του p. Για ευκολία, $\nabla\text{Cost}(p) \in \mathbb{R}^s$ θα συμβολίζει το διάνυσμα των μερικών παραγώγων, γνωστό ως *κλίση*, έτσι ώστε

$$(\nabla\text{Cost}(p))_r = \frac{\partial \text{Cost}(p)}{\partial p_r}.$$

Τότε η (4.7) γίνεται

$$\text{Cost}(p + \Delta p) \approx \text{Cost}(p) + \nabla\text{Cost}(p)^T \Delta p. \tag{4.8}$$

Στόχος μας είναι να μειώσουμε την τιμή της συνάρτησης κόστους. Η σχέση (4.8) μας παρακινεί να επιλέξουμε τη διαταραχή Δp έτσι ώστε να κάνουμε το $\nabla\text{Cost}(p)^T \Delta p$ όσο περισσότερο αρνητικό γίνεται. Μπορούμε να αντιμετωπίσουμε αυτό το πρόβλημα μέσω της ανισότητας Cauchy-Schwarz, από την οποία έχουμε, για κάθε $f, g \in \mathbb{R}^s$ ότι $f^T g \geq -\|f\|_2 \|g\|_2$. Έτσι, η πιο αρνητική τιμή που μπορεί να λάβει το $f^T g$ είναι $-\|f\|_2 \|g\|_2$, το οποίο συμβαίνει όταν $f = -g$. Έτσι, βασιζόμενοι στην (4.8), θα πρέπει να επιλέξουμε το Δp ως ένα διάνυσμα προς την κατεύθυνση $-\nabla\text{Cost}(p)$. Έχοντας υπόψη πως η σχέση (4.8) είναι μια προσέγγιση που αφορά μόνο μικρά Δp, θα περιοριστούμε σε ένα μικρό βήμα προς αυτή την κατεύθυνση. Αυτό οδηγεί στην ενημέρωση (update) της προσέγγισης p

$$p \to p - \eta \nabla\text{Cost}(p). \tag{4.9}$$

Εδώ, η είναι μια μικρή θετική σταθερά, ένα μικρό *βήμα* που, σε αυτό το πλαίσιο, είναι γνωστό ως *ρυθμός μάθησης* (learning rate). Αυτή η εξίσωση ουσιαστικά ορίζει τη μέθοδο καθόδου. Επιλέγουμε ένα αρχικό διάνυσμα και επαναλαμβάνουμε με την (4.9) έως ότου ικανοποιηθεί κάποιο κριτήριο τερματισμού ή μέχρι ο αριθμός των επαναλήψεων να υπερβεί τον υπολογιστικό προϋπολογισμό.

Η συνάρτηση κόστους (4.6) περιλαμβάνει ένα άθροισμα μεμονωμένων όρων πάνω στα training data. Επομένως, η μερική παράγωγος $\nabla\text{Cost}(p)$ είναι ένα άθροισμα πάνω στα training data των επιμέρους μερικών παραγώγων. Συγκεκριμένα, αν

$$C_{x^{\{i\}}} = \frac{1}{2} ||y(x^{\{i\}}) - a^{[L]}(x^{\{i\}})||_2^2, \tag{4.10}$$

τότε από την (4.6) έχουμε

$$\nabla\text{Cost}(p) = \frac{1}{N} \sum_{i=1}^{N} \nabla C_{x^{\{i\}}}(p). \tag{4.11}$$

Όταν έχουμε ένα μεγάλο αριθμό παραμέτρων ή/και μεγάλο αριθμό training points, ο υπολογισμός του διανύσματος κλίσης (4.11) σε κάθε επανάληψη της μεθόδου καθόδου (4.9) μπορεί να είναι απαγορευτικά ακριβός. Μια πολύ φθηνότερη εναλλακτική είναι η αντικατάσταση του μέσου όρου των επιμέρους κλίσεων σε όλα τα training points από την κλίση σε ένα μόνο, τυχαίας επιλογής, training point. Αυτό οδηγεί στην απλούστερη μορφή του αλγόριθμου που ονομάζεται *στοχαστική μέθοδος απότομης καθόδου* (stochastic gradient method). Ένα μόνο βήμα αλγορίθμου αυτού μπορεί να συνοψιστεί ως εξής:

1. Επιλέγουμε τυχαία έναν ακέραιο i από το σύνολο $\{1, 2, 3, \ldots, N\}$
2. Ενημερώνουμε την προσέγγιση p μέσω της σχέσης

$$p \to p - \eta \nabla C_{x^{\{i\}}}(p). \tag{4.12}$$

Με λίγα λόγια, σε κάθε βήμα, η stochastic gradient method χρησιμοποιεί ένα τυχαίο training point ως αντιπρόσωπο ολόκληρου του training set. Σημειώνουμε ότι, ακόμη και για πολύ μικρές τιμές του βήματος η, η σχέση (4.12) δεν εγγυάται ότι θα μειώσει *συνολικά* τη συνάρτηση κόστους—έχουμε ανταλλάξει τον μέσο όρο με ένα μόνο δείγμα. Ως εκ τούτου, αν και η ονομασία stochastic gradient descent χρησιμοποιείται ευρέως για τον συγκεκριμένο αλγόριθμο, προτιμούμε να χρησιμοποιούμε τον όρο *stochastic gradient*.

Η εκδοχή της στοχαστικής μεθόδου απότομης καθόδου που εισαγάγαμε στην (4.12) είναι η απλούστερη από ένα μεγάλο εύρος δυνατοτήτων. Ειδικότερα, ο δείκτης i στην (4.12) επιλέχθηκε με δειγματοληψία με αντικατάσταση με αποτέλεσμα μετά την χρήση του συγκεκριμένου training point αυτό να είναι εξίσου πιθανό με οποιοδήποτε άλλο να επιλεχθεί ως αντιπρόσωπος στο επόμενο βήμα. Μια εναλλακτική είναι η δειγματοληψία χωρίς αντικατάσταση, δηλαδή να επιλέξουμε με τυχαία σειρά κάθε ένα από τα N training points. Θα μπορούσαμε να συνοψίσουμε N βήματα αυτού του αλγορίθμου ως εξής:

1. Ανακατεύουμε τους ακέραιους $\{1, 2, 3, \dots, \}$ σε μια νέα σειρά $\{k_1, k_2, k_3, \dots, k_N\}$
2. Για $i = 1$ μέχρι N εκτελούμε τις ενημερώσεις της προσέγγισης p

$$p \to p - \eta \nabla C_{x^{\{k_i\}}}(p). \tag{4.13}$$

Αν θεωρήσουμε ότι η στοχαστική μέθόδος απότομης καθόδου προσεγγίζει τον μέσο όρο πάνω σε όλα τα training points στην (4.11) από ένα μόνο δείγμα, τότε είναι φυσικό να εξεταστεί μια ακόμα παραλλαγή όπου χρησιμοποιούμε ένα μικρό μέσο δείγμα. Συγκεκριμένα, για κάποιο $m \ll N$ κάνουμε τα εξής βήματα:

1. Επιλέγουμε m ακέραιους $k_1, k_2, \dots, k_m$ μεταξύ των $\{1, 2, \dots, N\}$.
2. Κάνουμε τις ενημερώσεις της προσέγγισης p

$$p \to p - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla C_{x^{\{k_i\}}}(p). \tag{4.14}$$

Σε αυτή την παραλλαγή, το σύνολο $\{x^{\{k_i\}}\}_{i=1}^m$ είναι γνωστό ως *μίνι-παρτίδα* (mini batch). Υπάρχει ακόμα μια εναλλακτική χωρίς αντικατάσταση, όπου, υποθέτοντας ότι $N = Km$ για κάποιο K, χωρίζουμε το training set σε K διακριτές μίνι-παρτίδες και χρησιμοποιούμε διαδοχικά κάθε μία από αυτές.

4.3 Διάδοση προς τα πίσω (back propagation)

Μπορούμε τώρα να εφαρμόσουμε τη στοχαστική μέθοδο απότομης καθόδου για να εκπαιδεύσουμε ένα τεχνητό νευρωνικό δίκτυο. Κάνουμε την μετάβαση από το γενικό διάνυσμα παραμέτρων, p, στους πίνακες βαρών και τα διανύσματα μεροληψιών. Στόχος μας αποτελεί ο υπολογισμός των μερικών παραγώγων της συνάρτησης κόστους ως προς τα $w_{jk}^{[l]}$ και $b_j^{[l]}$. Προσπαθούμε να εκμεταλλευτούμε την δομή της συνάρτησης κόστους: επειδή η (4.6) είναι γραμμικός συνδυασμός μεμονομένων όρων πάνω σε όλα τα training data, το ίδιο θα ισχύει και για τις μερικές παραγώγους της. Έτσι, επικεντρώνουμε την προσοχή μας στον υπολογισμό αυτών των μερικών παραγώγων.

Θεωρούμε την $C_{x^{\{i\}}}$ στην (4.10) ως μια συνάρτηση των βαρών και μεροληψιών. Επομένως, μπορούμε να γράψουμε

$$C = \frac{1}{2}\|y - a^{[L]}\|_2^2. \tag{4.15}$$

Απο την (4.5) θυμόμαστε ότι το $a^{[L]}$ είναι η έξοδος του τεχνητού νευρωνικού δικτύου. Η εξάρτηση της C απά τα βάρη και τις μεροληψίες επιτυγχάνεται μόνο μέσω του $a^{[L]}$. Για να εξαγάγουμε αξιόλογα αποτελέσματα για τις μερικές παραγώγους, είναι χρήσιμο να να εισαγάγουμε δυο επιπλέον σύνολα μεταβλητών. Πρώτα,

$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]} \in \mathbb{R}^{n_l}, \quad l = 2, 3, \dots, L. \tag{4.16}$$

Το $z_j^{[l]}$ αντιπροσωπεύει τη σταθμισμένη είσοδο του νευρώνα j στο layer l. Η θεμελιώδης σχέση (4.5), που μεταφέρει πληροφορίες μέσω του δικτύου, μπορεί να γραφεί ως

$$a^{[l]} = \sigma(z^{[l]}), \quad l = 2, 3, \dots, L. \tag{4.17}$$

Ορίζουμε ακόμα το διάνυσμα $\delta^{[l]} \in \mathbb{R}^{n_l}$ ως

$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}}, \quad 1 \le j \le n_l, \quad 2 \le l \le L. \tag{4.18}$$

Η παραπάνω σχέση, γνωστή ως *σφάλμα* (error) στον j νευρώνα του l layer, είναι μια ενδιάμεση ποσότητα που χρησιμεύει τόσο στην ανάλυση όσο και στον υπολογισμό. Ωστόσο, σημειώνεται ότι η χρήση του

όρου σφάλματος είναι κάπως διφορούμενη. Σε ένα γενικό επίπεδο υπολογισμού δεν είναι ξεκάθαρο πόσο "ευθύνεται" κάθε νευρώνας για αποκλίσεις στην έξοδο. Επίσης, στο επίπεδο εξόδου L, η σχέση (4.18) δεν ποσοτικοποιεί αυτές τις αποκλίσεις άμεσα. Η ιδέα της αναφοράς στο $\delta_j^{[l]}$ στην (4.18) ως σφάλμα, φαίνεται να έχει προκύψει επειδή η συνάρτηση κόστους μπορεί να έχει ελάχιστο μόνο αν όλες οι μερικοί παράγωγοι της μηδενίζονται, δηλαδή θέλουμε $\delta_j^{[l]} = 0$ για όλα τα j. Το $\delta_j^{[l]}$ μετρά την ευαισθησία της συνάρτησης κόστους ως προς τη σταθμισμένη είσοδο για τον νευρώνα j στο layer l.

Ορίζουμε τώρα το *γινόμενο Hadamard*. Αν $x, y \in \mathbb{R}^n$ τότε το γινόμενο Hadamard $x \circ y \in \mathbb{R}^n$ ορίζεται από τη σχέση $(x \circ y)_i = x_i y_i$. Το γινόμενο Hadamard είναι, δηλαδή, ο πολλαπλασιασμός κατά ζεύγη των αντίστοιχων συνιστωσών των διανυσμάτων x και y.

Λήμμα 4.1. *Έχουμε τις σχέσεις*

$$\delta^{[L]} = \sigma'(z^{[L]}) \circ (a^L - y), \tag{4.19}$$

$$\delta^{[l]} = \sigma'(z^{[l]}) \circ (W^{[l+1]})^T \delta^{[l+1]}, \quad 2 \leq l \leq L-1, \tag{4.20}$$

$$\frac{\partial C}{\partial b_j^{[l]}} = \delta_j^{[l]}, \quad 2 \leq l \leq L, \tag{4.21}$$

$$\frac{\partial C}{\partial w_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]}, \quad 2 \leq l \leq L. \tag{4.22}$$

Απόδειξη. Αρχίζουμε αποδεικνύοντας την (4.19). Η σχέση (4.17) με $l = L$ δείχνει πως τα $z_j^{[L]}$ και $a_j^{[L]}$ σχετίζονται μέσω της σχέσης $a^{[L]} = \sigma(z^{[L]})$, και έτσι

$$\frac{\partial a_j^{[L]}}{\partial z_j^{[L]}} = \sigma'(z_j^{[L]}).$$

Ακόμα, από την (4.15) έχουμε

$$\frac{\partial C}{\partial a_j^{[L]}} = \frac{\partial}{\partial a_j^{[L]}} \frac{1}{2} \sum_{k=1}^{n_L} (y_k - a_k^{[L]})^2 = -(y_j - a_j^{[L]}).$$

Χρησιμοποιώντας τον κανόνα της αλυσίδας έχουμε

$$\delta_j^{[L]} = \frac{\partial C}{\partial z_j^{[L]}} = \frac{\partial C}{\partial a_j^{[L]}} \frac{\partial a_j^{[L]}}{\partial z_j^{[L]}} = (a_j^{[L]} - y_j)\sigma'(z_j^{[L]}),$$

το οποίο είναι η (4.19) κατά συνιστώσες. Για την απόδειξη της (4.20) θα χρησιμοποιήσουμε τον κανόνα της αλυσίδας για να μετατρέψουμε το $z_j^{[l]}$ σε $\{z_k^{[l+1]}\}_{k=1}^{n_{l+1}}$. Χρησιμοποιώντας τον ορισμό (4.18) έχουμε

$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \frac{\partial C}{\partial z_k^{[l+1]}} \frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} \frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}}. \tag{4.23}$$

Από την (4.16) γνωρίζουμε πως τα $z_k^{[l+1]}$ και $z_j^{[l]}$ συνδέονται μέσω της

$$z_k^{[l+1]} = \sum_{s=1}^{n_l} w_{ks}^{[l+1]} \sigma(z_s^{[l]}) + b_k^{[l+1]}.$$

Έτσι,

$$\frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}} = w_{kj}^{[l+1]} \sigma'(z_j^{[l]}).$$

Αντικαθιστώντας στην (4.23) έχουμε

$$\delta_j^{[l]} = \sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} w_{kj}^{[l+1]} \sigma'(z_j^{[l]}),$$

το οποίο μπορεί να γραφεί

$$\delta_j^{[l]} = \sigma'(z_j^{[l]})((W^{[l+1]})^T \delta^{[l+1]})_j,$$

το οποίο είναι η (4.20) κατά συνιστώσες. Για την απόδειξη της (4.21) παρατηρούμε ότι από τις (4.16) και (4.17) το $z_j^{[l]}$ σχετίζεται με το $b_j^{[l]}$ μέσω της

$$z_j^{[l]} = (W^{[l]}\sigma(z^{[l-1]}))_j + b_j^{[l]}.$$

Εφόσον το $z^{[l-1]}$ δεν εξαρτάται απο το $b_j^{[l]}$, έχουμε πως $\frac{\partial z_j^{[l]}}{\partial b_j^{[l]}} = 1$. Συνεπώς, από τον κανόνα της αλυσίδας προκύπτει ότι

$$\frac{\partial C}{\partial b_j^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial b_j^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} = \delta_j^{[l]}.$$

Η (4.21) προκύπτει με χρήση του ορισμού (4.18). Για την απόδειξη της (4.22) ξεκινάμε με την (4.16) κατά συνιστώσες

$$z_j^{[l]} = \sum_{k=1}^{n_{l-1}} w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]},$$

η οποία δίνει

$$\frac{\partial z_j^{[l]}}{\partial w_{jk}^{[l]}} = a_k^{[l-1]}, \tag{4.24}$$

ανεξάρτητα του j, και

$$\frac{\partial z_s^{[l]}}{\partial w_{jk}^{[l]}} = 0, \quad s \neq j \tag{4.25}$$

Οι (4.24) και (4.25) προκύπτουν επειδή ο j -νευρώνας στο layer l χρησιμοποιεί τα βάρη μόνο από την j- σειρά του πίνακα $W^{[l]}$ και εφαρμόζει αυτά τα βάρη γραμμικά. Τότε, από τον κανόνα της αλυσίδας οι (4.24), (4.25) δίνουν

$$\frac{\partial C}{\partial w_{jk}^{[l]}} = \sum_{s=1}^{n_l} \frac{\partial C}{\partial z_s^{[l]}} \frac{\partial z_s^{[l]}}{\partial w_{jk}^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial w_{jk}^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} a_k^{[l-1]} = \delta_j^{[l]} a_k^{[l-1]},$$

όπου στο τελευταίο βήμα χρησιμοποιήθηκε ο ορισμός (4.18). □

Από τις (4.4), (4.16) και (4.17) βλέπουμε ότι η έξοδος $a^{[L]}$ μπορεί να υπολογισθεί από μια διάσχιση προς τα εμπρός (forward pass) του δικτύου, υπολογίζοντας τα $a^{[1]}, z^{[2]}, a^{[2]}, z^{[3]}, \ldots, a^{[L]}$, με αυτή τη σειρά. Έχοντας κάνει αυτό, από την (4.19) έχουμε ότι το $\delta^{[L]}$ είναι άμεσα διαθέσιμο. Έτσι, απο την (4.20), τα $\delta^{[L-1]}, \delta^{[L-2]}, \ldots, \delta^{[2]}$, μπορούν να υπολογισθούν από μια διάσχιση προς τα πίσω (backward pass). Από τις (4.21) και (4.22) έχουμε πρόσβαση στις μερικές παραγώγους. Ο υπολογισμός των κλίσεων κατά αυτόν τον τρόπο είναι γνωστός ως *διάδοση προς τα πίσω*.

Για να κατανοήσουμε περισσότερο τις σχέσεις (4.21) και (4.22) στο Λήμμα 4.1, είναι χρήσιμος ο θεμελιώδης ορισμός μιας μερικής παραγώγου. Η ποσότητα $\partial C/\partial w_{jk}^{[l]}$ μετρά πως αλλάζει η C όταν διαταράξουμε λίγο το $w_{jk}^{[l]}$. Στο Σχήμα 4.4, φαίνεται η δράση του βάρους $w_{43}^{[3]}$. Μια αλλαγή σε αυτό δεν επηρεάζει την έξοδο των προηγούμενων επιπέδων, επομένως για να υπολογίσουμε την $\partial C/\partial w_{43}^{[3]}$ δεν χρειαζόμαστε

τις μερικές παραγώγους στα προηγούμενα επίπεδα. Θα έπρεπε, ωστόσο, να μπορούμε να εκφράσουμε την $\partial C/\partial w_{43}^{[3]}$ με όρους των μερικών παραγώγων των επόμενων επιπέδων. Συγκεκριμένα, η ενεργοποίηση (activation) που τροφοδοτεί τον τέταρτο νευρώνα στο επίπεδο 3 είναι $z_4^{[3]}$ και, εξ' ορισμού, το $\delta_4^{[3]}$ μετρά την "ευαισθησία" της C ως προς αυτή την είσοδο. Τροφοδοτώντας αυτόν τον νευρώνα παίρνουμε $w_{43}^{[3]}a_3^{[2]} + \textit{σταθερά}$, επομένως έχει νόημα η

$$\frac{\partial C}{\partial w_{43}^{[3]}} = \delta_4^{[3]} a_3^{[2]}.$$

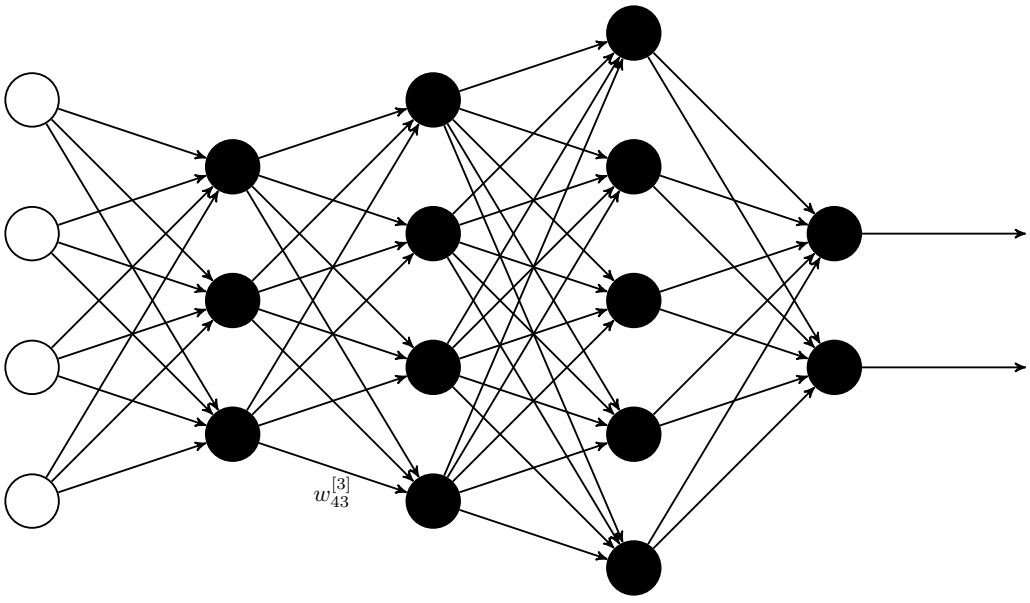
Όμοια, για τις μεροληψίες, το $b_4^{[3]} + \textit{σταθερά}$ τροφοδοτεί τον νευρώνα, το οποίο εξηγεί γιατί

$$\frac{\partial C}{\partial b_4^{[3]}} = \delta_4^{[3]} \times 1.$$

Μπορούμε να αποφύγουμε το γινόμενο Hadamard στις (4.19) και (4.20) εισάγοντας τους διαγώνιους πίνακες. Έστω $D^{[l]} \in \mathbb{R}^{n_l \times n_l}$ ο διαγώνιος πίνακας με το στοιχείο (i,i) να δίνεται απο το $\sigma'(z_i^{[l]})$. Τότε έχουμε ότι $\delta^{[L]} = D^{[L]}(a^{[L]} - y)$ και $\delta^{[l]} = D^{[l]}(W^{[l+1]})^T\delta^{[l+1]}$. Μπορούμε να το γράψουμε ως

$$\delta^{[l]} = D^{[l]}(W^{[l+1]})^T D^{[l+1]}(W^{[l+2]})^T \cdots D^{[L-1]}(W^{[L]})^T D^{[L]}(a^{[L]} - y)$$

Η σχέση (4.21) δείχνει πως το $\delta^{[l]}$ αντιστοιχεί ακριβώς στην κλίση της cost function ως προς τις biases στο layer l. Αν θεωρήσουμε την $\partial C/\partial w_{jk}^{[l]}$ ως το στοιχείο (j,k) ενός πίνακα μερικών παραγώγων στο layer l, τότε η (4.22) δείχνει πως αυτός ο πίνακας είναι το εξωτερικό γινόμενο $\delta^{[l]}{a^{[l-1]}}^T \in \mathbb{R}^{n_l \times n_{l-1}}$. Μπορούμε τώρα να γράψουμε τον ψευδοκώδικα ενός αλγόριθμου ο οποίος εκπαιδεύει ένα δίκτυο χρη-

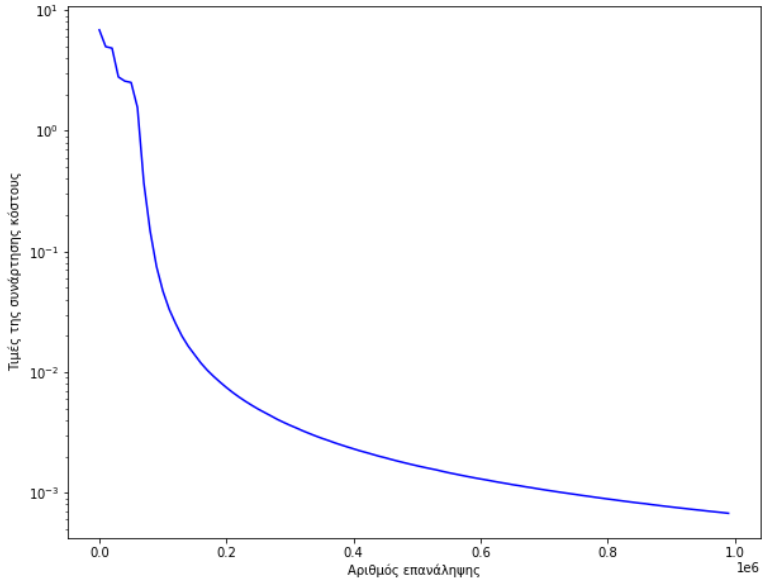


Σχήμα 4.4: Ένα δίκτυο με 5 επίπεδα. Η έξοδος από τον νευρώνα τρία στο επίπεδο δύο σταθμίζεται με το βάρος $w_{43}^{[3]}$ και εισάγεται στον νευρώνα τέσσερα του επιπέδου τρία.

σιμοποιώντας έναν σταθερό αριθμό επαναλήψεων niter της στοχαστικής μεθόδου απότομης καθόδου. Για απλούστευση, θεωρούμε τη βασική εκδοχή (4.13) της μεθόδου, όπου επιλέγονται μεμονωμένα δείγματα με αντικατάσταση. Για κάθε training point, εκτελούμε ένα forward pass του δικτύου προκειμένου να αξιολογηθούν οι ενεργοποιήσεις, οι σταθμισμένοι είσοδοι και η συνολική έξοδος $a^{[L]}$. Έπειτα,

for counter απο 1 μέχρι niter **do**
  επιλέγω ακέραιο k τυχαία απο $\{1, 2, ..., N\}$
  $x^{\{k\}}$ είναι το τρέχον training point
  $a^{[l]} = x^{\{k\}}$
  for l=2 μέχρι L **do**
    $z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$
    $a^{[l]} = \sigma(z^{[l]})$
    $D^{[l]} = diag(\sigma'(z^{[l]}))$
  end for
  $\delta^{[L]} = D^{[L]}(a^{[L]} - y(x^{\{k\}}))$
  for l=L-1 μέχρι 2 **do**
    $\delta^{[l]} = D^{[l]}(W^{[l+1]})^T\delta^{[l+1]}$
  end for
  for l=L μέχρι 2 **do**
    $W^l \to W^{[l]} - \eta\delta^{[l]}a^{[l-1]^T}$
    $b^{[l]} \to b^{[l]} - \eta\delta^{[l]}$
  end for
end for

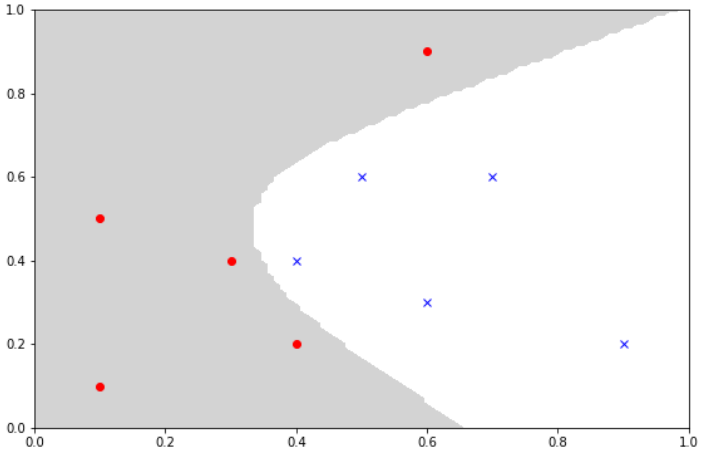
πραγματοποιούμε μία διάσχιση προς τα πίσω (backward pass) για να υπολογίσουμε σφάλματα και να ενημερώσουμε την προσέγγιση του ελαχίστου.
Εκτελώντας τον παραπάνω αλγόριθμο στο παράδειγμα που χρησιμοποιούμε σε αυτό το κεφάλαιο, παίρνουμε τα αποτελέσματα που φαίνονται στο Σχήμα 4.5. Ο κατακόρυφος άξονας δείχνει μια κλιμακούμενη τιμή της συνάρτησης κόστους, ενώ ο οριζόντιος άξονας δείχνει τον αριθμό επανάληψης. Εδώ χρησιμοποιήσαμε τη στοχαστική μέθοδο απότομης καθόδου για να εκπαιδεύσουμε ένα δίκτυο όπως αυτό που φαίνεται στο Σχήμα 4.3 και χρησιμοποιώντας τα δεδομένα του Σχήματος 4.1.



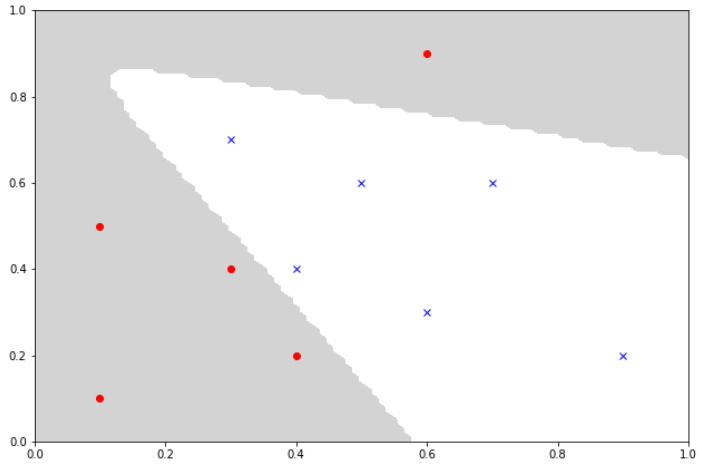
Σχήμα 4.5: Τιμές της συνάρτησης κόστους σε σχέση με τον αριθμό επανάληψης της στοχαστικής μεθόδου απότομης καθόδου.

Για τα δεδομένα στο Σχήμα 4.1 ελαχιστοποιήσαμε τη συνάρτηση κόστους (4.3) ως προς τις 23 παραμέτρους που ορίζουν τα $W[2], W[3], W[4], b[2], b[3]$ και $b[4]$. Για το εκπαιδευμένο δίκτυο, το Σχήμα 4.6 δείχνει το όριο όπου $F_1(x) > F_2(x)$. Έτσι, με αυτήν την προσέγγιση, οποιοδήποτε σημείο στη σκια-

σμένη περιοχή θα αποδοθεί στην κατηγορία A και οποιοδήποτε σημείο στην μη σκιασμένη περιοχή στην κατηγορία B. Το Σχήμα 4.6 δείχνει πώς το δίκτυο ανταποκρίνεται σε πρόσθετα training data. Εδώ, προσθέσαμε ένα ακόμη σημείο της κατηγορίας B, στη θέση $(0.3, 0.7)$, και επαναλαμβάνοντας την διαδικασία βελτιστοποίησης.



Σχήμα 4.6: Οπτικοποίηση του output από ένα τεχνητό νευρωνικό δίκτυο που εφαρμόζεται στα δεδομένα στο Σχήμα (4.1).



Σχήμα 4.7: Επανάληψη του πειράματος στο Σχήμα (4.6) με ένα επιπλέον data point.

ΚΕΦΑΛΑΙΟ 5

Υποστηρικτικό Υλικό

5.1 Κώδικας Python Απότομης Μεθόδου Καθόδου

```
import numpy as np
import matplotlib.pyplot as plt

#gradient of f
def grad_f(x):
  return np.array([x[0], 9*x[1]])

#step t
def t_step(x):
  return (x[0]**2 + 81*x[1]**2) / (x[0]**2 + 9*81*x[1]**2)

#initial point
x = np.array([ 9, 1 ])

L = []

for i in range(100):
  t = t_step(x)
  x = x - t * grad_f(x)
  print(x)
  L.append(tuple(x))


plt.scatter(*zip(*L))
plt.title("Steepest Descent Method")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

5.2 Κώδικας Python Γραμμικής Αναζήτησης με Οπισθοχώρηση

```
import math
import numpy as np
import scipy.linalg as la

def f(x):
  return 0.5*x[0]**2 + 4.5*x[1]**2
def grad(x):
  return np.array([x[0], 9*x[1]])

#For backtracking use alpha = 10^{-4}
alpha = 1.0e-4

def linesearch(xk, gk, pk, alpha):
  fk = f(xk)
  slope = gk @ pk

  #Start with lambda = 1
  lam = 1
  while True:
    xnew = xk + lam * pk
  fnew = f(xnew)

  #Wolfe condition satisfied
  if fnew <= fk + alpha * lam * slope:
    return 0, lam
  else:
    if lam == 1:
      #reduce lamp
    lamtmp = -0.5*slope / (fnew-fk-slope)
    else:
    print('Cubic backtracking')
    dl = lam - lamprv
    #page 7
    a = ((fnew-fk-lam*slope) / lam**2 -

      (fprv-fk-lamprv*slope) / lamprv**2 ) / dl

    b = (-lamprv*(fnew-fk-lam*slope) / lam**2 +

      lam*(fprv-fk-lamprv*slope) / lamprv**2) /dl

    if abs(a) < 1.0e-6:
      lamtmp=-slope/(2*b)
    else:
      d = b**2-3*a*slope
      lamtmp = (-b + math.sqrt(d)) / (3*a)

    lamtmp = min(lamtmp, 0.5*lam)
    lamprv=lam
    fprv = fnew
    lam = max(lamtmp, 0.1*lam)
    print(lam)

# Stoping criterion tolerance
tol = 0.5e-6

#Start at (9,1).Chooce as descent direction the vector
#p = -grad(x)/ || grad(x) ||
k = 0
xk = np.array([7.,5.])
```

```
K = [xk]
while True:
#Chooce as descent direction the vector p = -grad(x) / || grad(x) ||
gk = grad(xk)
pk = -gk/la.norm(gk)

#Find step size
err, lam = linesearch(xk, gk, pk, alpha)
if err: break

#Printing
k = k + 1
xnew = xk + lam*pk
K.append(tuple(xk))
print(k, xnew)

#Check for convergence
if la.norm(xnew - xk) < tol: break

#Continue with the next step
np.copyto(xk, xnew)

#plot
del K[0]
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
#ax.spines['left'].set_position('center')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
X = [x[0] for x in K]
Y = [x[1] for x in K]
plt.plot(X,Y)
plt.plot(X,Y,'or')
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

5.3 Κώδικας Python Broyden

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv

def F(x):
  return np.array([x[0]+x[1]-3, (x[0])**2+(x[1])**2-9])

x = np.array([1,5])

A = [[1, 1],
   [2, 10]]

L = [x]
def broyden_method(x,F,A):
  for i in range(7):
    s =-np.linalg.solve(A, F(x))

    f = F(x)
    x = x + s
    L.append(tuple(x))
    y = F(x)-f
    A =A+np.outer(y-np.dot(A,s),s)/ (np.dot(s,s))
    print(x)

broyden_method(x, F, A)

#plot
#set axis
ax = plt.gca()
#ax.grid(True)
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

x_val = [x[0] for x in L]
y_val = [x[1] for x in L]
plt.plot(x_val,y_val)
plt.plot(x_val,y_val,'or')
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

5.4 Κώδικας Python Nelder-Mead

```
# nelder-mead for multimodal function optimization
from scipy.optimize import minimize
from numpy.random import rand
from numpy import exp
from numpy import sqrt
from numpy import cos
from numpy import e
from numpy import pi

# objective function
def objective(v):
  x, y = v
  return -20.0 * exp(-0.2 * sqrt(0.5 * (x**2 + y**2)))
    - exp(0.5 * (cos(2 * pi * x) + cos(2 * pi * y))) + e + 20

# define initial guess
pt = np.array([-0.93887527, -1.46324318])

result = minimize(objective, pt, method='nelder-mead')

# summarize the result
print('Status : %s' % result['message'])
print('Total Evaluations: %d' % result['nfev'])
#solution
solution = result['x']
evaluation = objective(solution)
print('Solution: f(%s) = %.5f' % (solution, evaluation))
```

5.5 Κώδικας Network Layers

```
import numpy as np
from numpy.random import randn, randint
from scipy.linalg import norm
import matplotlib.pyplot as plt

#Activation function
def activate(x, W, b):
  return 1/(1+np.exp(-(W@x+b)))

#Cost function
def cost(xd, yd, W2, W3, W4, b2, b3, b4):
  v = np.zeros(10)
  for i in range(10):
    x = xd[:,i]
    a2 = activate(x,  W2, b2)
    a3 = activate(a2, W3, b3)
    a4 = activate(a3, W4, b4)
    v[i] = norm(yd[:,i] - a4)
  return norm(v)**2

#Coordinates of drilling sites
xd = np.array([ [0.1, 0.3, 0.1, 0.6, 0.4, 0.6, 0.5, 0.9, 0.4, 0.7],
            [0.1, 0.4, 0.5, 0.9, 0.2, 0.3, 0.6, 0.2, 0.4, 0.6]])

#Target output
yd = np.array([ [1, 1, 1, 1, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]])

#Initialize weights and biases
W2 = 0.5*randn(2,2);  b2 = 0.5*randn(2)
W3 = 0.5*randn(3,2);  b3 = 0.5*randn(3)
W4 = 0.5*randn(2,3);  b4 = 0.5*randn(2)

#Learning rate
eta = 0.05

#Number of iterations and cost function per iteration
niter = 1000000
costh = np.zeros(niter)

for counter in range(niter):
  k = randint(10)
  x = xd[:,k]

  #Forward pass
  a2 = activate(x,  W2, b2)
  a3 = activate(a2, W3, b3)
  a4 = activate(a3, W4, b4)

  #Backward pass
  delta4 = a4*(1-a4)*(a4-yd[:,k])
  delta3 = a3*(1-a3)*(W4.T@delta4)
  delta2 = a2*(1-a2)*(W3.T@delta3)

  #Gradient step
  W2 = W2 - eta*np.outer(delta2,  x)
  W3 = W3 - eta*np.outer(delta3, a2)
  W4 = W4 - eta*np.outer(delta4, a3)
  b2 = b2 - eta*delta2
  b3 = b3 - eta*delta3
  b4 = b4 - eta*delta4
```

```
    #Save cost
    costh[counter] = cost(xd, yd, W2, W3, W4, b2, b3, b4)

#Plot cost function
step = 10000
ix = np.arange(0,niter,step)
iy = costh[::step]

fig, ax = plt.subplots()
ax.semilogy(ix, iy, color='blue',linestyle='-')
plt.ylabel('Values of Cost function')
plt.xlabel('Number of iteration')
plt.show()
```

Βιβλιογραφία

[1] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 3rd ed. The Johns Hopkins University Press, 1996.

[2] Catherine F. Higham and Desmond J. Higham. Deep Learning: An Introduction for Applied Mathematicians. In: *SIAM Review* 61.4 (2019), pp. 860–891. DOI: 10.1137/18M1165748.

[3] Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. In: *Journal of Global Optimization* 21 (2001). DOI: 10.1023/A:1012771025575.

[4] J.E. Dennis Jr. και Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equation*. SIAM, Philadelphia: Classics in Applied Mathematics 16, 1996.

[5] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. 2019.

[6] J.A. Nelder and R. Mead. A Simplex Method for Function Minimization. In: *Computer Journal* 7 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

[7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2e. New York, NY, USA: Springer, 2006.

[8] Rody Oldenhuis. *Test functions for global optimization algorithms*. 2022. URL: https://github.com/rodyo/FEX-testfunctions/releases/tag/v1.5.

[9] J. M. Ortega και W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial και Applied Mathematics, 2000. DOI: 10.1137/1.9780898719468. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898719468. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898719468.

[10] F. Pedregosa et al. Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[11] Luis Rios and Nikolaos Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. In: *Journal of Global Optimization* 56 (2009), pp. 1247–1293. DOI: 10.1007/s10898-012-9951-y.

[12] Shiliang Sun et al. A Survey of Optimization Methods From a Machine Learning Perspective. In: *IEEE Transactions on Cybernetics* 50 (2020), pp. 3668–3681.

[13] Γ.Δ. Ακρίβης και Β.Α. Δουγαλής. *Εισαγωγή στην Αριθμητική Ανάλυση*. 4η έκδοση. Πανεπιστημιακές Εκδόσεις Κρήτης, 2015.

[14] Β. Α. Δουγαλής. *Διδακτικές σημειώσεις για το μεταπτυχιακό μάθημα 350 Αριθμητική Ανάλυση*. Ηράκλειο Κρήτης: Ινστιτούτο Υπολογιστικών Μαθηματικών, Ερευνητικό Κέντρο Κρήτης, 1987.

ΠΑΡΑΡΤΗΜΑ Α

Παράγωγοι βαθμωτών και διανυσματικών συναρτήσεων

Στο παράρτημα αυτό θα αναφερθούμε με συντομία στην έννοια της παραγώγου για διανυσματικές συναρτήσεις, Το κίνητρό μας είναι η μελέτη μεθόδων για την αριθμητική επίλυση μη γραμμικών συστημάτων n εξισώσεων με n αγνώστους, δηλαδή, συστημάτων της μορφής

$$f_i(x_1, ..., x_n) = 0, \quad i = 1, 2, \ldots, n, \tag{A.1}$$

τα οποία γράφουμε συνήθως στην διανυσματική μορφή

$$F(x) = 0, \tag{A.2}$$

όπου F μια, μη γραμμική, εν γένει, απεικόνιση ενός συνόλου D στον R^n, δηλαδή, $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$, με συνιστώσες $F(x) = (f_1(x), \ldots, f_n(x))^T$ και $x = (x_1, \ldots, x_n)^T$, κατά τα γνωστά.

Προβλήματα που οδηγούν στην επίλυση μη γραμμικών συστημάτων της μορφής (A.1) εμφανίζονται πολύ συχνά στις εφαρμογές. Μια σημαντική πηγή προβλημάτων είναι ο υπολογισμός τοπικών ακροτάτων ενός συναρτησιακού $g : \mathbb{R}^n \to \mathbb{R}$, οπότε $F = \nabla g$, υπό την προϋπόθεση βέβαια ότι η κλίση ∇g υπάρχει και μπορεί να υπολογισθεί για κάθε x. Το υλικό που παρουσιάζεται εδώ βασίζεται στις σημειώσεις [14] και το κλασικό βιβλίο [9].

Σημαντικό ρόλο τόσο στην θεωρία όσο και στις αριθμητικές μεθόδους για την λύση του (A.1) παίζουν οι ιδιότητες παραγωγισιμότητας της F καθώς και διαφόρου τύπου θεωρήματα "μέσης τιμής".

Θα λέμε ότι η απεικόνιση $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι *παραγωγίσιμη* σε ένα σημείο $x \in \text{Int}(D)$ (ακριβέστερα παραγωγίσιμη με την έννοια του Frechet, ή F-παραγωγίσιμη) αν υπάρχει γραμμικός τελεστής $A_x : \mathbb{R}^n \to \mathbb{R}^n$ τέτοιος ώστε για κάποια νόρμα $\|\cdot\|$ του $\mathbb{R}^n$ να ισχύει

$$\lim_{h \to 0} \frac{\|F(x+h) - F(x) - A_x h\|}{\|h\|} = 0. \tag{A.3}$$

Λόγω της ισοδυναμίας των νορμών στον $\mathbb{R}^n$, η ύπαρξη του A_x είναι ανεξάρτητη της νόρμας $\|\cdot\|$. Είναι επίσης προφανές ότι ο ορισμός (A.3) γενικεύει την έννοια της παραγωγισιμότητας πραγματικών συναρτήσεων μιας μεταβλητής. Αν η F είναι παραγωγίσιμη στο x τότε ο γραμμικός τελεστής A_x είναι μοναδικός. Πράγματι, αν υπήρχαν δυο γραμμικοί τελεστές A_1, A_2 τέτοιοι ώστε για τον κάθε ένα να ισχύει η (A.3) θα είχαμε, από την τριγωνική ανισότητα, για κάθε $0 \neq y \in \mathbb{R}^n$, $t \neq 0$,

$$\begin{aligned} \|(A_1 - A_2)y\|/\|y\| \leq \|F(x+ty) - F(x) - A_1(ty)\|/\|ty\| \\ + \|F(x+ty) - F(x) - A_2(ty)\|/\|ty\|, \end{aligned}$$

απο την οποία, θέτοντας $h = ty$ και παίρνοντας $t \to 0$ έχουμε, λόγω της (A.3) ότι $A_1 y = A_2 y, \forall y \in \mathbb{R}^n$, δηλαδή ότι $A_1 = A_2$. Αν λοιπόν η F είναι παραγωγίσιμη στο x θα λέμε ότι ο τελεστής A_x είναι η παράγωγος της F (ακριβέστερα η παράγωγος της F με την έννοια του Frechet ή η F- παράγωγος της F) στο σημείο x και θα γράφουμε $A_x = F'(x)$. Γενικά, λέμε ότι η $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι παραγωγίσιμη σε ένα σύνολο $D_0 \subset D$ αν $D_0 \subset \text{Int}(D)$ και αν η F είναι παραγωγίσιμη σε κάθε $x \in D_0$. Τότε, η F' μπορεί να θεωρηθεί ως απεικόνιση του D_0 στο σύνολο $\mathcal{L}(\mathbb{R}^n)$ των γραμμικών τελεστών απο τον $\mathbb{R}^n$ στον εαυτό του. Εύκολα επίσης μπορούμε να αποδείξουμε τη γραμμικότητα της πράξης της παραγώγισης: αν οι απεικονίσεις $F_1, F_2 : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι παραγωγίσιμες στο $x \in \text{Int}(D)$, τότε για $\alpha, \beta \in \mathbb{R}$ η $\alpha F_1 + \beta F_2$ είναι παραγωγίσιμη στο x και $(\alpha F_1 + \beta F_2)'(x) = \alpha F_1'(x) + \beta F_2'(x)$.

Αν η παράγωγος $F'(x)$ υπάρχει στο σημείο $x \in \text{Int}(D)$ τότε υπάρχουν όλες οι μερικοί παράγωγοι $\partial_j f_j = (\partial f_i / \partial x_j), 1 \le i, j \le n$ των συνιστωσών f_i της F στο σημείο x, ο δε $n \times n$ πίνακας που παριστάνει τον γραμμικό τελεστή $F'(x)$ ως προς την κανονική βάση $\{ e^j \}$, $1 \le j \le n$ του $\mathbb{R}^n$ ($e_i = \delta_{ij}$) είναι ο *Ιακωβιανός πίνακας* $J(x) : J_{ij}(x) = \partial_j f_i(x)$, $1 \le i, j \le n$. Πράγματι, θέτοντας, για $j = 1, \ldots, n$, $h = te^j$ στην (A.3) (με $\|\cdot\| = \|\cdot\|_2$) και υποθέτοντας ότι στο σημείο x η $F'(x) = A_x$ παριστάνεται ως προς την βάση $\{ e^j \}$ απο τον πίνακα (a_{ij}) έχουμε, για $1 \le i, j \le n$ ότι

$$\lim_{t \to 0} |(f_i(x + te^j) - f_i(x))/t - a_{ij}| = 0,$$

δηλαδή, ότι όντως $a_{ij} = \partial_j f_i(x) = J_{ij}(x)$. Η ύπαρξη μόνο των μερικών παραγώγων $\partial_j f_i(x)$ δεν εγγυάται όμως ότι η F είναι παραγωγίσιμη στο x. Αυτό φαίνεται αμέσως από την σημαντική συνέπεια της παραγωγισιμότητας που παρουσιάζεται στην επόμενη πρόταση:

Πρόταση Α.1. *Έστω ότι η $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι παραγωγίσιμη στο σημείο $x \in Int(D)$. Τότε η F είναι συνεχής στο x.*

Απόδειξη. Επειδή $x \in \text{Int}(D)$ υπάρχει $\delta_1 > 0$ τέτοιο ώστε $x + h \in D$, αν $\|h\| < \delta_1$. Η (A.3) συνεπάγεται τώρα ότι για δεδομένο $\varepsilon > 0$ υπάρχει $\delta > 0$, το οποίο μπορούμε να πάρουμε μικρότερο του δ_1, τέτοιο ώστε $\|F(x + h) - F(x) - F'(X)h\| \le \varepsilon \|h\|$, αν $\|h\| \le \delta$, από την οποία έπεται ότι

$$\|F(x + h) - F(x)\| \le (\|F'(x)\| + \varepsilon)\|h\|.$$

Σταθεροποιώντας το ε λοιπόν συμπεραίνουμε ότι για δεδομένο $x \in \text{Int}(D)$ υπάρχει $\delta > 0$ και $c \ge 0$ τέτοια ώστε $x + h \in D$ και $\|F(x+h) - F(x)\| \le c\|h\|$ αν $\|h\| \le \delta$, που είναι μάλιστα ένα συμπέρασμα ισχυρότερο απο την συνέχεια της F στο x. □

Το γνωστό θεώρημα μέσης τιμής για παραγωγίσιμες συναρτήσεις $f : \mathbb{R} \to \mathbb{R}$, δηλαδή, ότι για κάθε $x, y \in \mathbb{R}$ έχουμε $f(x) - f(y) = f'(z)(x - y)$, για κάποιο z μεταξύ των x και y, δεν ισχύει για απεικονίσεις $F : \mathbb{R}^n \to \mathbb{R}^n$ αν $n \ge 2$. Υπάρχουν όμως εναλλακτικά αποτελέσματα του τύπου μέσης τιμής, πολύ χρήσιμα στην ανάλυση. Παραδείγματος χάριν, πολλές φορές ενδιαφερόμαστε απλά να φράξουμε την ποσότητα $\|F(x) - F(y)\|$ συναρτήσει της F' :

Πρόταση Α.2. *Υποθέτουμε ότι η $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι παραγωγίσιμη σε ένα κυρτό σύνολο $D_0 \subset D$. Τότε, αν $x, y \in D_0$*

$$\|F(x) - F(y)\| \le \sup_{0 \le t \le 1} \|F'(x + t(y - x))\| \, \|x - y\|. \tag{A.4}$$

Απόδειξη. Εξ υποθέσεως, $x + t(y - x) \in D_0$ για $t \in [0, 1]$. Έστω τώρα ότι $M = \sup_{0 \le t \le 1} \|F'(x + t(y - x))\| < \infty$. Για δεδομένο $\epsilon > 0$, έστω Γ_ϵ το σύνολο των $t \in [0, 1]$ τέτοιων ώστε

$$\|F(x + t(y - x)) - F(x)\| \le Mt \, \|y - x\| + \epsilon t \, \|x - y\|. \tag{A.5}$$

Προφανώς, το Γ_ϵ δεν είναι κενό γιατί $0 \in \Gamma_\epsilon$. 'Εστω $\gamma_\epsilon = \sup_{t \in \Gamma_\epsilon} t$. Τότε $0 \le \gamma_\epsilon \le 1$ και επειδή λόγω της Πρότασης 1 η συνάρτηση $t \mapsto F(x + t(y - x))$ είναι συνεχής στο [0,1], παίρνοντας το όριο στην προηγούμενη σχέση μιας ακολουθίας $t_i \to \gamma_\epsilon$ σημείων του Γ_ϵ, έχουμε

$$\|F(x + \gamma_\epsilon(y - x)) - F(x)\| \le M\gamma_\epsilon \|y - x\| + \epsilon\gamma_\epsilon \|x - y\|. \tag{A.6}$$

Αν για κάθε $\epsilon > 0, \gamma_\epsilon = 1$, τότε η (Α.6) δίνει τη ζητούμενη ανισότητα. Αν για κάποιο $\epsilon > 0$ έχουμε $0 \leq \gamma_\epsilon < 1$, επειδή η F' υπάρχει στο σημείο $x + \gamma_\epsilon(y - x)$ έχουμε, απο τον ορισμό της παραγώγου με h ένα κατάλληλο (μικρό) πολλαπλάσιο του $y - x$, ότι υπάρχει $\beta_\epsilon \in (\gamma_\epsilon, 1)$ τέτοιο ώστε

$$\|F(x + \beta_\epsilon(y - x)) - F(x + \gamma_\epsilon(y - x)) - F'(x + \gamma_\epsilon(y - x))(\beta_\epsilon - \gamma_\epsilon)(y - x)\| \leq \epsilon(\beta_\epsilon - \gamma_\epsilon)\|y - x\|,$$

απο την οποία έπεται ότι

$$\|F(x + \beta_\epsilon(y - x)) - F(x + \gamma_\epsilon(y - x))\| \leq M(\beta_\epsilon - \gamma_\epsilon)\|y - x\| + \epsilon(\beta_\epsilon - \gamma_\epsilon)\|y - x\|. \quad \text{(A.7)}$$

Οι δύο τελευταίες σχέσεις δίνουν τότε μέσω της τριγωνικής ανισότητας ότι

$$\|F(x + \beta_\epsilon(y - x)) - F(x)\| \leq M\beta_\epsilon\|y - x\| + \epsilon\beta_\epsilon\|y - x\|,$$

δηλαδή ότι για αυτό το ϵ η (Α.5) ισχύει για κάποιο $\beta_\epsilon : \gamma_\epsilon < \beta_\epsilon < 1$, πράγμα που αντιφάσκει στον ορισμό του γ_ϵ. Συνεπώς, $\gamma_\epsilon = 1, \forall \epsilon > 0$ και το αποτέλεσμα προκύπτει όπως παραπάνω. □

Ενός άλλου τύπου αποτελέσματα είναι οι ολοκληρωτικές μορφές του θεωρήματος μέσης τιμής. Κατά τα γνωστά, αν η διανυσματική συνάρτηση μιας μεταβλητής $G : [a, b] \to \mathbb{R}^n$ έχει συνιστώσες $G = (g_1, ..., g_n)^T$, λέμε ότι είναι ολοκληρώσιμη (με την έννοια του Riemann) αν και μόνο αν για κάθε i οι συναρτήσεις $g_i[a, b] \to \mathbb{R}$ είναι ολοκληρώσιμες κατά Riemann στο $[a, b]$, και ορίζουμε

$$\int_a^b G(t)\,dt = \left(\int_a^b g_1(t)\,dt, \ldots, \int_a^b g_n(t)\,dt\right)^T.$$

Πρόταση Α.3. *Υποθέτουμε ότι η $F : D \subset R^n \to R^n$ είναι συνεχώς παραγωγίσιμη σε ένα κυρτό σύνολο $D_0 \subset D$. Τότε, αν $x, y \in D_0$*

$$F(y) - F(x) = \int_0^1 F'(x + t(y - x)) \cdot (y - x)\,dt. \quad \text{(A.8)}$$

Απόδειξη. Επειδή η F' είναι συνεχής στο D_0, τότε, για $x, y \in D_0$ η συνάρτηση $t \mapsto F'(x + t(y - x))$ είναι συνεχής στο $[0, 1]$. Συνεπώς, οι διανυσματικές συναρτήσεις $\nabla f_i(x+t(y-x)), 1 \leq i \leq n$ (γραμμές του Ιακωβιανού πίνακα που παριστάνει την $F'(x + t(y - x))$) είναι συνεχείς συναρτήσεις του t για $t \in [0, 1]$ και συνεπώς ολοκληρώσιμες κατά Riemann στο $[0, 1]$. Επειδή τώρα για $x, y \in D_0$

$$df_i(x + t(y - x))/dt = (\nabla f_i(x + t(y - x)))(y - x),$$

έχουμε, ολοκληρώνοντας ως προς t από 0 έως 1, ότι για $1 \leq i \leq n$

$$f_i(y) - f_i(x) = \int_0^1 \nabla f_i(x + t(y - x))^\intercal (y - x)\,dt,$$

που είναι ακριβώς η (Α.8) γραμμένη κατά συνιστώσες. □

Λήμμα Α.1. *Έστω ότι η $G : [a, b] \to \mathbb{R}^n$ είναι συνεχής στο $[a.b]$. Τότε*

$$\|\int_a^b G(t)\,dt\| \leq \int_a^b \|G(t)\|\,dt. \quad \text{(A.9)}$$

Απόδειξη. Η συνάρτηση $t \mapsto \|G(t)\|$ είναι συνεχής στο $[a, b]$ λόγω της υπόθεσης μας και της συνέχειας της $x \mapsto \|x\|$. Συνεπώς, η $t \mapsto \|G(t)\|$ είναι ολοκληρώσιμη κατά Riemann στο $[a, b]$. Λόγω της ολοκληρωσιμότητας της $G(t)$ έχουμε ότι για κάθε $\epsilon > 0$ υπάρχει διαμερισμός $a \leq t_0 < t_1 < ... < t_s \leq b$ τέτοιος ώστε

$$\|\int_a^b G(t)\,dt - \sum_{j=1}^{s} G(t_j) \cdot (t_j - t_{j-1})\| \leq \epsilon,$$

και

$$\left| \int_a^b \|G(t)\| \, dt - \sum_{j=1}^{s} \|G(t_j)\| \cdot (t_j - t_{j-1}) \right| \leq \epsilon.$$

Συνεπώς, από την τριγωνική ανισότητα και τις δυο αυτές σχέσεις έπεται ότι

$$\begin{aligned} \left\| \int_a^b G(t) \, dt \right\| &\leq \left\| \sum_{j=1}^{s} G(t_j) \cdot (t_j - t_{j-1}) \right\| + \epsilon \\ &\leq \sum_{j=1}^{s} \|G(t_j)\| \cdot (t_j - t_{j-1}) + \epsilon \\ &\leq \int_a^b ||G(t)|| \, dt + 2\epsilon. \end{aligned}$$

Επειδή το $\epsilon > 0$ ήταν αυθαίρετο, αυτό ολοκληρώνει την απόδειξη του λήμματος. □

Πρόταση Α.4. *Υποθέτουμε ότι η $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ είναι συνεχώς παραγωγίσιμη σε ένα κυρτό σύνολο $D_0 \subset D$ και ότι επιπλέον υπάρχουν σταθερές $a, p \geq 0$ τέτοιες ώστε*

$$\|F'(u) - F'(v)\| \leq a\|u - v\|^p, \quad u, v \in D_0. \tag{A.10}$$

Τότε, για κάθε $x, y \in D_0$

$$\|F(y) - F(x) - F'(x)(y - x)\| \leq a\|x - y\|^{p+1}/(p + 1). \tag{A.11}$$

Απόδειξη. Η F ικανοποιεί τις υποθέσεις της Πρότασης 3. Συνεπώς, για $x, y \in D_0$

$$F(y) - F(x) = \int_0^1 F'(x + t(y - x))(y - x) \, dt.$$

Επομένως, λόγω της (Α.9) και της (Α.10) έχουμε

$$\begin{aligned} \|F(y) - F(x) - F'(x)(y - x)\| &= \left\| \int_0^1 [F'(x + t(y - x)) - F'(x)](y - x) \, dt \right\| \\ &\leq \|x - y\| \int_0^1 \|F'(x + t(y - x)) - F'(x)\| \, dt \\ &\leq a||x - y||^{p+1} \int_0^1 t^p \, dt \\ &= a||x - y||^{p+1}/(p + 1), \end{aligned}$$

το οποίο ολοκληρώνει την απόδειξη της πρότασης. □