



RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων
Δεδομένων Παρατήρησης Γης

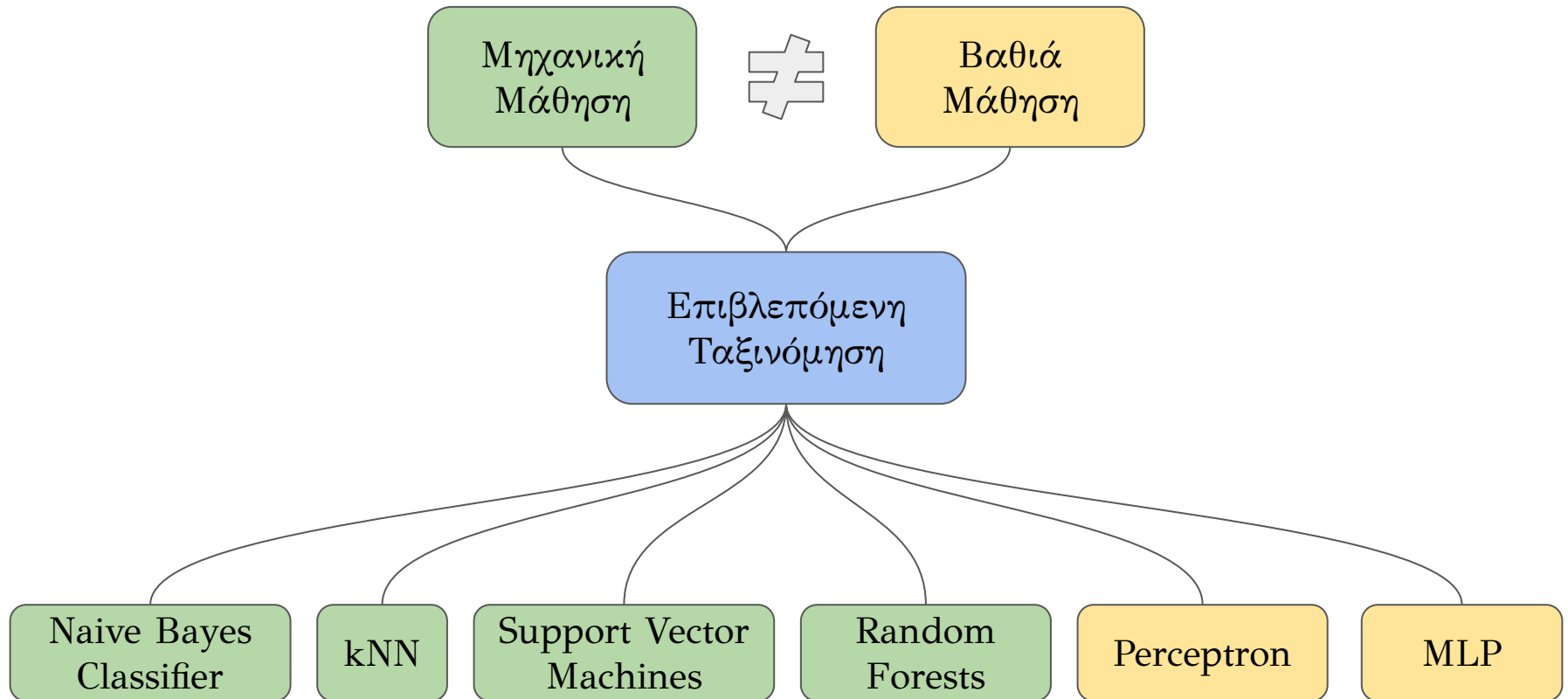
Εισαγωγή στη Βαθιά Μάθηση

Μέρος II

Αθηνά Ψάλτα
Βασίλειος Τσιρώνης
Κωνσταντίνος Καράντζαλος

Εαρινό εξάμηνο 2022

Ανακεφαλαίωση



Περιεχόμενα

1. Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

- a. Συναρτήσεις Κόστους
- b. Μέθοδος Καταβιβασμού Κλίσης
- c. Υπερπαράμετροι
- d. Μέθοδοι βελτιστοποίησης

2. Τεχνικές Regularization

3. Αξιολόγηση

- a. Γενικές μετρικές αξιολόγησης
- b. Μετρικές προσανατολισμένες σε συγκεκριμένες εφαρμογές

Εκπαίδευση ΝΔ

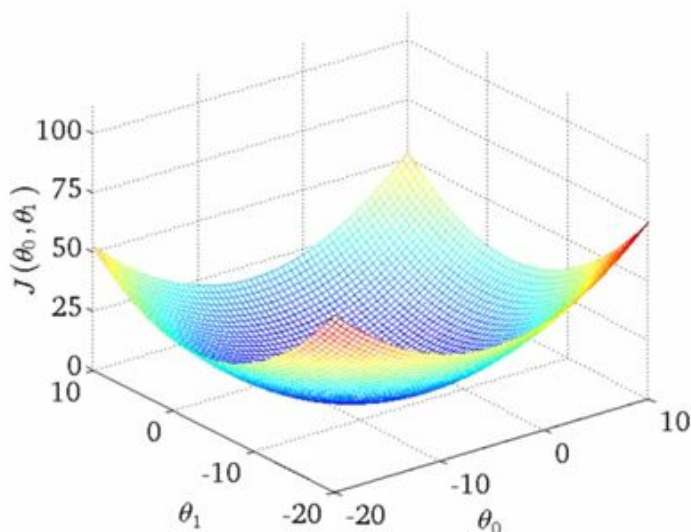
Βελτιστοποίηση κόστους και Μέθοδος
Καταβιβασμού της Κλίσης

Βελτιστοποίηση του κόστους

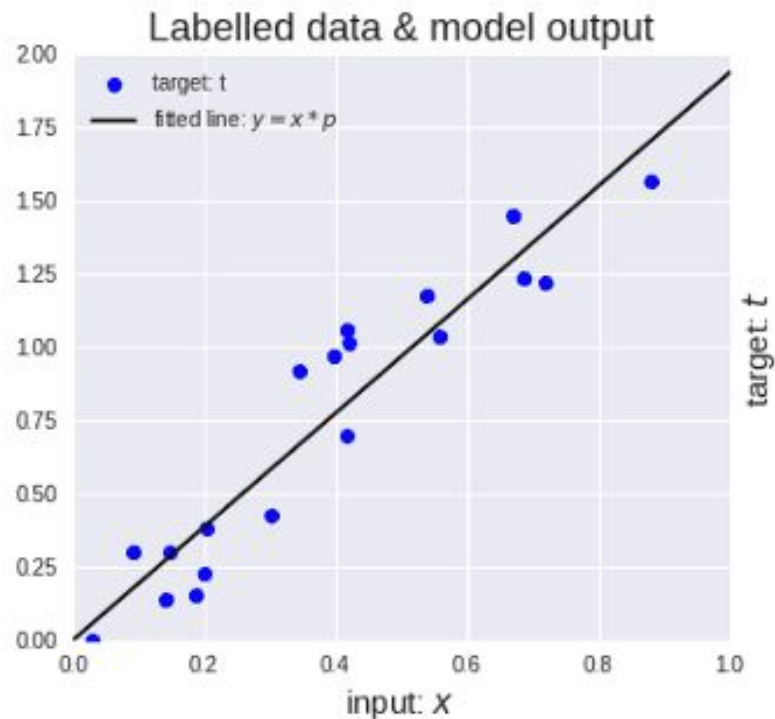
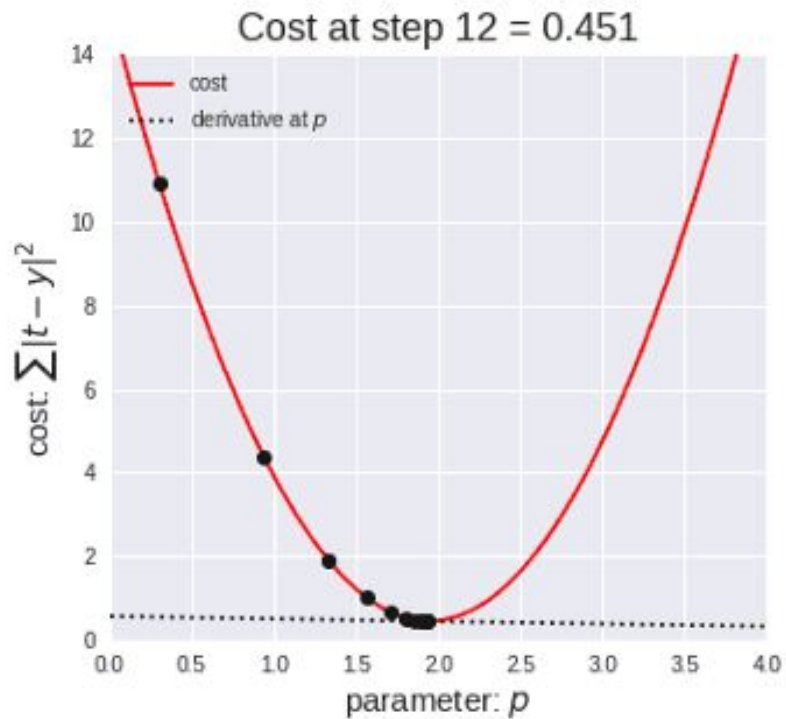
Στόχος είναι να βρούμε εκείνα τα βάρη (weights) του νευρωνικού δικτύου που επιτυγχάνουν το χαμηλότερο κόστος :

$$J(w) = \underset{w}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x^{(i)}; w), y^{(i)})$$

δηλαδή επιθυμούμε να βρούμε το **ολικό ελάχιστο** ή (κάποιο) **τοπικό ελάχιστο** της συνάρτησης κόστους.



Βελτιστοποίηση του κόστους



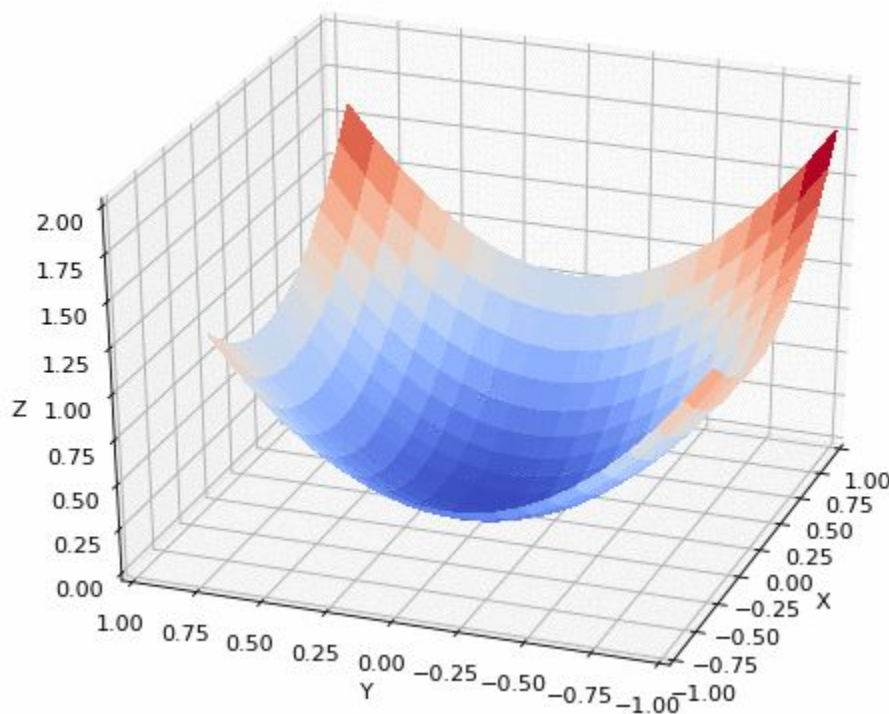
Βελτιστοποίηση vs Μηχανική Μάθηση

Θεωρούμε σε κάθε περίπτωση πως τα δεδομένα μας προέρχονται από μία (κοινή) **γεννήτρια συνάρτηση κατανομής** p_{data} καθώς επίσης υποθέτουμε την **στατιστική ανεξαρτησία των δειγμάτων**. Ακόμη θεωρούμε, ότι τα σετ εκπαίδευσης και αξιολόγησης στα οποία χωρίσαμε τα δεδομένα, **κατανέμονται ομότροπα**.

- **Ίδια μεθοδολογία επίλυσης**
 - Μέθοδος καταβιβασμού κλίσης
 - Gauss-Newton
 - Levenberg Marquardt
 - ...
- **Διαφορετικός στόχος !**
 - **Βελτιστοποίηση** → Ελαχιστοποίηση σφάλματος στο σετ εκπαίδευσης
 - Τυπικά δεν ορίζεται σετ αξιολόγησης/ ελέγχου
 - **Μηχανική Μάθηση** → Ελαχιστοποίηση σφάλματος στο σετ εκπαίδευσης και ταυτόχρονη μείωση της διαφοράς στο σφάλμα μεταξύ του σετ εκπαίδευσης και του σετ ελέγχου!

Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

- Πρώτης τάξεως αλγόριθμος βελτιστοποίησης
 - Λαμβάνει υπόψη μόνο την πρώτη τάξεως παράγωγο για να ανανεώσει τις παραμέτρους
- Εύρεση της διεύθυνσης της μεγαλύτερης κλίσης (gradient)



Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

1. Τυχαία (;) αρχικοποίηση των βαρών \mathbf{W} , \mathbf{b}
2. Επιλογή του ρυθμού εκμάθησης (learning rate) α
3. Μέχρις ότου να συγκλίνει η συνάρτηση, σε κάθε επανάληψη υπολογίζονται οι μερικές παράγωγοι της συνάρτησης κόστους $J(\mathbf{w})$ ως προς τα \mathbf{W} , \mathbf{b} :

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \nabla_{\mathbf{w}} J$$

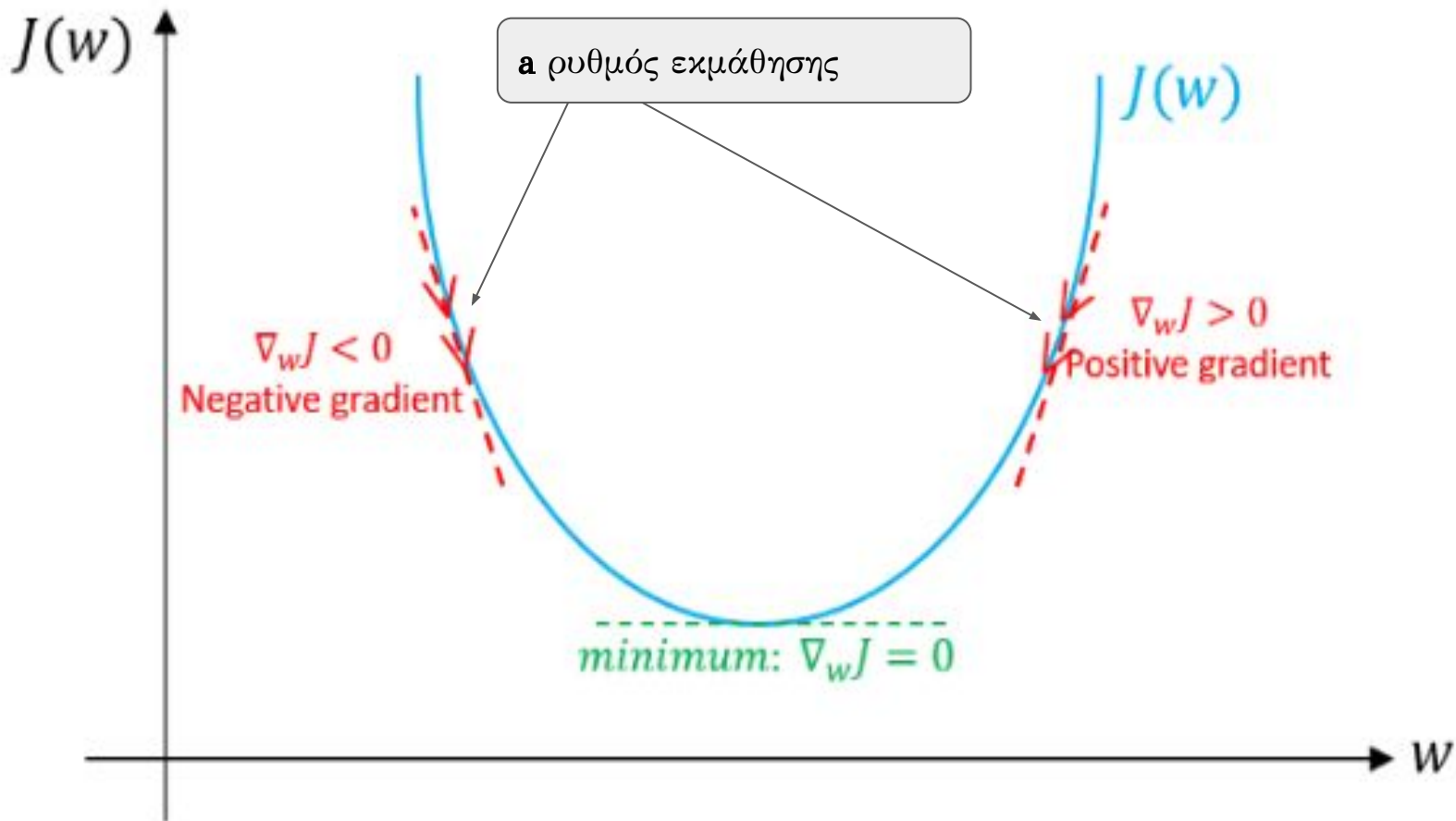
$$\frac{\partial}{\partial \mathbf{b}} J(\mathbf{w}) = \nabla_{\mathbf{b}} J$$

και αντίστοιχα ανανεώνονται τα \mathbf{W} , \mathbf{b} ως εξής :

$$\mathbf{w} = \mathbf{w} - \alpha \nabla_{\mathbf{w}} J$$

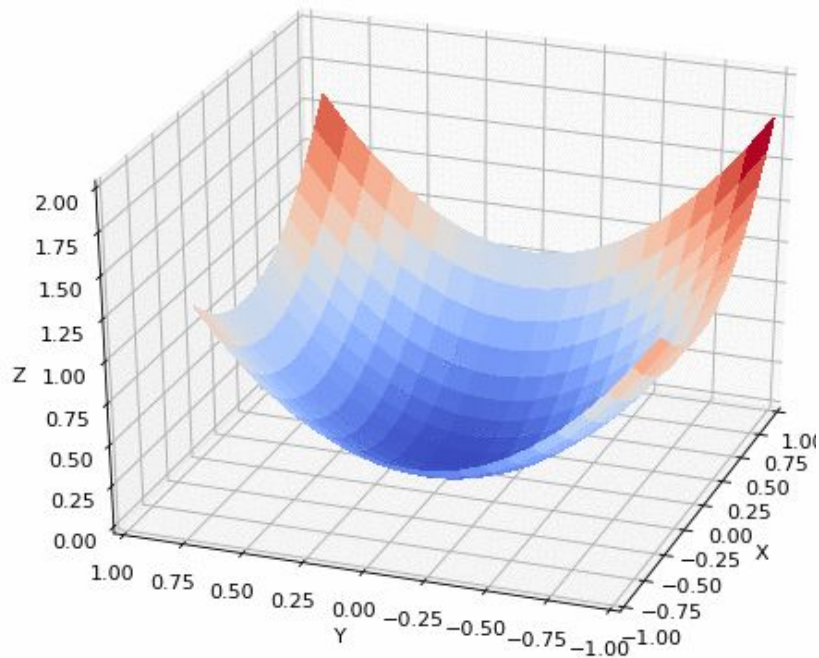
$$\mathbf{b} = \mathbf{b} - \alpha \nabla_{\mathbf{b}} J$$

Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

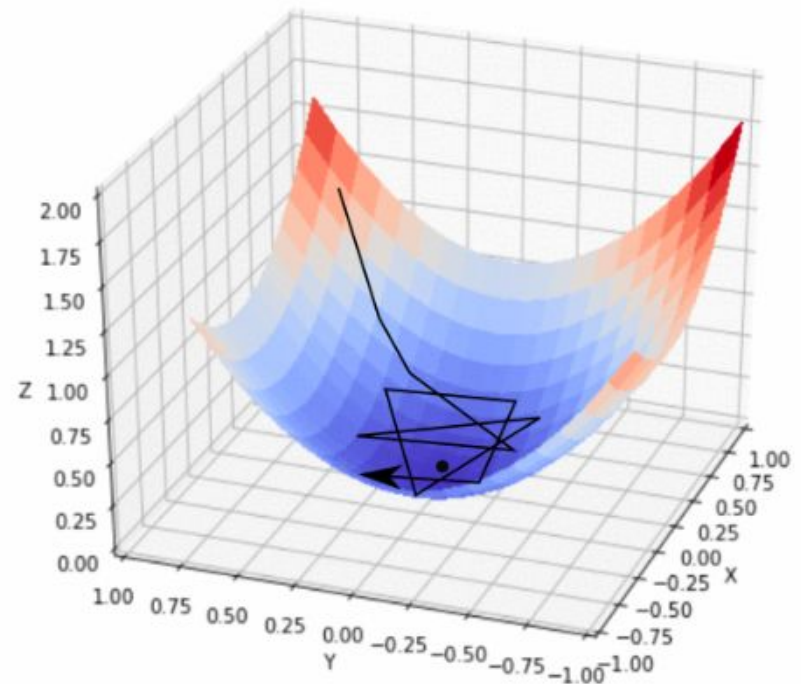


Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

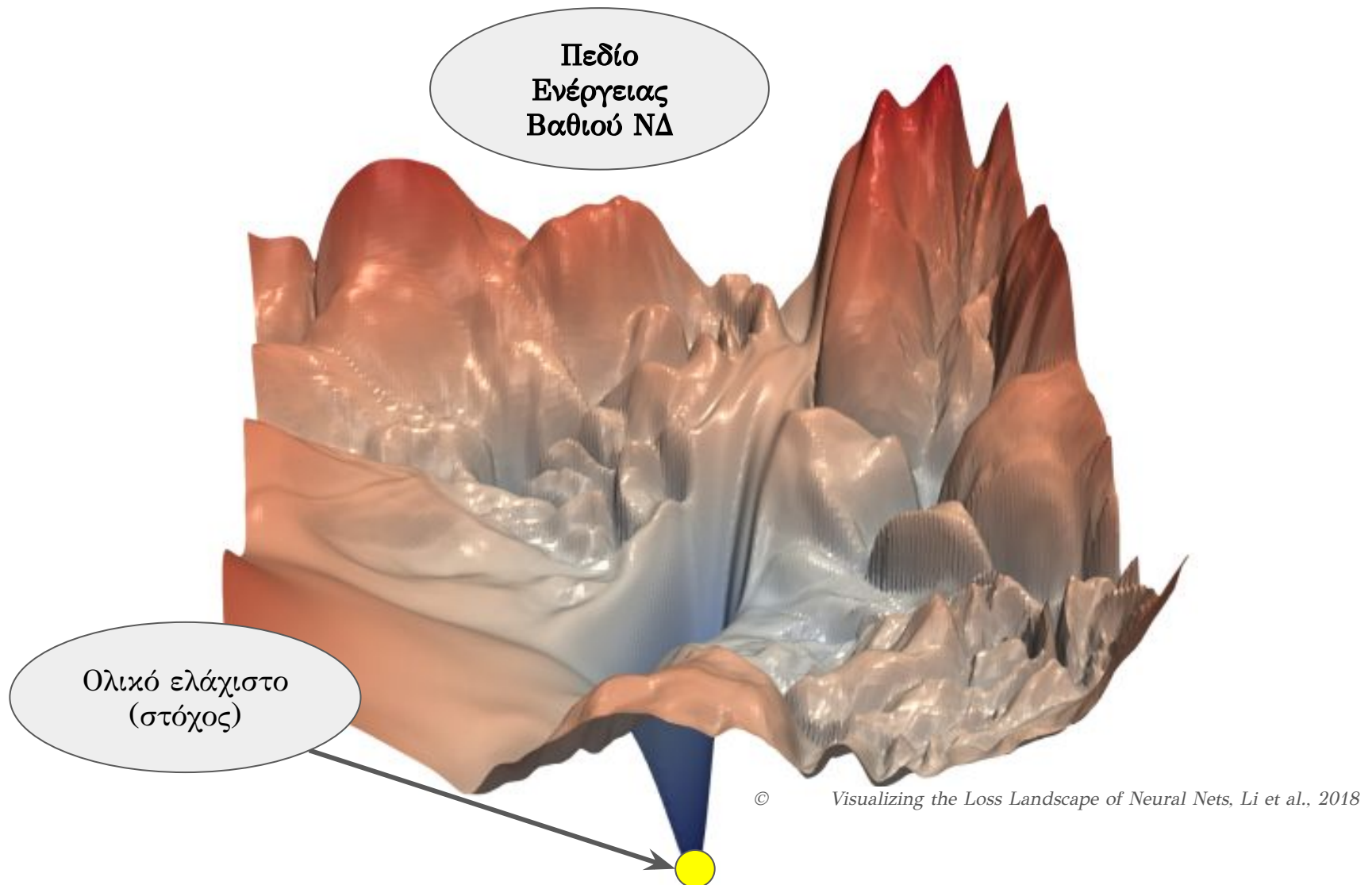
Θεωρία



Πράξη



Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

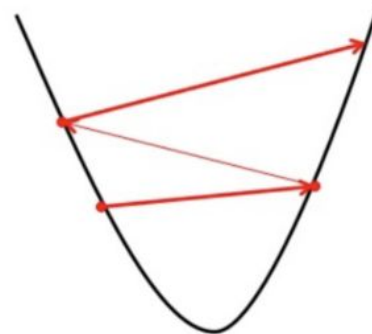


Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

Ο ρυθμός εκμάθησης (learning rate) καθορίζει το “μέγεθος” του βήματος σε κάθε επανάληψη (step)

- Η διόρθωση σε κάθε επανάληψη έχει **νόρμα** ανάλογη του ρυθμού εκμάθησης
- Η κατάλληλη επιλογή του ρυθμού αποτελεί μία υπερπαράμετρο
 - Τυπική τάξη μεγέθους: 10^{-3} - 10^{-4}

Μεγάλος ρυθμός εκμάθησης

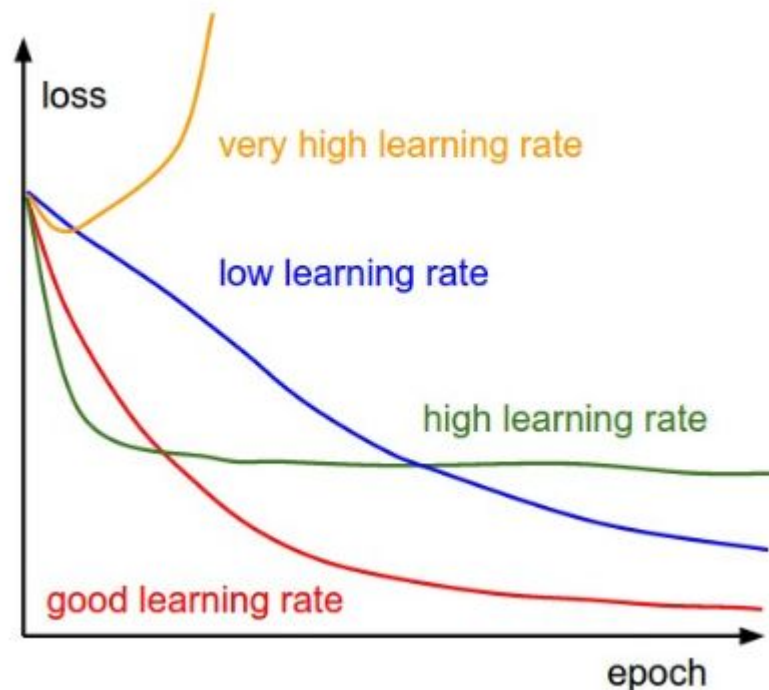


Μικρός ρυθμός εκμάθησης



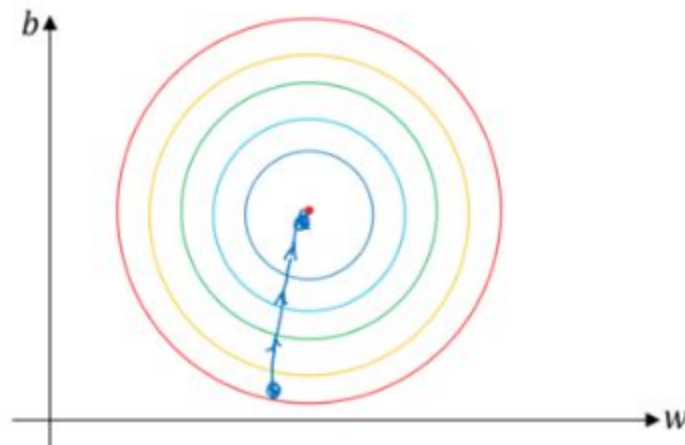
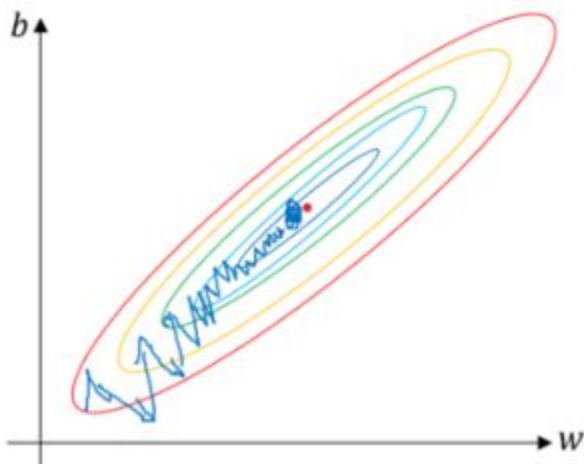
Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

- Ως **εποχή** (epoch) ορίζεται ένας κύκλος εκπαίδευσης σε όλο το σύνολο των δεδομένων εκπαίδευσης.
 - Ένα προς τα εμπρός “πέρασμα” (forward-pass)
 - Ένα προς τα πίσω “πέρασμα” (backward pass)
- Η επιλογή του ρυθμού εκμάθησης επηρεάζει σημαντικά την ποιότητα της τελικής εκτίμησης
- Η βέλτιστη επιλογή δεν είναι προφανής!



Μέθοδος Καταβιβασμού Κλίσης (Gradient Descent)

- **Κανονικοποίηση** των δεδομένων σε περίπτωση που έχουν διαφορετική κλίμακα
- Αν δεν τα κανονικοποιήσουμε τότε οι παρακάτω ισοϋψείς θα είναι στενότερες άρα ο **χρόνος σύγκλισης** θα είναι σημαντικά μεγαλύτερος!



Back-Propagation

- Μέθοδος υπολογισμού των μερικών παραγώγων της συνάρτησης κόστους ως προς τα βάρη του δικτύου.
- Το back-propagation βασίζεται στον κανόνα της αλυσίδας:

Για το μονοδιάστατο πρόβλημα $z = f(g(x)) = f(y)$ ως γνωστόν ισχύει

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx}.$$

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i}.$$

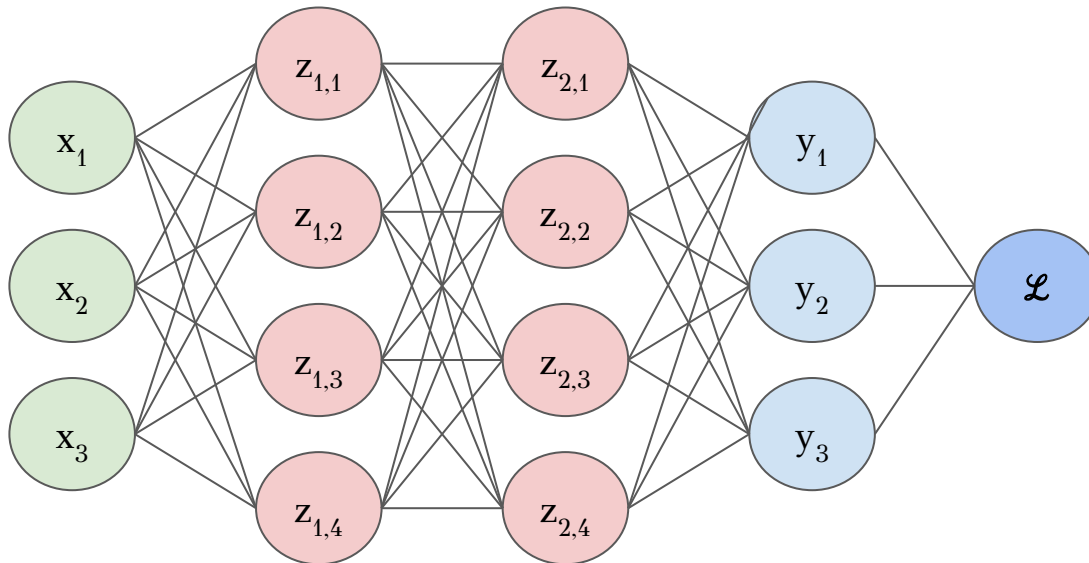
Πολυδιάστατο πρόβλημα

$$\nabla_{\mathbf{x}} z = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^\top \nabla_{\mathbf{y}} z.$$

Διανυσματική μορφή

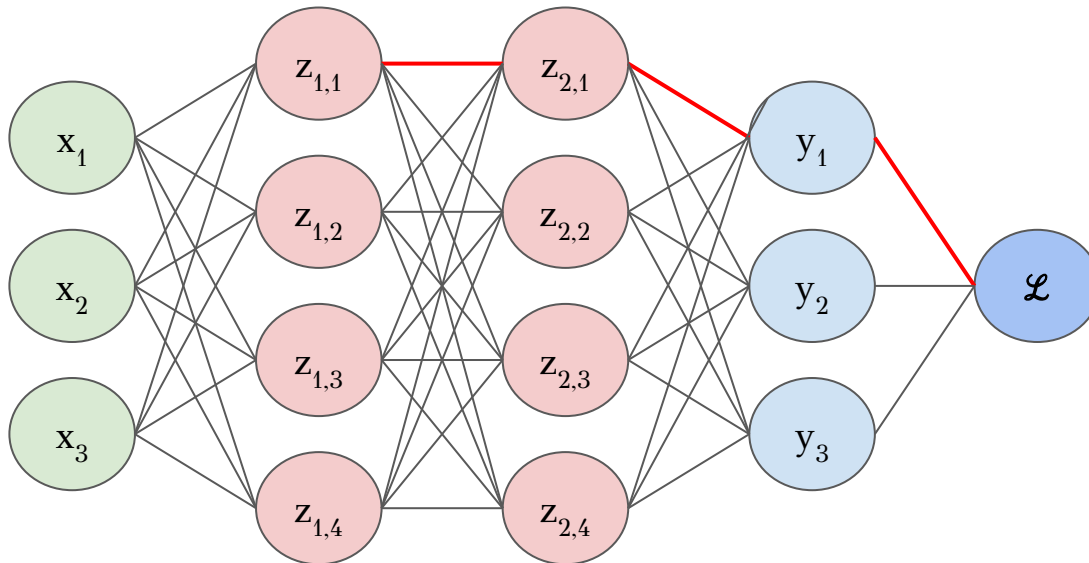
Back-Propagation

Παράδειγμα σε ένα απλό MLP



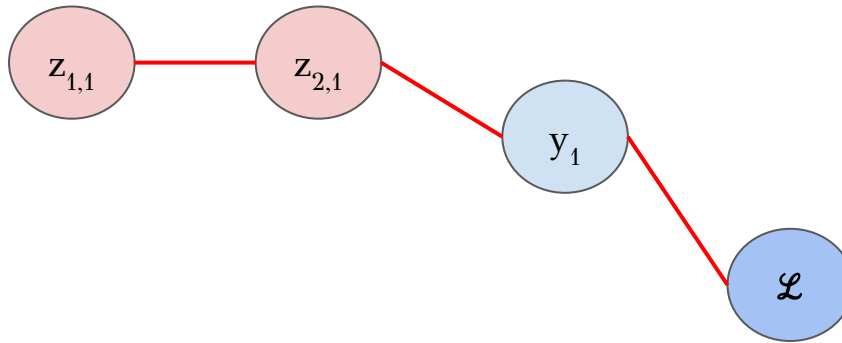
Back-Propagation

Παράδειγμα σε ένα απλό MLP



Back-Propagation

Απλοποιημένο πρόβλημα

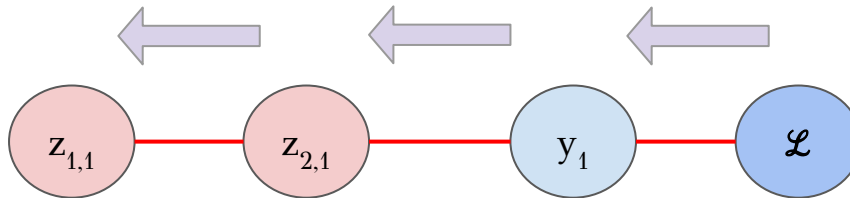


$$\mathcal{L}(y_1(z_{2,1}(z_{1,1})))$$

Back-Propagation

$$\mathcal{L}(y_1(z_{2,1}(z_{1,1})))$$

Κανόνας αλυσίδας σε κατευθυντικό γράφημα



$$\frac{\partial \mathcal{L}}{\partial z_{1,1}}$$

$$\frac{\partial \mathcal{L}}{\partial z_{2,1}}$$

$$\frac{\partial \mathcal{L}}{\partial y_1}$$

Επίπεδο 1

$$\frac{\partial \mathcal{L}}{\partial z_{2,1}} \frac{\partial z_{2,1}}{\partial z_{1,1}}$$

$$\frac{\partial \mathcal{L}}{\partial y_1} \frac{\partial y_1}{\partial z_{2,1}}$$

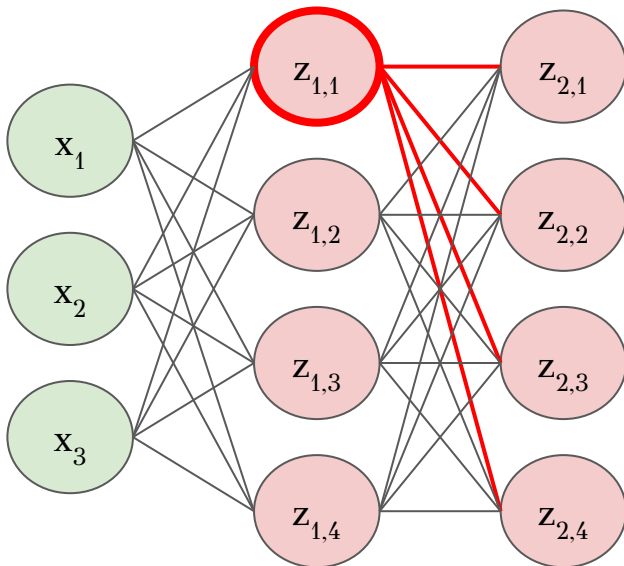
Επίπεδο 2
1 x κανόνα αλυσίδας

$$\frac{\partial \mathcal{L}}{\partial y_1} \frac{\partial y_1}{\partial z_{2,1}} \frac{\partial z_{2,1}}{\partial z_{1,1}}$$

Επίπεδο 3
2 x κανόνα αλυσίδας

Back-Propagation

Παράδειγμα σε ένα απλό MLP



$$\frac{\partial \mathcal{L}}{\partial z_{2,1}}$$

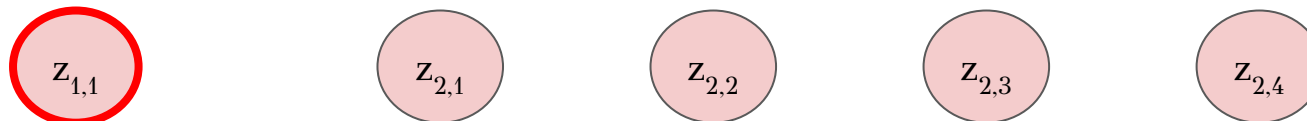
$$\frac{\partial \mathcal{L}}{\partial z_{2,2}}$$

$$\frac{\partial \mathcal{L}}{\partial z_{2,3}}$$

$$\frac{\partial \mathcal{L}}{\partial z_{2,4}}$$

Back-Propagation

Παράδειγμα σε ένα απλό MLP

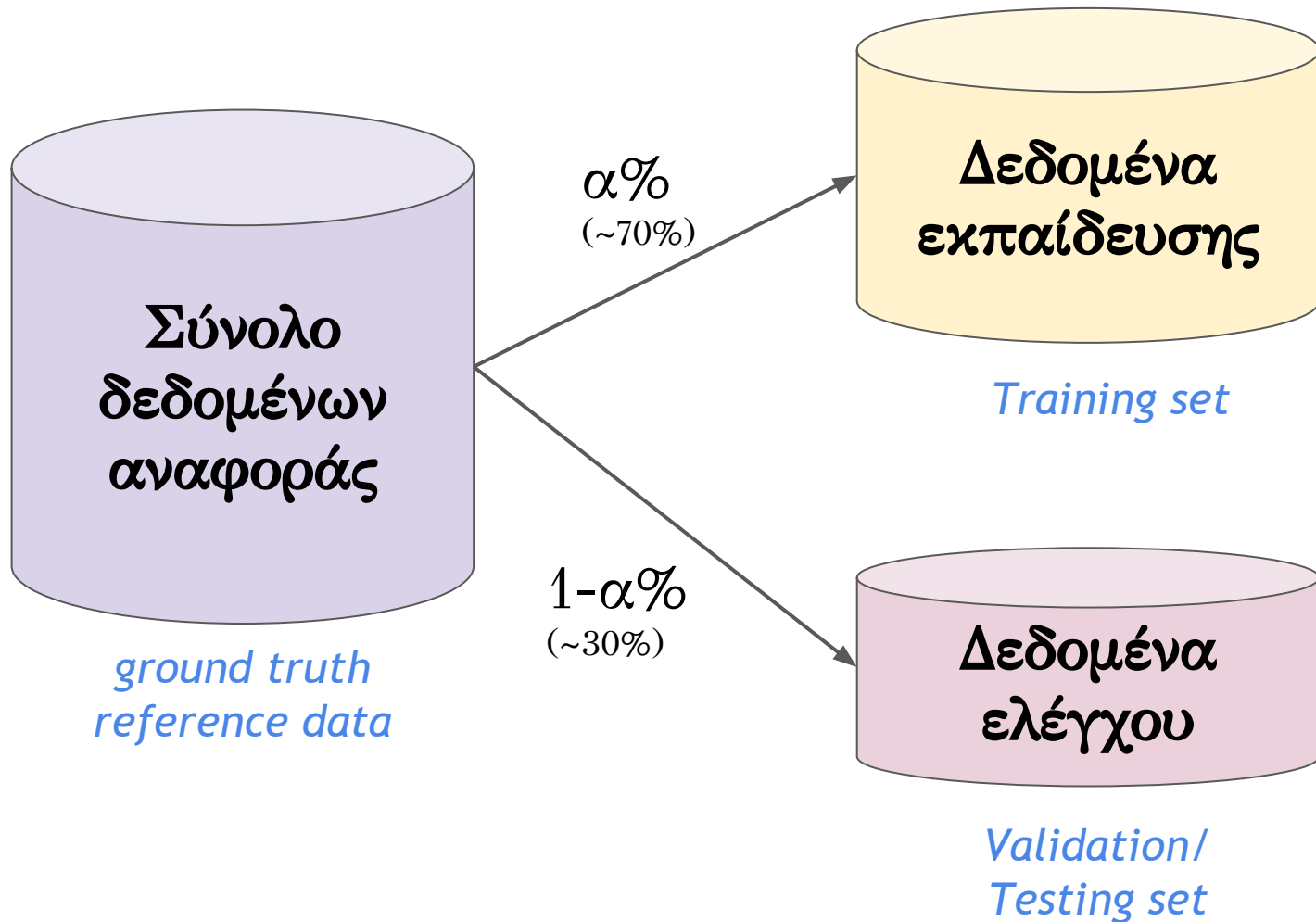


The diagram shows a simple MLP with two layers. The first layer has nodes $z_{1,1}$, $z_{2,1}$, $z_{2,2}$, $z_{2,3}$, and $z_{2,4}$. The node $z_{1,1}$ is highlighted with a red circle.

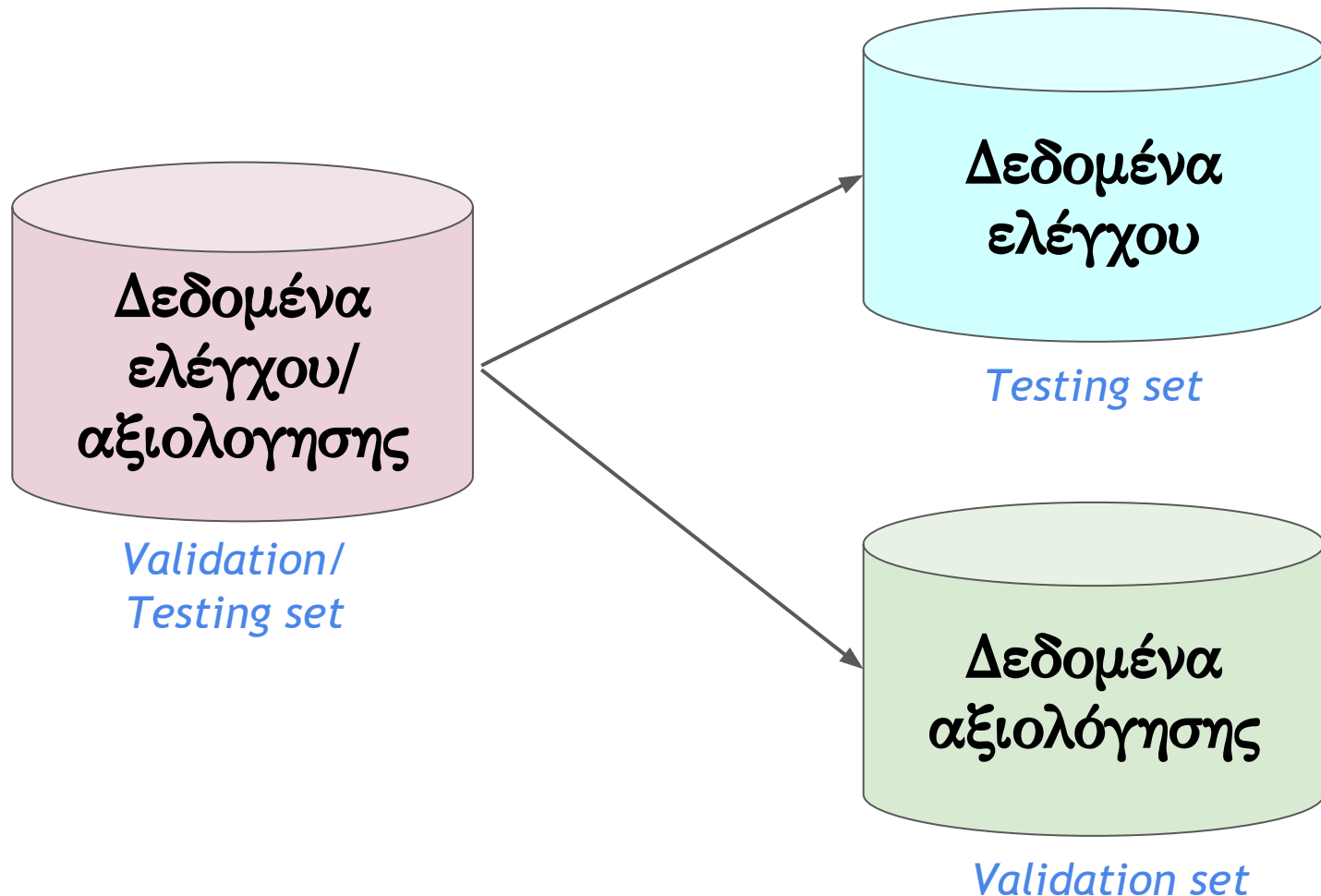
$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial z_{1,1}} &= \frac{\partial \mathcal{L}}{\partial z_{2,1}} \frac{\partial z_{2,1}}{\partial z_{1,1}} + \frac{\partial \mathcal{L}}{\partial z_{2,2}} \frac{\partial z_{2,2}}{\partial z_{1,1}} + \frac{\partial \mathcal{L}}{\partial z_{2,3}} \frac{\partial z_{2,3}}{\partial z_{1,1}} + \frac{\partial \mathcal{L}}{\partial z_{2,4}} \frac{\partial z_{2,4}}{\partial z_{1,1}} \\ &= \sum_{i=1}^4 \frac{\partial \mathcal{L}}{\partial z_{2,i}} \frac{\partial z_{2,i}}{\partial z_{1,1}} \\ &= \frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1} * \nabla \mathcal{L}_{\mathbf{z}_2}\end{aligned}$$

Υπερπαράμετροι & Βελτιστοποίηση

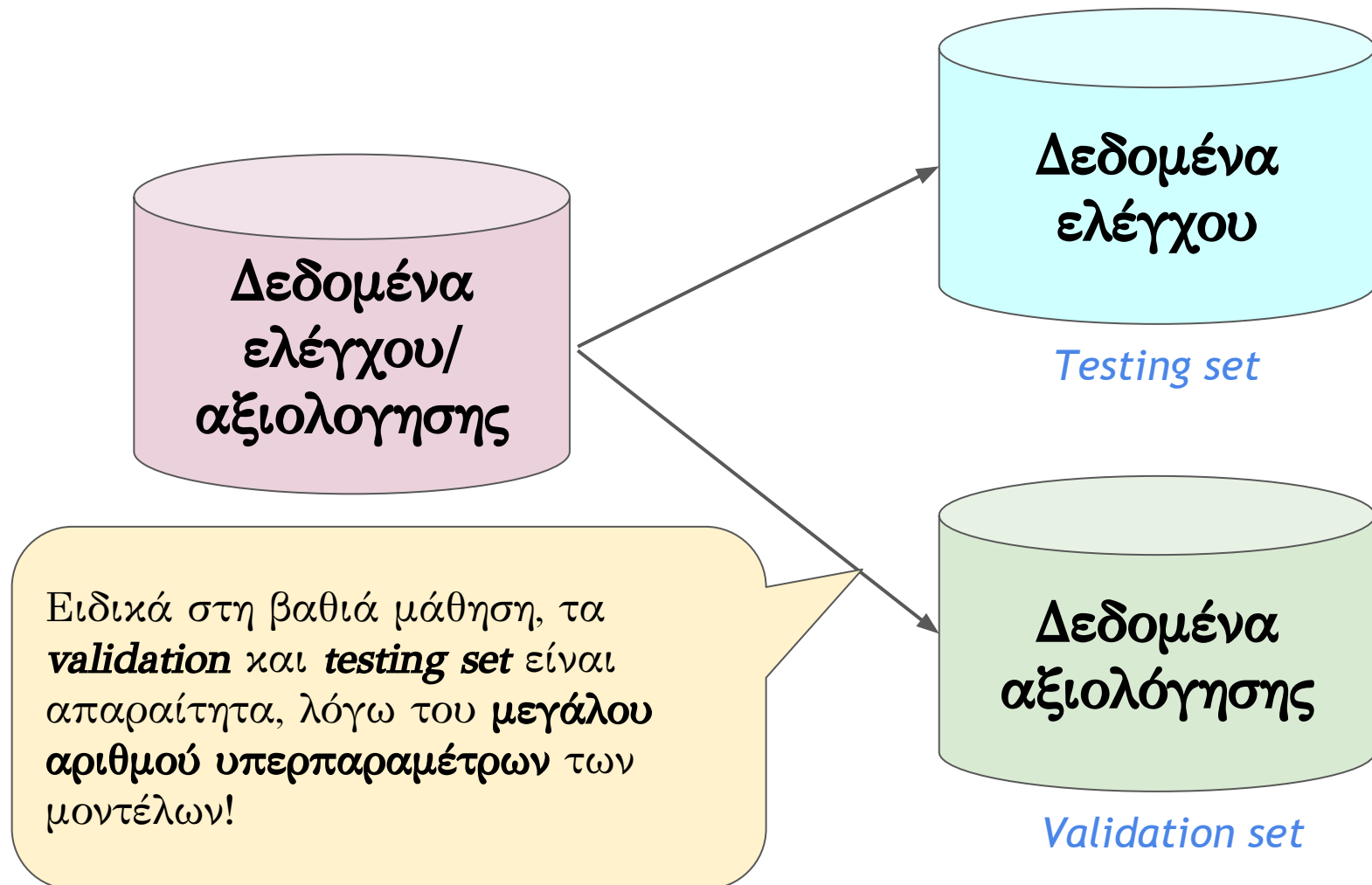
Επιβλεπόμενη Μάθηση (Supervised Learning)



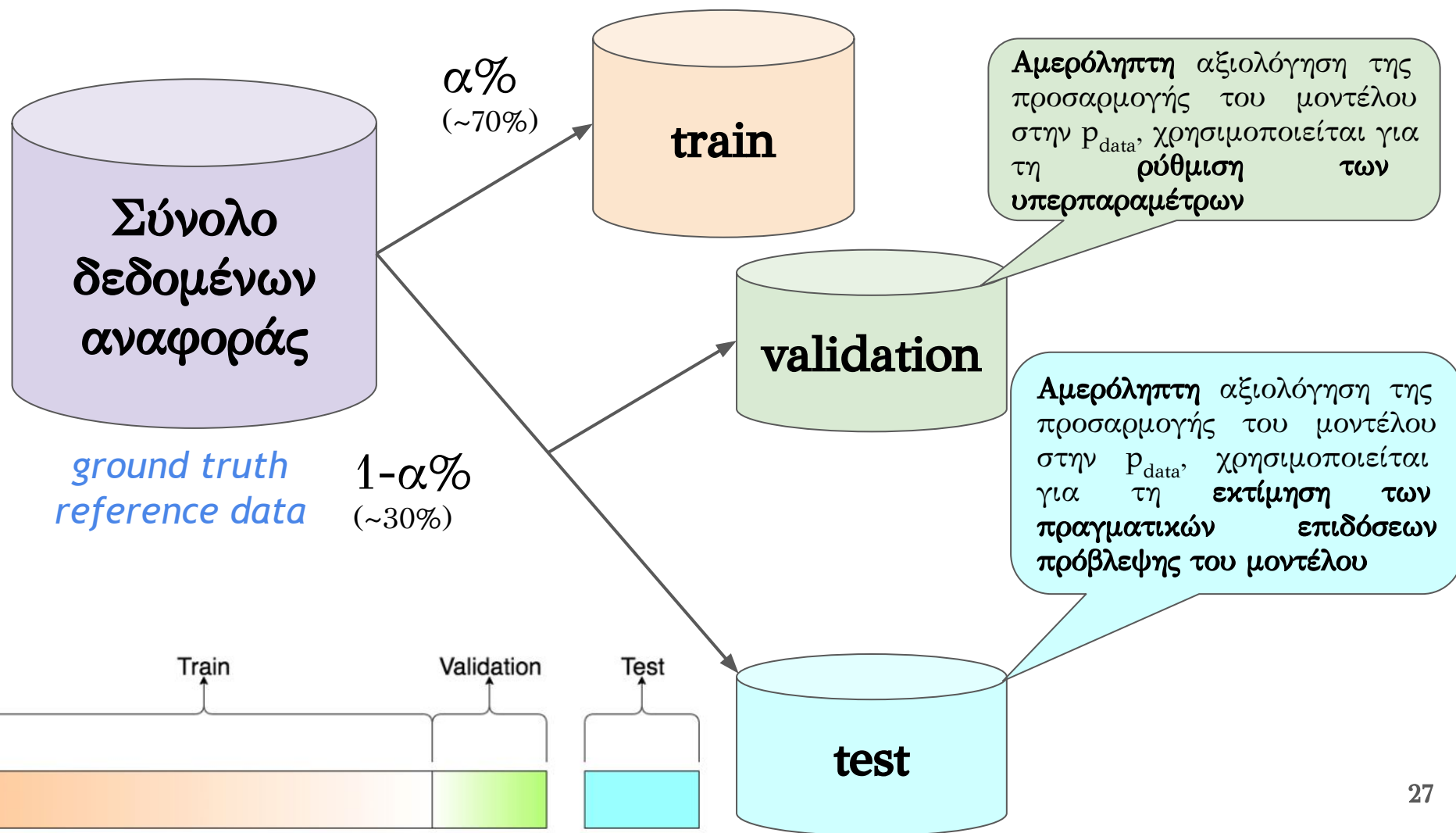
Επιβλεπόμενη Μάθηση (Supervised Learning)



Επιβλεπόμενη Μάθηση (Supervised Learning)



Επιβλεπόμενη Μάθηση (Supervised Learning)



Υπερπαράμετροι

Στη μηχανική μάθηση, και κατα συνέπεια στη Βαθιά Μάθηση, κάθε παράμετρος που επηρεάζει τις επιδόσεις του μοντέλου στο σετ ελέγχου, αλλά δε συμπεριλαμβάνεται στις “προς-εκμάθηση” (trainable) παραμέτρους χαρακτηρίζεται ως **υπερπαράμετρος**.

Παραδείγματα:

- Ρυθμός εκμάθησης, Σύνολο εποχών εκπαίδευσης. Μέγεθος batch.
- Πλήθος κρυφών στρώσεων (αρχιτεκτονική).
- ...

Οι υπερπαράμετροι τυπικά ρυθμίζονται στο **validation set**, συνήθως με κάποια από τις επόμενες μεθόδους:

1. Χειροκίνητη αναζήτηση
2. Αναζήτηση Πλέγματος (Grid Search)
3. Τυχαία αναζήτηση (Random Search)
4. Βελτιστοποίηση κατά Bayes (Bayesian optimization)

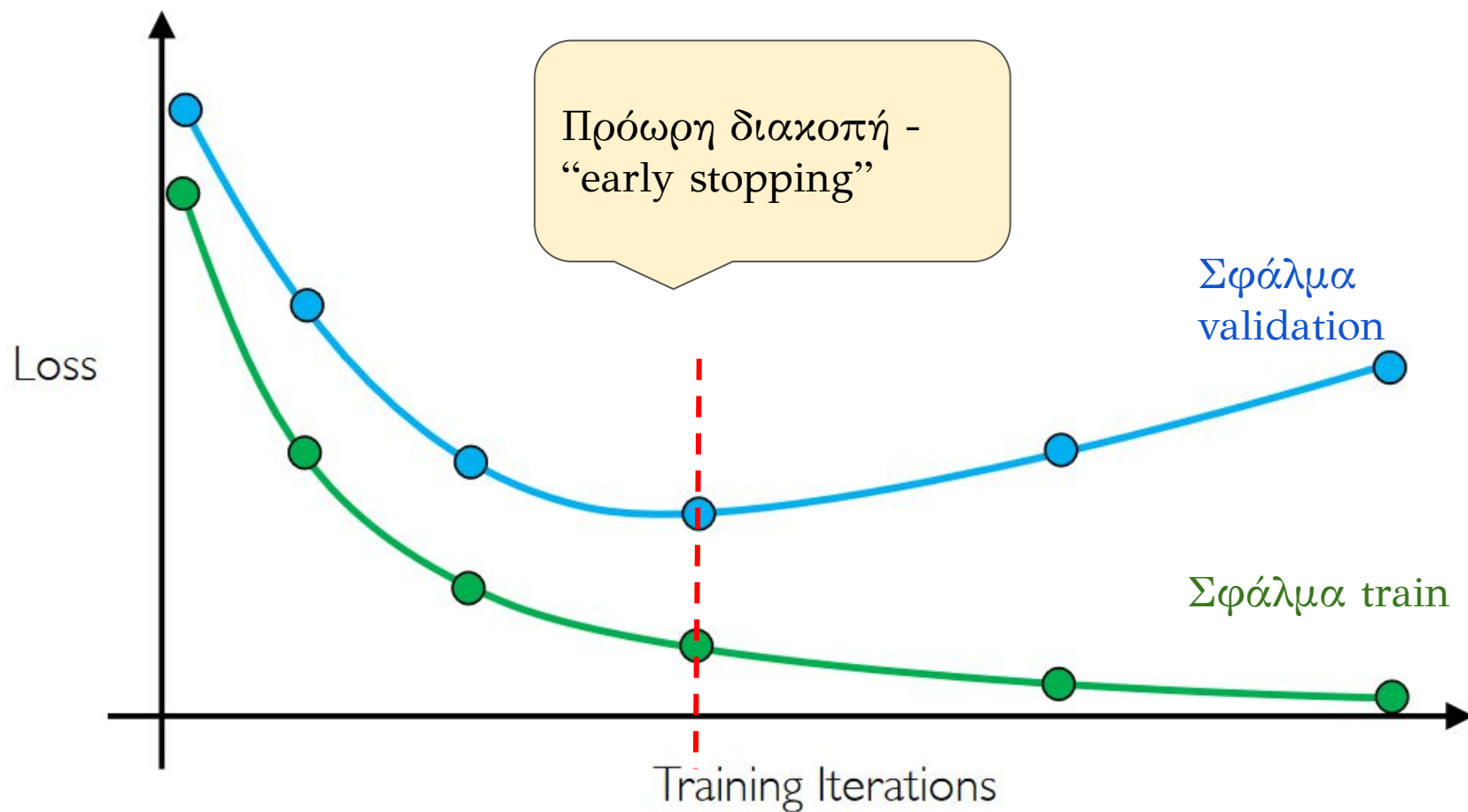
Υπερπαράμετροι

Παράδειγμα: Το πλήθος επαναλήψεων/εποχών ως υπερπαράμετρος



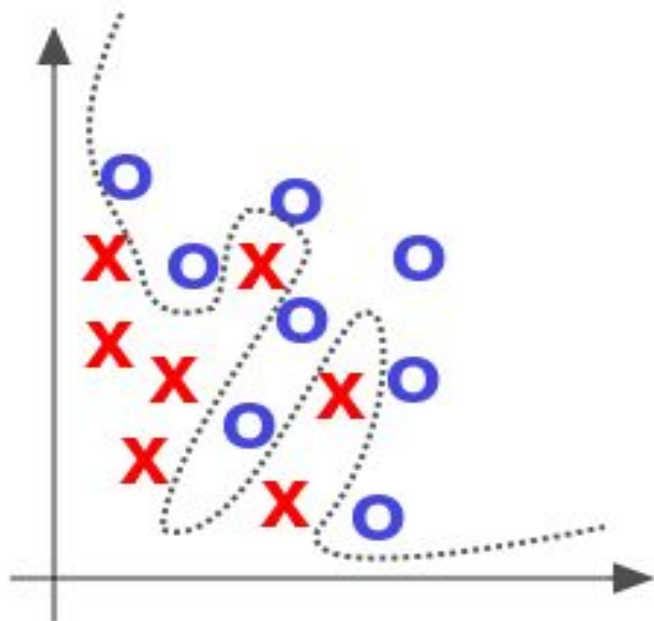
Υπερπαράμετροι

Παράδειγμα: Το πλήθος επαναλήψεων/εποχών ως υπερπαράμετρος



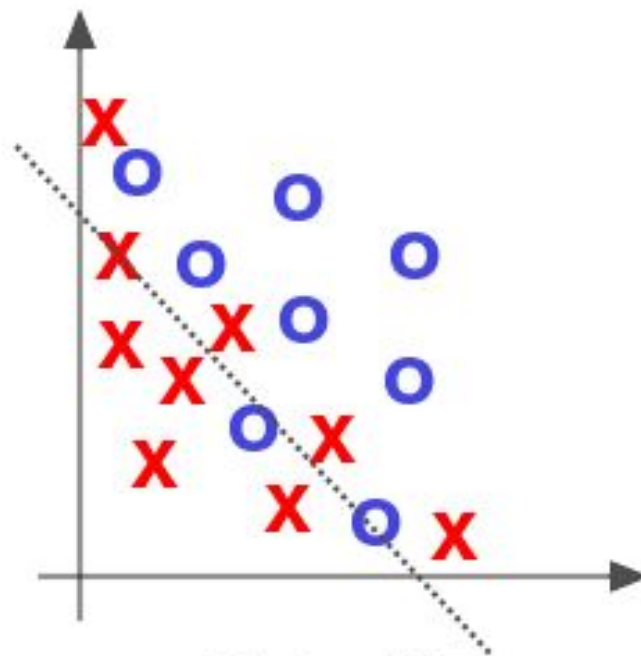
Overfit vs Underfit

Overfit: Το μοντέλο δεν προσαρμόζεται μόνο στην p_{data} αλλά προσαρμόζεται και στον θόρυβο του σετ δεδομένων εκπαίδευσης



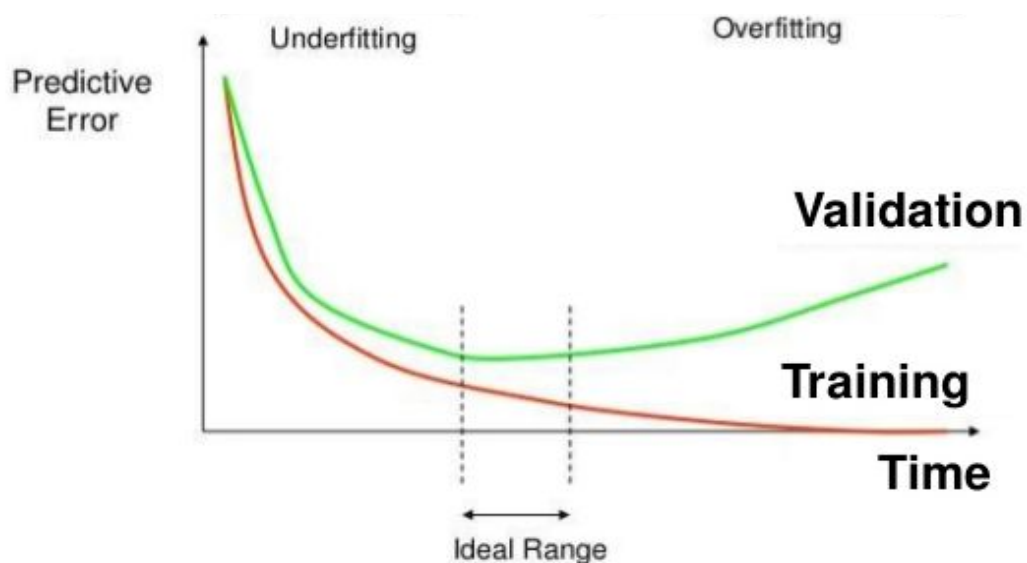
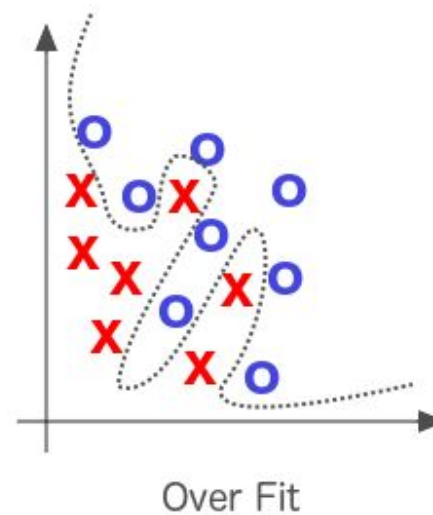
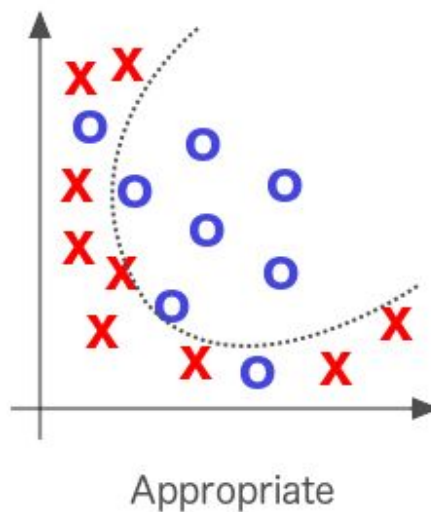
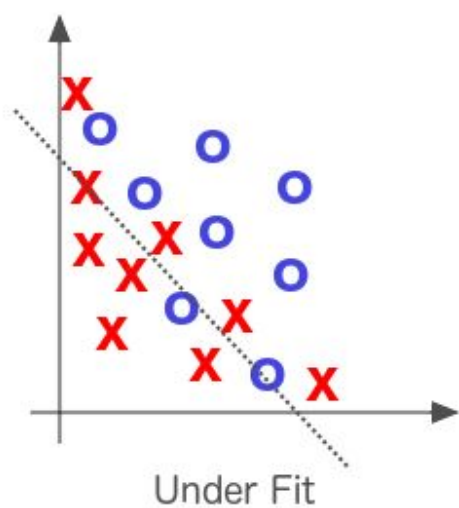
Over Fit

Underfit: Το μοντέλο δεν προσαρμόζεται στην p_{data} ικανοποιητικά



Under Fit

Overfit vs Underfit



Εκπαίδευση σε batches (τμήματα)

Τις περισσότερες φορές **δεν είναι τεχνικά δυνατό** να εφαρμόσουμε τη μέθοδο καταβιβασμού κλίσης απευθείας στα δεδομένα εκπαίδευσης.

Γιατί;

- Το πλήθος των δεδομένων εκπαίδευσης τείνει να είναι υπερβολικά μεγάλο.
- Δεδομένα όπως οι εικόνες έχουν μεγάλο όγκο δεδομένων
 - Π.χ. ~3MB/MPx
- Η διαθέσιμη μνήμη (RAM/VRAM) είναι σημαντικά περιορισμένη
- Υπάρχουν περιπτώσεις όπου τα σετ δεδομένων αποτελούν συνεχείς ροές δεδομένων

Εκπαίδευση σε batches - Stochastic Gradient Descent (SGD)

$$\frac{\partial \mathcal{J}(W)}{\partial W}$$

Η συνολική παράγωγος της **εμπειρικής** (σε όλα τα σημεία) συνάρτησης κόστους τις περισσότερες φορές είναι πολύ κοστοβόρα υπολογιστικά!

$$\frac{\partial \mathcal{J}_i(W)}{\partial W}$$

Αν θεωρήσουμε μόνο **ένα σημείο** (i) τότε η εμπειρική συνάρτηση ταυτίζεται με τη συνάρτηση κόστους στο σημείο αυτό.

Η παράγωγος της συνάρτησης κόστους σε ένα σημείο είναι φθηνή υπολογιστικά, αλλά αρκετά “θορυβώδης” εκτίμηση της πραγματικής εμπειρικής κλίσης (stochastic)

Εκπαίδευση σε batches - Stochastic Gradient Descent (SGD)

Αν θεωρήσουμε ένα πλήθος \mathbf{B} (batch size) από σημεία, τότε μπορούμε να ορίσουμε την αντίστοιχη (batch) εμπειρική συνάρτηση $\mathcal{J}_B(W)$

$$\frac{\partial \mathcal{J}_B(W)}{\partial W} = \frac{1}{B} \sum_{i=1}^B \frac{\partial \mathcal{J}_i(W)}{\partial W}$$

Η $\frac{\partial \mathcal{J}_B(W)}{\partial W}$ αποτελεί μία πολύ **καλύτερη ποιοτικά εκτίμηση** της πραγματικής κλίσης σε σχέση με την αντίστοιχη από ένα σημείο, παραμένει όμως **υπολογιστικά διαχειρίσιμη**.

Η παράμετρος **B** αποτελεί μία (πολύ σημαντική) υπερπαράμετρο!

Μέθοδος Stochastic Gradient Descent

1. Τυχαία (;) αρχικοποίηση των βαρών \mathbf{W}, \mathbf{b}
2. Διαμερισμός του σετ εκπαίδευσης σε τμήματα ομάδων (batches) από \mathbf{B} (**batch size**) σημεία
3. Επιλογή του ρυθμού εκμάθησης (learning rate) α
4. Μέχρις ότου να συγκλίνει η συνάρτηση, σε κάθε επανάληψη υπολογίζονται οι μερικές παράγωγοι της συνάρτησης κόστους \mathcal{J}_B ως προς τα \mathbf{W}, \mathbf{b} :

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \nabla_{\mathbf{w}} J$$

$$\frac{\partial}{\partial \mathbf{b}} J(\mathbf{w}) = \nabla_{\mathbf{b}} J$$

Πλέον επανάληψη \neq
εποχή!

και αντίστοιχα ανανεώνονται τα \mathbf{W}, \mathbf{b} ως εξής :

$$\mathbf{w} = \mathbf{w} - \alpha \nabla_{\mathbf{w}} J$$

$$\mathbf{b} = \mathbf{b} - \alpha \nabla_{\mathbf{b}} J$$

Άλλοι αλγόριθμοι βελτιστοποίησης

- **Momentum SGD** : σε σχέση με τον απλό SGD αλγόριθμο, η ανανέωση των βαρών πλέον δεν γίνεται κατά τη διεύθυνση της μέγιστης κλίσης αλλά θεωρείται ένα δυναμικό μοντέλο κατά το οποίο τα βάρη κινούνται με σταθερή (κατά μέτρο) “ταχύτητα” και επιδρά πάνω τους μία δύναμη εν είδει “βαρύτητας”
 - Nesterov Accelerated Gradient (NAG)
- **AdaGrad** : μεταβάλλει δυναμικά τον ρυθμό εκμάθησης ανά παράμετρο μειώνοντάς τον καθώς προσεγγίζει κάποιο τοπικό ελάχιστο



SGD χωρίς momentum

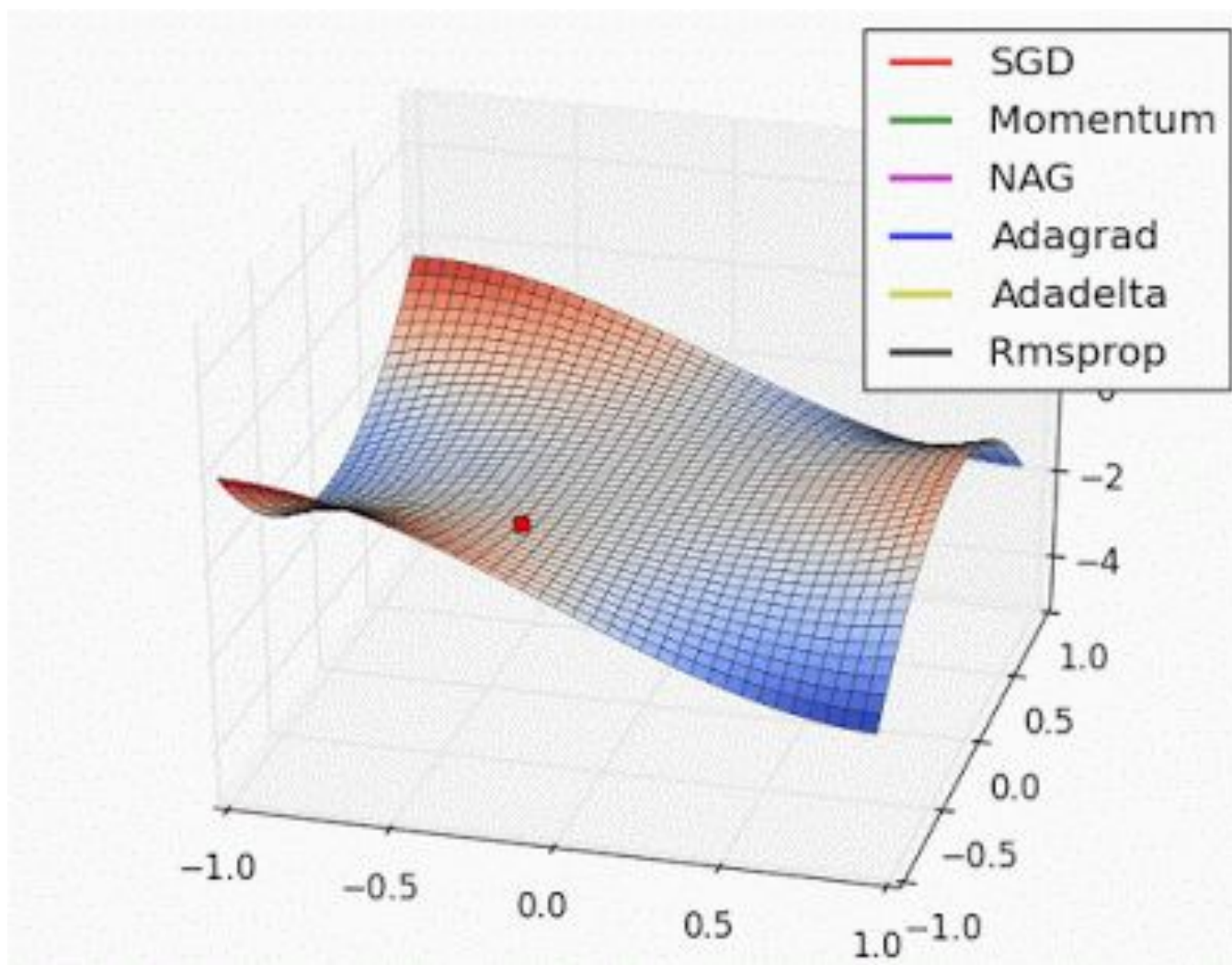


SGD με momentum

Άλλοι αλγόριθμοι βελτιστοποίησης

- **AdaDelta** : παραλλαγή του AdaGrad κατά την οποία η μεταβολή του ρυθμού εκμάθησης εξαρτάται από ένα συγκεκριμένο χρονικό “παράθυρο”
- **RMSProp** : παραλλαγή του AdaGrad όπου ο ρυθμός εκμάθησης μεταβάλλεται χρησιμοποιώντας ένα κυλιόμενο μέσο όρο των τετραγώνων των παρελθοντικών κλίσεων - ο ρυθμός εκμάθησης εφαρμόζεται ανά παράμετρο !
- **Adam** : συνδυασμός των RMSProp και Momentum SGD - ο ρυθμός εκμάθησης ανά παράμετρο μεταβάλλεται συναρτήσει ενός φθίνοντος εκθετικού μέσου των παρελθοντικών κλίσεων.

Αλγόριθμοι βελτιστοποίησης



Εκπαίδευση σε batches

Ορισμοί μεγεθών κατά την εκπαίδευση:

Step/Iteration/Επανάληψη : Μία επανάληψη της μεθόδου SGD

Batch size : Πλήθος σημείων που αξιοποιεί ανά επανάληψη η μέθοδος SGD

Epoch/Εποχή : Μία πλήρης σειρά επαναλήψεων της μεθόδου SGD σε όλα τα σημεία του σετ εκπαίδευσης

Steps per epoch : Ο αριθμός των επαναλήψεων που απαιτούνται για να ολοκληρωθεί μία εποχή. Υπολογίζεται ρητά (στρογγυλοποιώντας προς τα πάνω) από τον λόγο του πλήθους των σημείων εκπαίδευσης διά το batch size

Παράδειγμα : Έστω ότι διαθέτουμε ένα σετ δεδομένων με 5000 σημεία εκπαίδευσης και ότι έχουμε επιλέξει batch size = 256.

$$\text{Steps per epoch} : 5000/256 = 49.53 \sim 20$$

Regularization

Regularization

Σημαντικοί ορισμοί :

Capacity : Μέγεθος που υποδεικνύει το πλήθος όλων των πιθανών μοντέλων που μπορεί να αναπαραστήσει η εκάστοτε αρχιτεκτονική για κάθε πιθανό συνδυασμό των βαρών.

Bias : Κάθε δέσμευση στους πιθανούς συνδυασμούς των βαρών που μεταβάλλει τη χωρητικότητα (capacity) του δικτύου.

Variance : Το σύνολο της διασποράς των δεδομένων που μοντελοποιείται από το δίκτυο.

Η συνολική διασπορά των δεδομένων αποτελεί το άθροισμα της διασποράς λόγω της κατανομής p_{data} και της διασποράς του θορύβου στα δεδομένα

Generalization error : Αποτελεί το σχετικό σφάλμα μεταξύ του εκτιμώμενου σφάλματος στο σετ εκπαίδευσης και του πραγματικού σφάλματος στο σετ ελέγχου ή/και σε νέα δεδομένα.

Regularization

Τι είναι ?

- Κάθε τεχνική που χρησιμοποιείται για να δεσμεύσει το πρόβλημα βελτιστοποίησης με στόχο την αποφυγή “αχρείαστα” σύνθετων μοντέλων.

Γιατί χρειαζόμαστε αυτές τις τεχνικές ?

- Για να επιτρέψουμε στο μοντέλο τη μέγιστη δυνατή γενίκευση στα δεδομένα ελέγχου και σε νέα δεδομένα.

Παραδείγματα

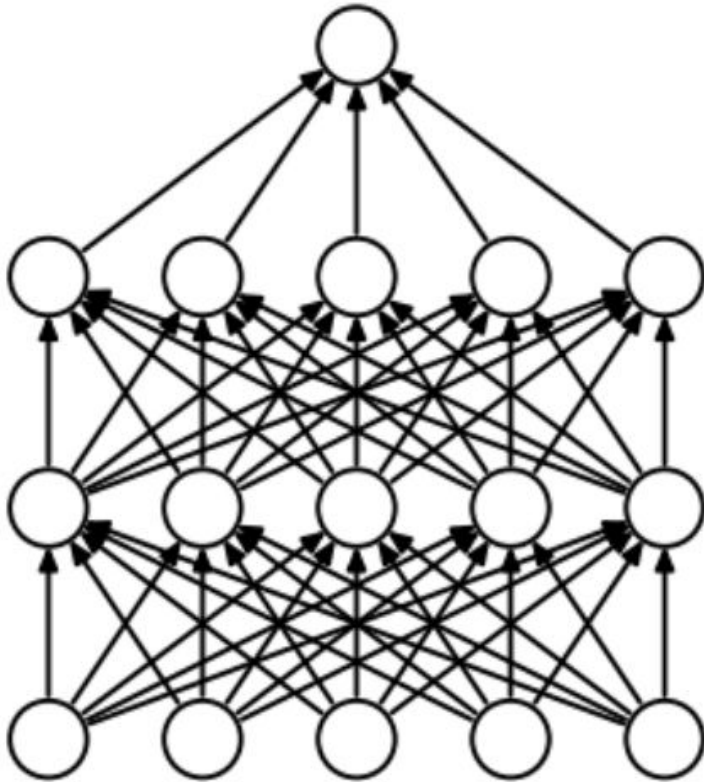
- Dropout
- Early stopping
- L1/L2 Regularization
- Data Augmentation
- ...

Regularization I : Dropout

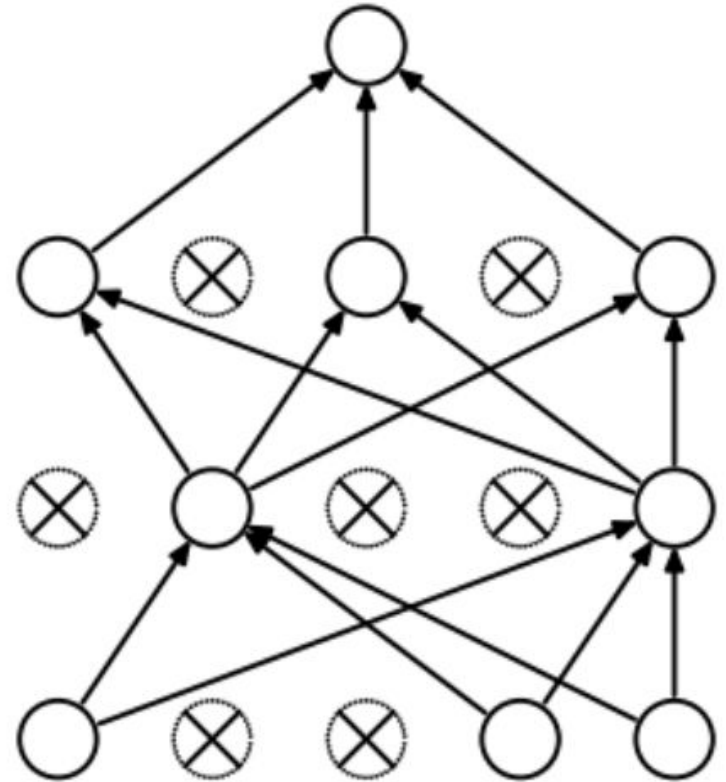
Αυτή η τεχνική αναφέρεται στην **επιλεκτική αφαίρεση νευρώνων** σε ένα νευρωνικό δίκτυο.

- Πρακτικά, επιλέγουμε να αγνοήσουμε **τυχαία** ένα συγκεκριμένο πλήθος νευρώνων κατά τη διάρκεια της εκπαίδευσης του δικτύου (forward/backward pass).
- Σε κάθε βήμα εκπαίδευσης, κάθε νευρώνας (για τον οποίο εφαρμόζουμε αυτή την τεχνική) παραμένει στο δίκτυο με πιθανότητα **p** ή αγνοείται με πιθανότητα **$1-p$** .
- Συνήθως εφαρμόζεται **ανά στρώση** (layer)
- Η πιθανότητα p αποτελεί μία ακόμη **υπερπαράμετρο** του δικτύου!

Regularization I : Dropout



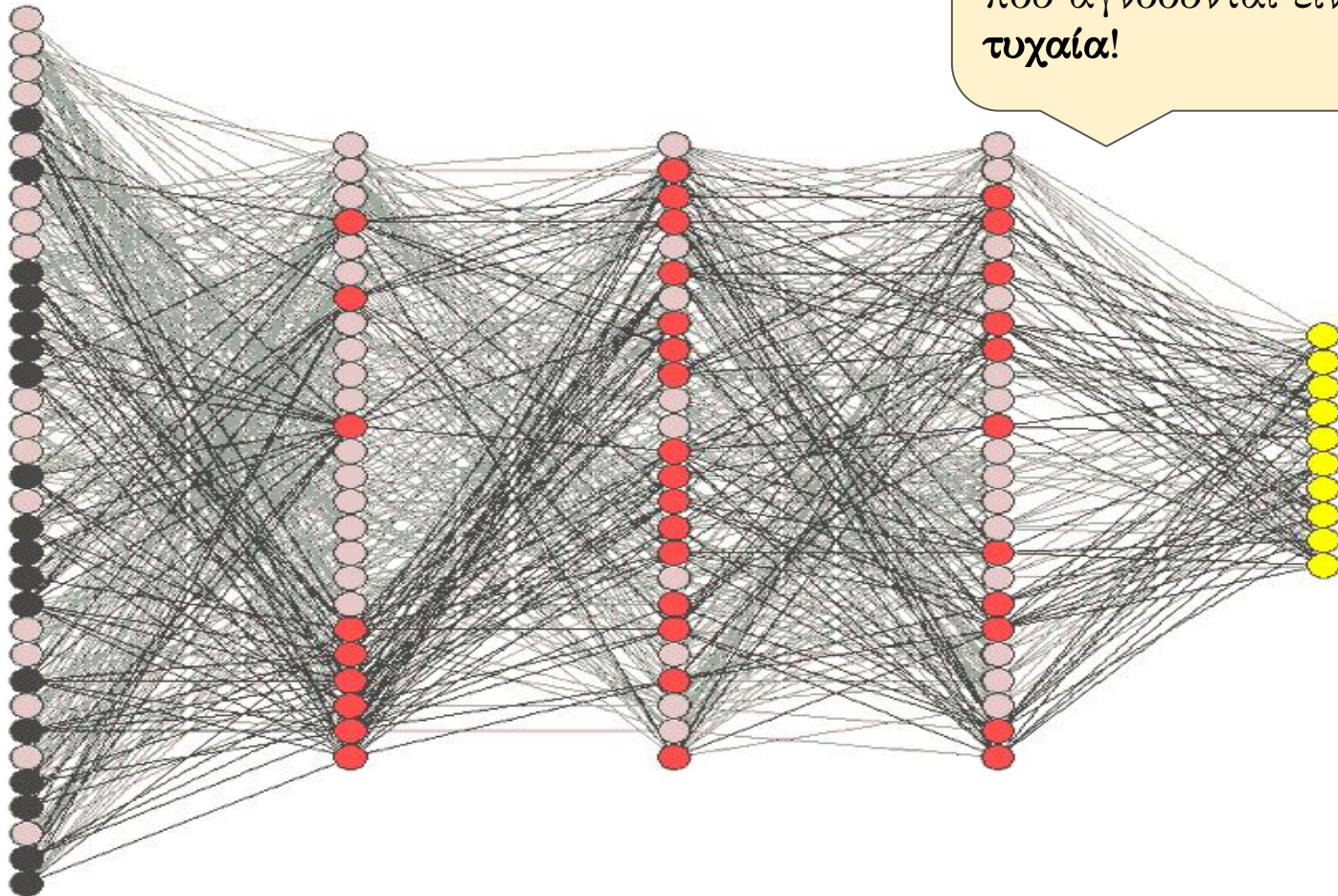
Τυπικό νευρωνικό δίκτυο



Εφαρμογή dropout σε
μία επανάληψη

Regularization I : Dropout

Σε κάθε επανάληψη, η επιλογή των νευρώνων που αγνοούνται είναι τυχαία!



Regularization I : Dropout

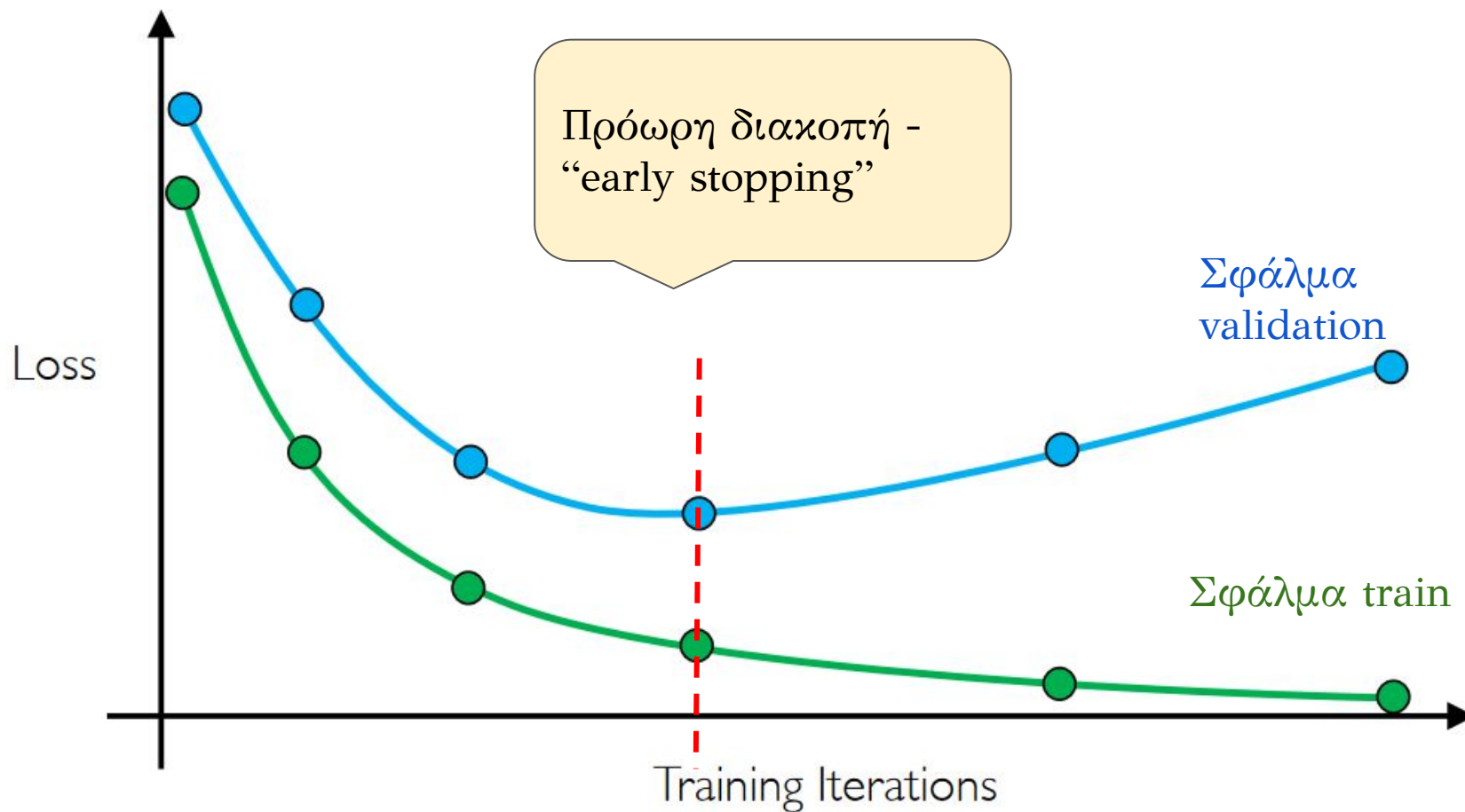
Γιατί, όμως, να θέλουμε να απενεργοποιήσουμε κάποιους νευρώνες του δικτύου?

- Θέλουμε να αποτρέψουμε το **overfitting** του μοντέλου!
- Θέλουμε να ισοκατανήμουμε τη στατιστική σημαντικότητα των νευρώνων.
 - ◆ Δεν θέλουμε το δίκτυο να εξαρτάται από ένα νευρώνα

Παρατηρήσεις

- Εκμάθηση πιο “εύρωστων” (robust) χαρακτηριστικών
- Το πλήθος των επαναλήψεων για τη σύγκλιση του μοντέλου διπλασιάζεται για $p=0.5$, αλλά πλέον ο χρόνος εκπαίδευσης ανά εποχή μειώνεται.
- Δεδομένων L κρυφών νευρώνων, τα πιθανά μοντέλα που προκύπτουν είναι 2^L .

Regularization II : Early Stopping



Regularization II : Early Stopping

Παρατηρήσεις

- Η πρόωρη διακοπή εκπαίδευσης του δικτύου αποσκοπεί στην **ελαχιστοποίηση του σφάλματος γενίκευσης** (Generalization Error)
- Αποφυγή **overfit**
- Μπορούμε να εφαρμόσουμε αυτή τη τεχνική και σε **διαφορετικές μετρικές** πέραν του εμπειρικού σφάλματος, όπως π.χ. την ακρίβεια ταξινόμησης
- Ως υπερπαραμέτρος του δικτύου αποτελεί το πλήθος των εποχών μετά από το οποίο θα σταματήσει η εκπαίδευση του δικτύου.
- Μπορεί να εφαρμοστεί είτε χειροκίνητα είτε αυτόματα

Regularization III : Weight Regularization

Η εκπαίδευση ενός δικτύου για πολλαπλές εποχές οδηγεί στην εξειδίκευση των βαρών στο σετ εκπαίδευσης.

Δημιουργία βαρών με πολύ υψηλές τιμές:

- Πιθανή αριθμητική αστάθεια.
- Μικρές αλλαγές στα δεδομένα → Έντονες αλλαγές στο αποτέλεσμα
- Υψηλή διασπορά (variance) - Μικρό bias

Τι μπορούμε να κάνουμε;

- Μπορούμε να **δεσμεύσουμε τη συνολική νόρμα** (μέγεθος) των βαρών σε επίπεδο νευρώνα ή στρώσης

Regularization III : Weight Regularization

Πως μπορούμε να δεσμεύσουμε τη νόρμα των βαρών;

- Εισάγοντας ένα νέο όρο στη συνολική συνάρτηση κόστους προς βελτιστοποίηση

$$\mathcal{L}_{total}(W) = \mathcal{L}_{loss}(W) + \lambda \mathcal{L}_{reg}(W_l)$$

$$\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_p$$

Παραλλαγές:

1. L_1 Regularization (Lasso) $\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_1$
2. L_2 Regularization (Ridge) $\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_2$

Weight Decay

$$w_{t+1} = w_t - \alpha \nabla_w J - \lambda w_t$$

Regularization III : Weight Regularization

Πως μπορούμε να δεσμεύσουμε το σχήμα των βαρών;

- Εισάγοντας έναν κανονικοποιητικό όρο στον συνολικό σκορ βελτιστοποίησης

Εφαρμόζοντας την κανονικοποίηση αυτή στις αποκρίσεις των νευρώνων προκύπτει μία άλλου τύπου κανονικοποίηση (**Activation regularization**)

$$\mathcal{L}_{total}(W) = \mathcal{L}_{data}(W) + \mathcal{L}_{reg}(W_l)$$

$$\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_p$$

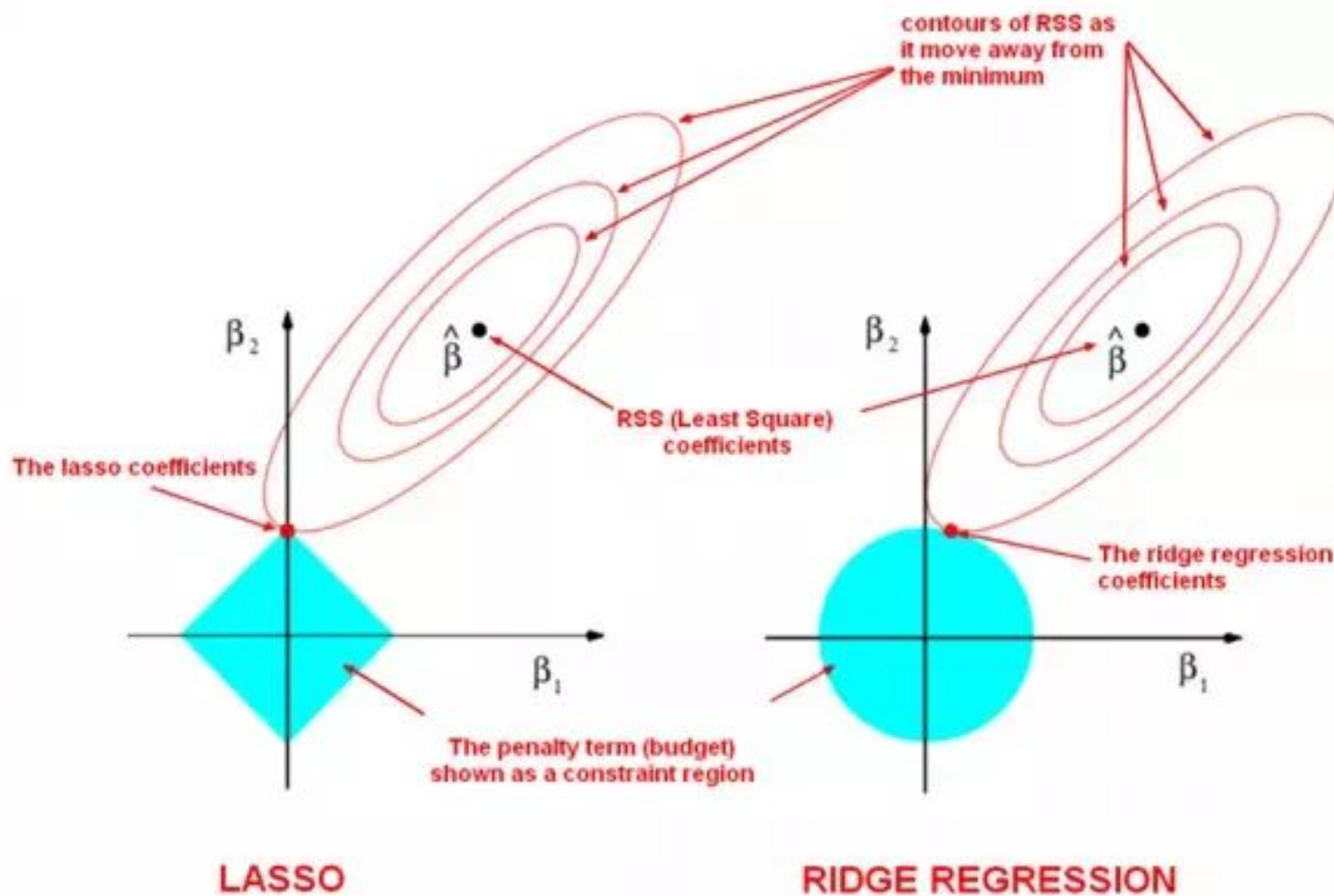
Παραλλαγές:

1. L_1 Regularization (Lasso) $\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_1$
2. L_2 Regularization (Ridge) $\mathcal{L}_{reg}(W) = \|\mathbf{W}\|_2$

Weight Decay

$$w_{t+1} = w_t - \alpha \nabla_w J - \lambda w_t$$

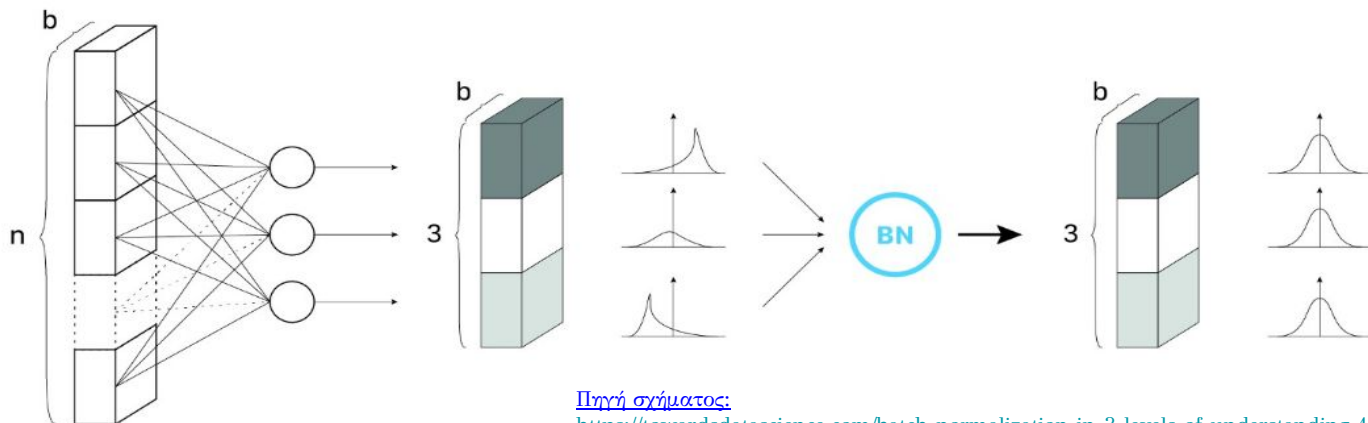
Regularization III : Weight Regularization



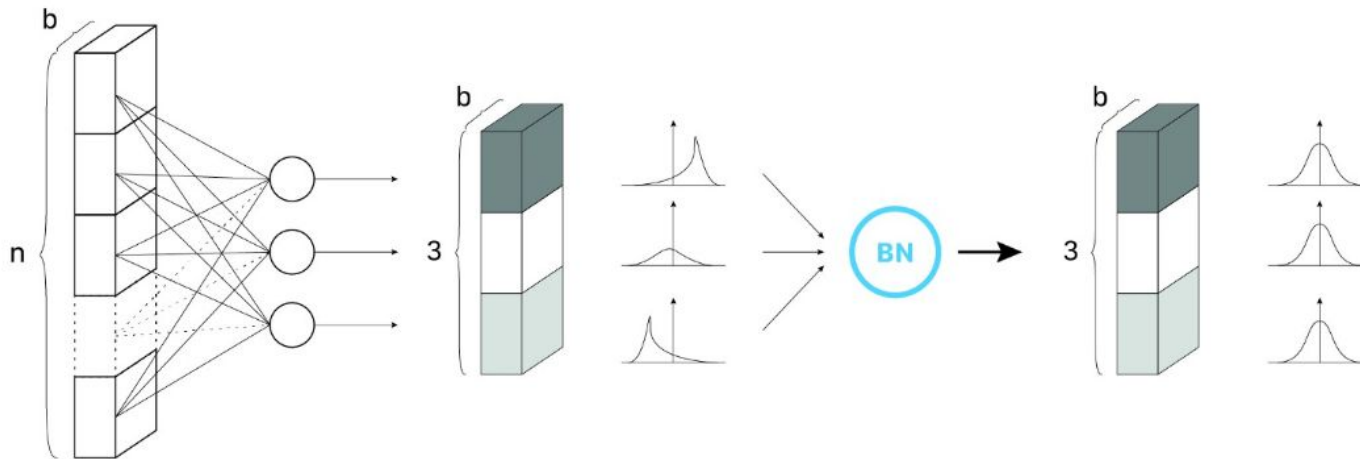
Regularization IIb : Batch Normalization

Το BN αφορά τον υπολογισμό στατιστικών μεγεθών 1ης και 2ης τάξης ανά χαρακτηριστικό σε ένα batch με στόχο την κανονικοποίηση του εκάστοτε χαρακτηριστικού ώστε αυτό να ακολουθεί την τυποποιημένη κανονική κατανομή.

- Συνήθως εφαρμόζεται πριν τη μη-γραμμική συνάρτηση ενεργοποίησης
- Βοηθάει στην ταχύτερη σύγκλιση της εκπαίδευσης
- Συνήθης πρακτική σε ΤΝΔ και ΣΝΔ (CNN)
 - Σε ανατροφοδοτούμενα δίκτυα (RNN) συνήθως προτιμάται μία παραλλαγή του BN, το Layer Normalization (καθολικά στο batch)



Regularization IIb : Batch Normalization



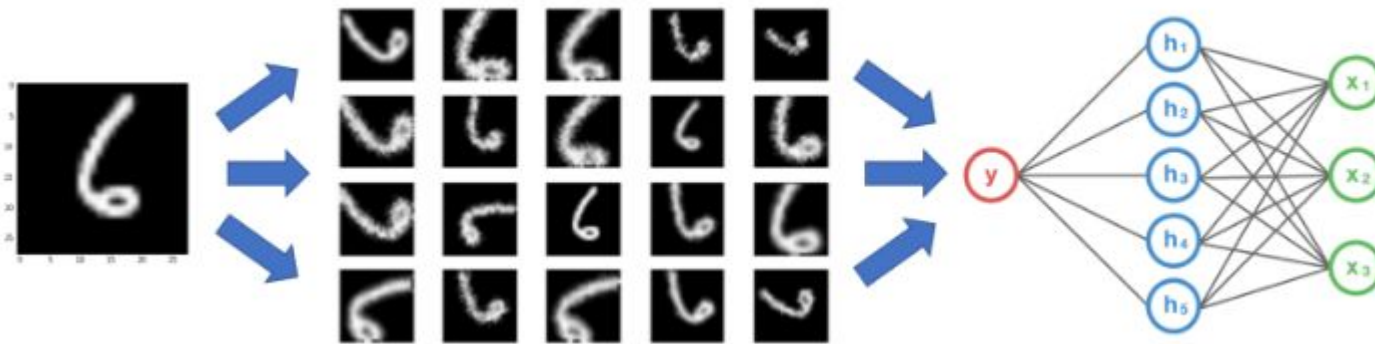
- Το BN εφαρμόζεται για όλα τα δείγματα ενός batch
- Υπολογίζεται ανά χαρακτηριστικό !
- Στην “παραγωγή” δεν έχει νόημα καθώς συνήθως έχουμε μοναδιαία batch sizes

$$(1) \mu = \frac{1}{n} \sum_i Z^{(i)} \quad (2) \sigma^2 = \frac{1}{n} \sum_i (Z^{(i)} - \mu)^2$$

$$(3) Z_{norm}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 - \epsilon}} \quad (4) \check{Z} = \gamma * Z_{norm}^{(i)} + \beta$$

Regularization IV : Data Augmentation

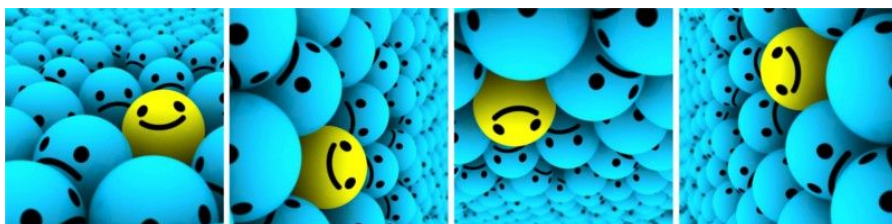
- Τεχνητή δημιουργία νέων δεδομένων εκπαίδευσης από τα διαθέσιμα δεδομένα εφαρμόζοντας τεχνικές ανάλογα με το προς-επίλυση πρόβλημα (domain-specific).
- Σε ένα πραγματικό σενάριο, μπορεί να διαθέτουμε ένα σύνολο εικόνων εκπαίδευσης, οι οποίες μπορούν να αξιοποιηθούν και σε περισσότερες πιθανές εκδοχές τους/ αναπαραστάσεις τους!
 - Π.χ. Διαφορετική κλίμακα, θόρυβος κλπ.



Regularization IV : Data Augmentation



Οριζόντια και
κάθετη ανάκλαση
(flip)



Στροφή (rotation)



Κλίμακα (scaling)

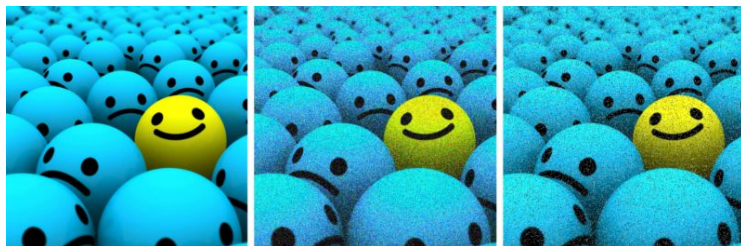


“Κόψιμο” εικόνας
(crop)

Regularization IV : Data Augmentation



Μετάθεση
(translation)



Προσθήκη γκαουσιανού
θορύβου (Gaussian Noise)



winter Yosemite → summer Yosemite



summer Yosemite → winter Yosemite

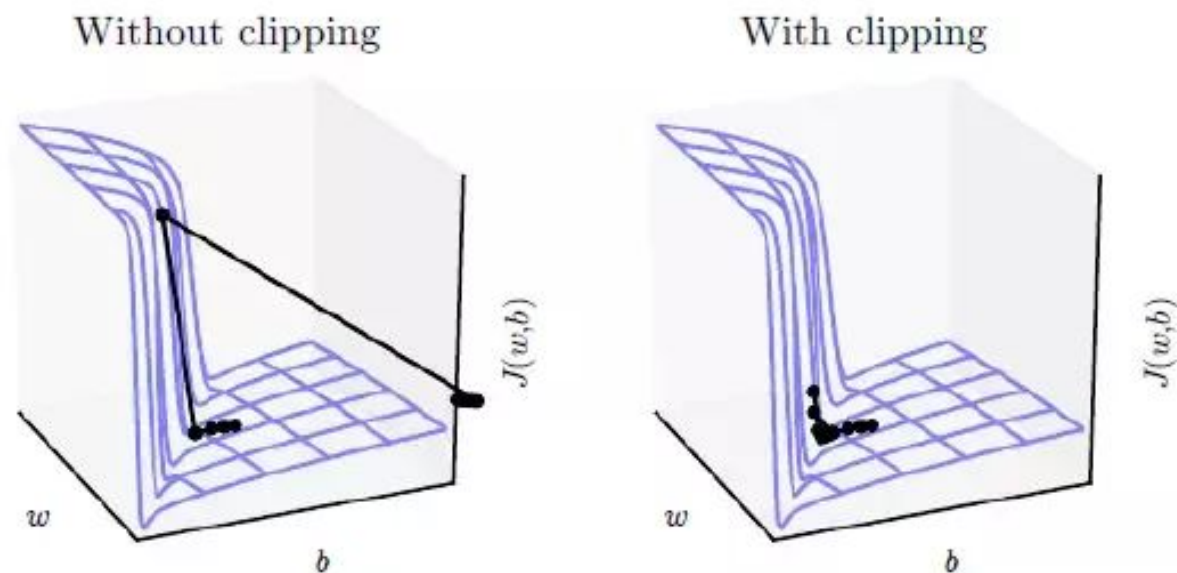
Style transfer με
GANs

Regularization V : Άλλες τεχνικές

Υπάρχουν πολυάριθμες τεχνικές για regularization

Παραδείγματα:

- **Gradient Norm Clipping**
- Adversarial training

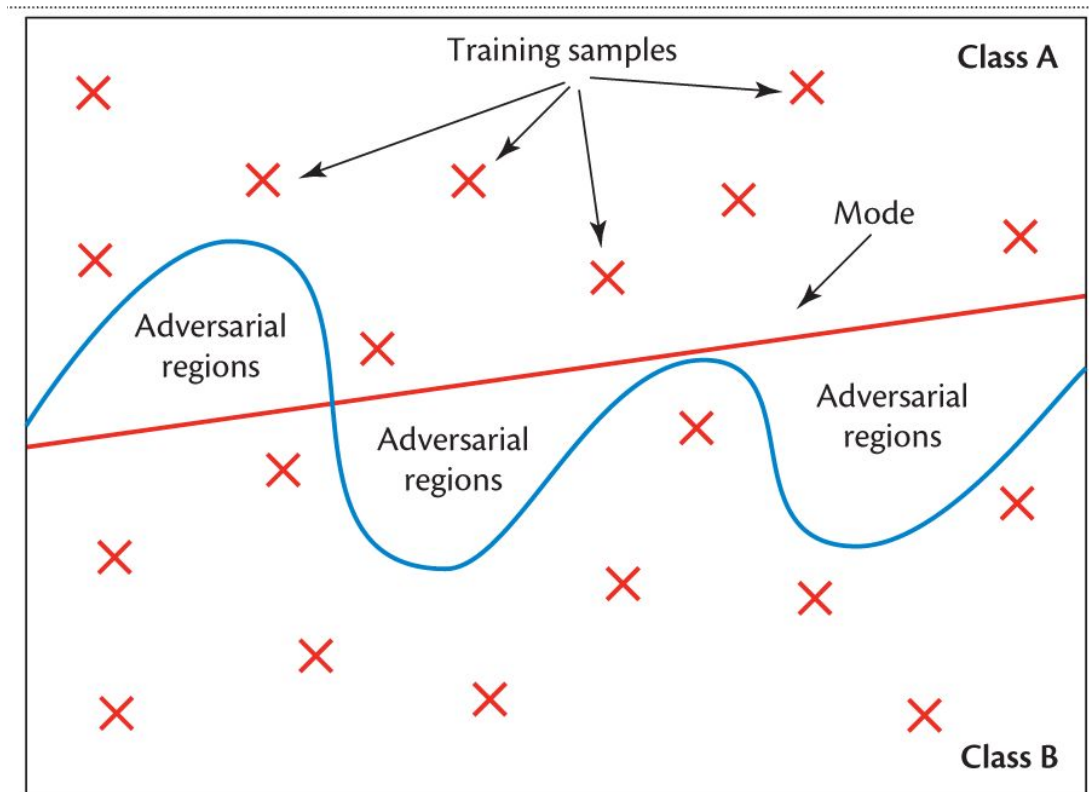


Regularization V : Άλλες τεχνικές

Υπάρχουν πολυάριθμες τεχνικές για regularization

Παραδείγματα:

- Gradient Norm Clipping
- **Adversarial training**



Αξιολόγηση

Μετρικές και σύγκριση ΤΝΔ

Μετρικές αξιολόγησης

Στη περίπτωση ενός **δυναδικού λογικού προβλήματος** (boolean) μπορούμε να κατηγοριοποιήσουμε το αποτέλεσμα της πρόβλεψης στις εξής κατηγορίες :

	Θετικό Δείγμα	Αρνητικό Δείγμα
Θετική Πρόβλεψη	True Positive (TP)	False Positive (FP)
Αρνητική Πρόβλεψη	False Negative (FN)	True Negative (TN)

Η περίπτωση της **δυναδικής ταξινόμησης** μπορεί να θεωρηθεί ως ένα δυναδικό λογικό πρόβλημα αν θεωρήσουμε τη μία κατηγορία ως “θετική” και την άλλη ως “αρνητική”

Μετρικές αξιολόγησης

Για τη περίπτωση της ταξινόμησης σε πολλαπλές κατηγορίες, ορίζονται οι ίδιες ποσότητες ανά κατηγορία, θεωρώντας επιμέρους δυαδικές ταξινομήσεις.

Αξιολόγηση κατηγορίας K		
	Θετικό K Δείγμα	Αρνητικό K Δείγμα
Θετική K Πρόβλεψη	True Positive (TP)	False Positive (FP)
Αρνητική K Πρόβλεψη	False Negative (FN)	True Negative (TN)

Μετρικές αξιολόγησης : Δείκτες

$$Precision = \frac{TP}{TP+FP}$$

Τι ποσοστό των προβλέψεων ήταν σωστό

$$Recall = \frac{TP}{TP+FN}$$

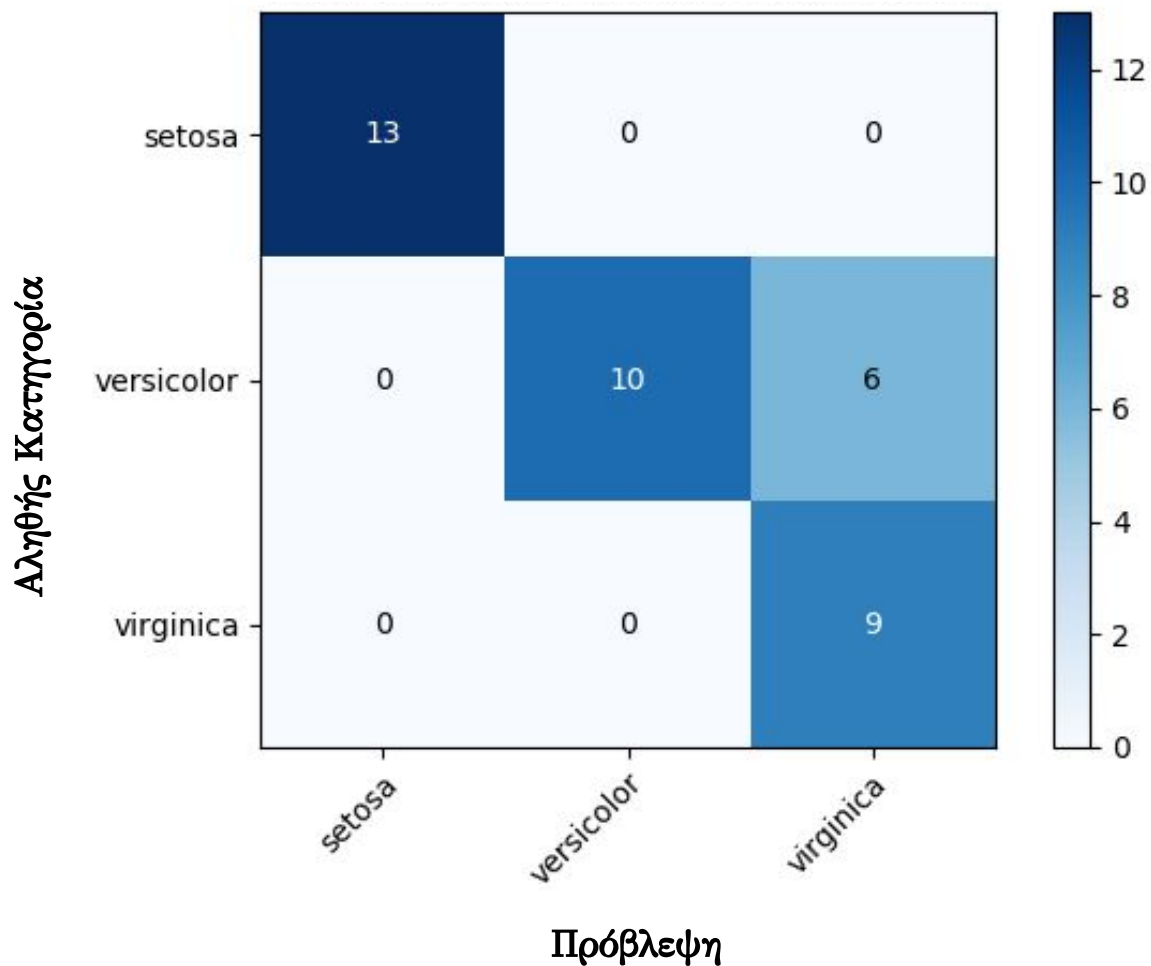
Τι ποσοστό των αληθών δειγμάτων προβλέφθηκε σωστά

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

$$F_1 = 2 * \frac{precision+recall}{precision*recall}$$

Δείκτες ακρίβειας της ταξινόμησης

Μετρικές αξιολόγησης : Πίνακας σύγχυσης (Confusion Matrix)



Μετρικές αξιολόγησης : Πίνακας σύγχυσης (Confusion Matrix)

Αληθής Κατηγορία

12	0	0	0	0	0	1	0	0	0
0	10	0	0	0	0	0	0	2	0
0	0	10	0	0	0	7	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	10	0	0	0	0	0
0	3	2	0	0	10	0	0	0	0
0	0	0	0	0	0	17	0	0	0
1	0	0	0	0	0	3	TP	0	FN
0	3	1	1	0	0	1	FP	13	TN
1	1	0	3	0	0	5	0	0	8

Πρόβλεψη

Μετρικές αξιολόγησης : Εφαρμογές

Ανίχνευση αντικειμένου (Object Detection)



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



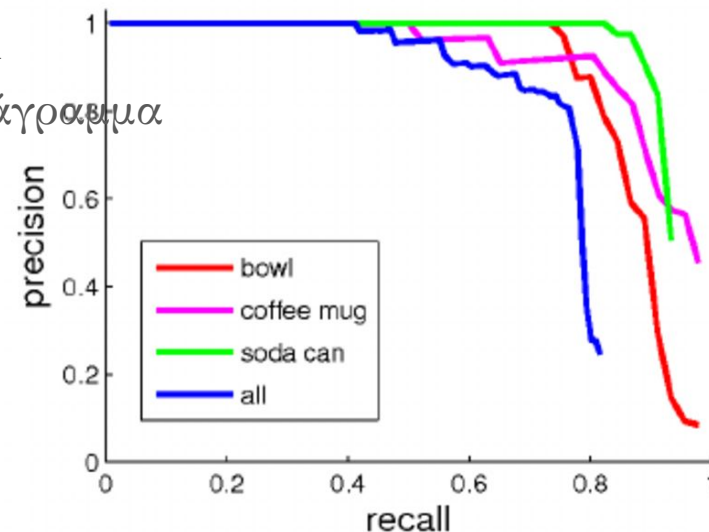
Η μετρική **IoU (Intersection over Union)** δείχνει το ποσοστό επικάλυψης μεταξύ δύο “κουτιών περιγράμματος” (bounding boxes), συγκεκριμένα μεταξύ της πρόβλεψης και του δεδομένου αληθείας για ένα αντικείμενο στην εικόνα.

Συνήθως προκαθορίζεται ένα κατώφλι (threshold), π.χ. $\text{IoU} = 0.5$, ώστε να κατηγοριοποιείται η πρόβλεψη ως σωστή (TP) ή λανθασμένη (FP).

Μετρικές αξιολόγησης : Εφαρμογές

Ανίχνευση αντικειμένου (Object Detection)

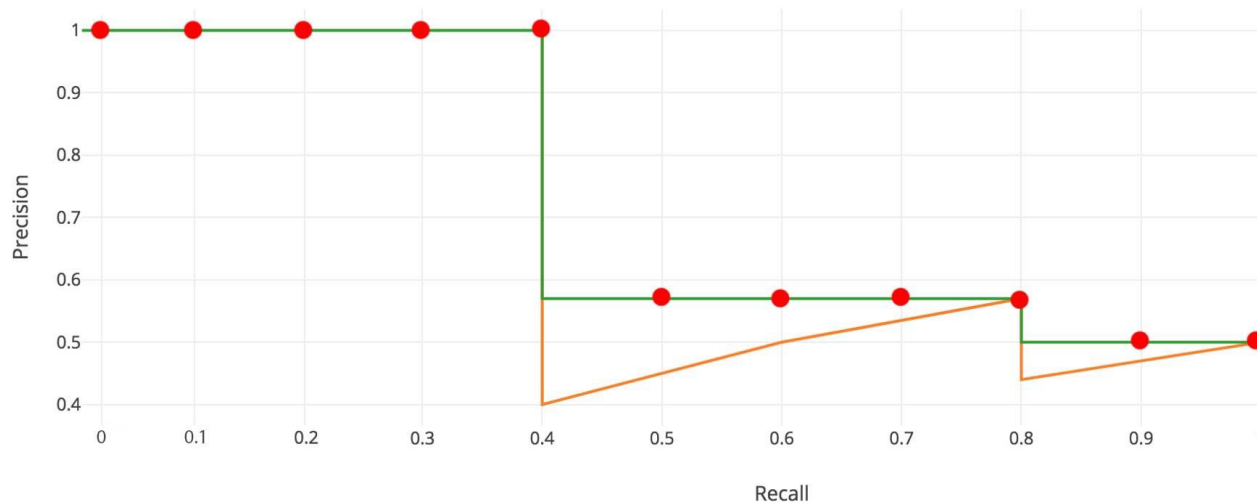
- **TP** : Πλήθος σωστών ανιχνεύσεων
- **FP** : Πλήθος λανθασμένων ανιχνεύσεων
- **Recall** : Ποσοστό των υπάρχοντων στη σκηνή αντικειμένων που ανιχνεύθηκαν σωστά ανά κατηγορία
- **Precision** : Ποσοστό των ανιχνεύσεων που όντως υπάρχουν στη σκηνή ανά κατηγορία
- **Καμπύλη Precision/Recall** : Για κάθε πιθανή τιμή κατωφλίου στην IoU υπολογίζεται το Precision και το Recall ανά κατηγορία δημιουργώντας ένα διάγραμμα της μορφής :



Μετρικές αξιολόγησης : Εφαρμογές

Ανίχνευση αντικειμένου (Object Detection)

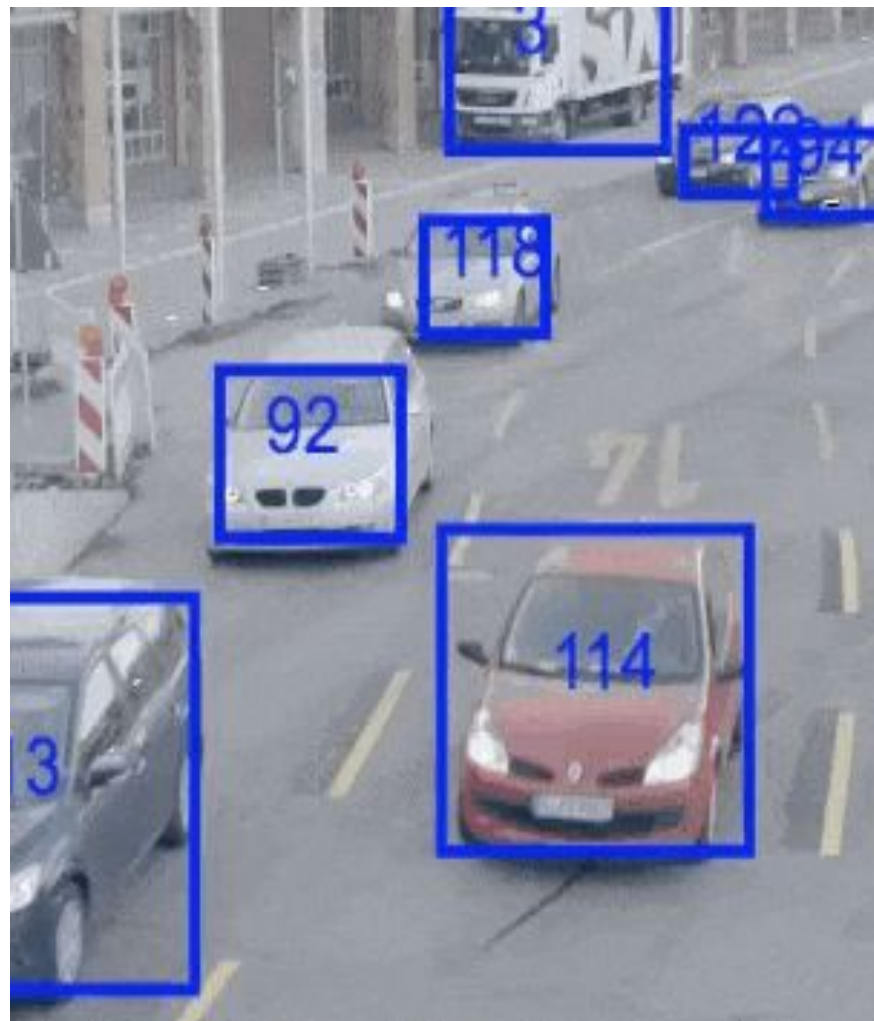
- **Average Precision (AP)** : Υπολογίζεται από το διάγραμμα Precision/Recall παίρνοντας τον μέσο όρο Precision τιμών από 11 ίσα χωρισμένων δειγμάτων Recall τιμών ανά κατηγορία.
 - Ως τιμή δείγματος ορίζεται η μέγιστη τιμή του Precision από το σημείο δειγματοληψίας r και προς “τα πάνω”.
- **Mean Average Precision (mAP)** : Ο μέσος όρος των ανά κατηγορία AP τιμών.



Μετρικές αξιολόγησης : Εφαρμογές

Παρακολούθηση πολλών αντικειμένων (Multiple Object Tracking)

- **MOTP** : Μέσο σφάλμα σε IoU στις εκτιμώμενες θέσεις των στόχων κατά τη διάρκεια του βίντεο. Αντικατοπτρίζει την ικανότητα του αλγορίθμου να παρακολουθεί αντικείμενα χωρίς να αξιολογεί τη δυνατότητά του να παρακολουθεί συνεχείς τροχιές (trajectories).
- **ID Switches** : Πλήθος στόχων που άλλαξαν ταυτότητα (ID) μεταξύ τους
- **MOTA** : Συνδυάζει το πλήθος των FP (εν είδει Object Detection per frame), των “χαμένων” στόχων και των ID Switches.
- **H_z** : Ταχύτητα επεξεργασίας



Σύνοψη

- **Βελτιστοποίηση παραμέτρων δικτύου**
 - Ελαχιστοποίηση του “Εμπειρικού” κόστους
 - Μέθοδος καταβιβασμού κλίσης
 - Stochastic Gradient Descent (batches)
 - Adam, RMSProp, ...
 - Back-propagation -- υπολογισμός μερικών παραγώγων
- **Υπερπαράμετροι**
 - Ρυθμός εκμάθησης
 - Πλήθος εποχών
 - Early stopping
 - Overfit vs Underfit
- **Τεχνικές Regularization**
- **Αξιολόγηση**
 - Ταξινόμηση
 - TP/FP/TN/FN
 - Accuracy/F1/precision/Recall
 - Ταξινόμηση σε πολλαπλές κατηγορίες
 - Πίνακας σύγχυσης
 - Λοιπές εφαρμογές



RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων Δεδομένων Παρατήρησης Γης



GitHub

<https://github.com/rslab-ntua>