Μηχανική Μάθηση: Μαθηματικό Υπόβαθρο

Κωνσταντίνος Καραμανής

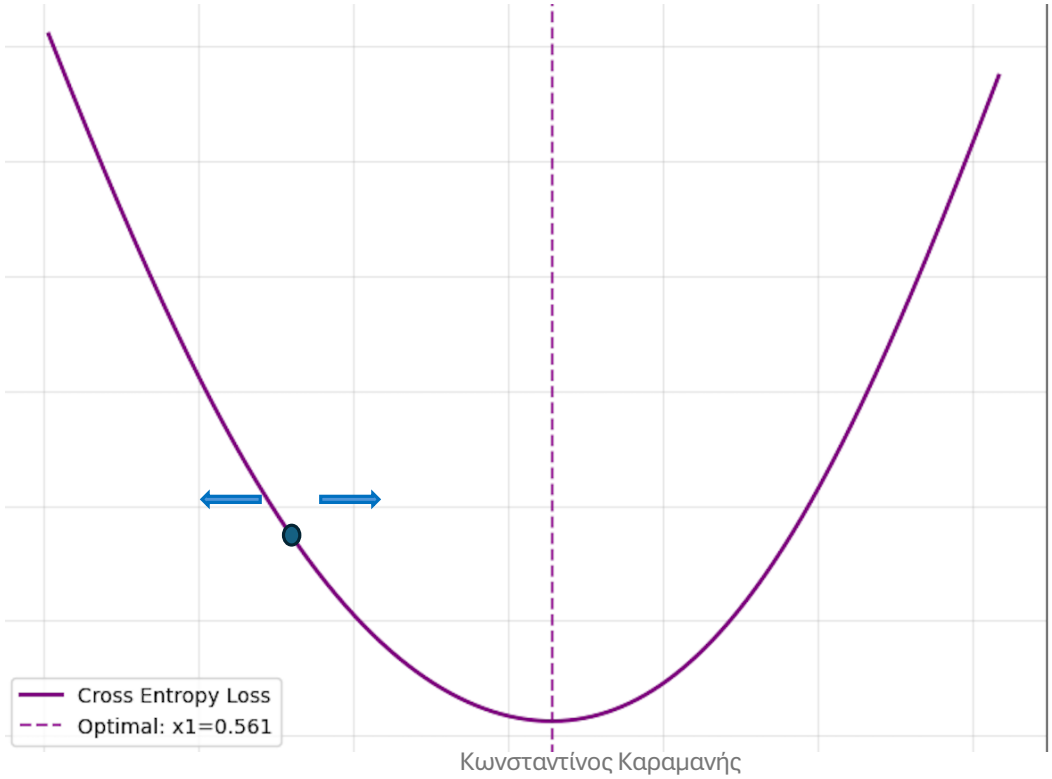
The University of Texas at Austin & Archimedes/Athena RC

constantine@utexas.edu

https://caramanis.github.io/

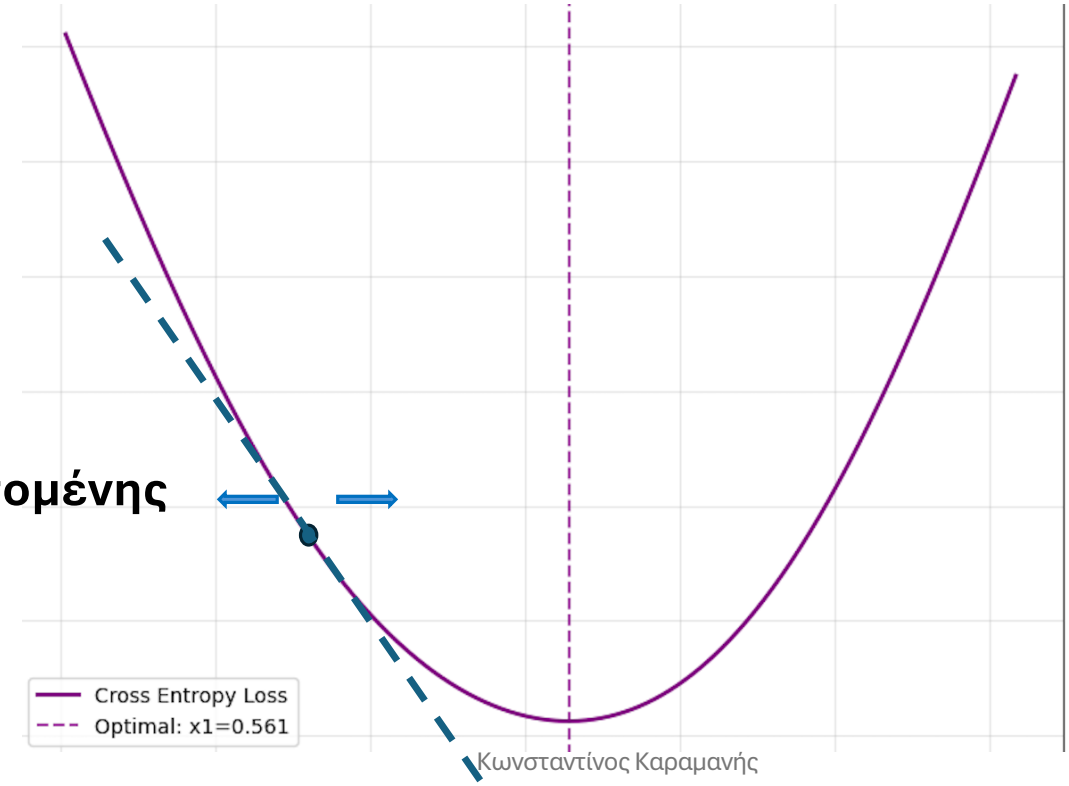
Ας θυμηθούμε τα προηγούμενα...

Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)



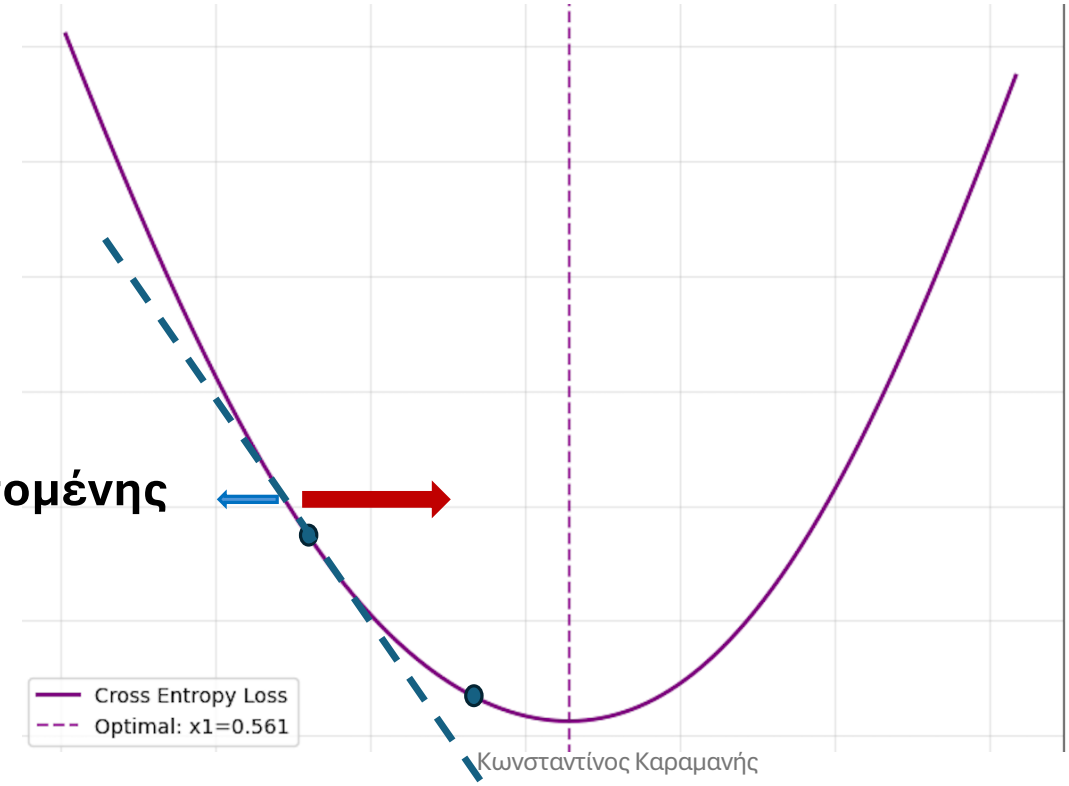
Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**



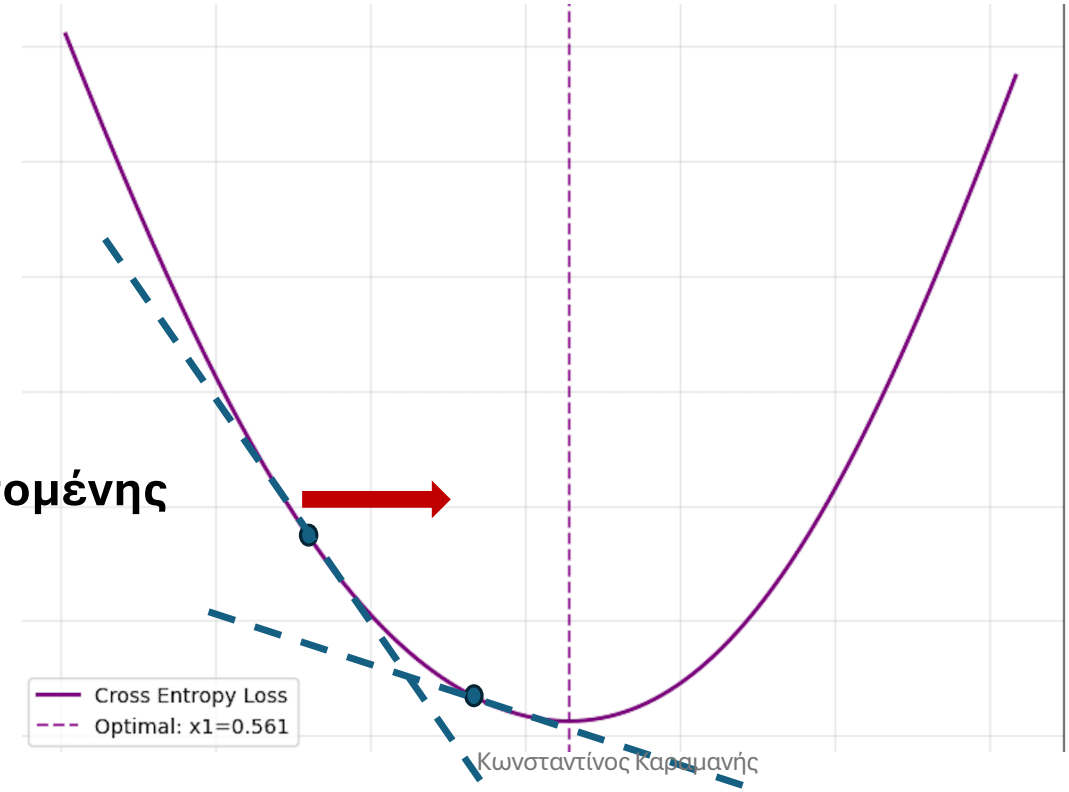
Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**



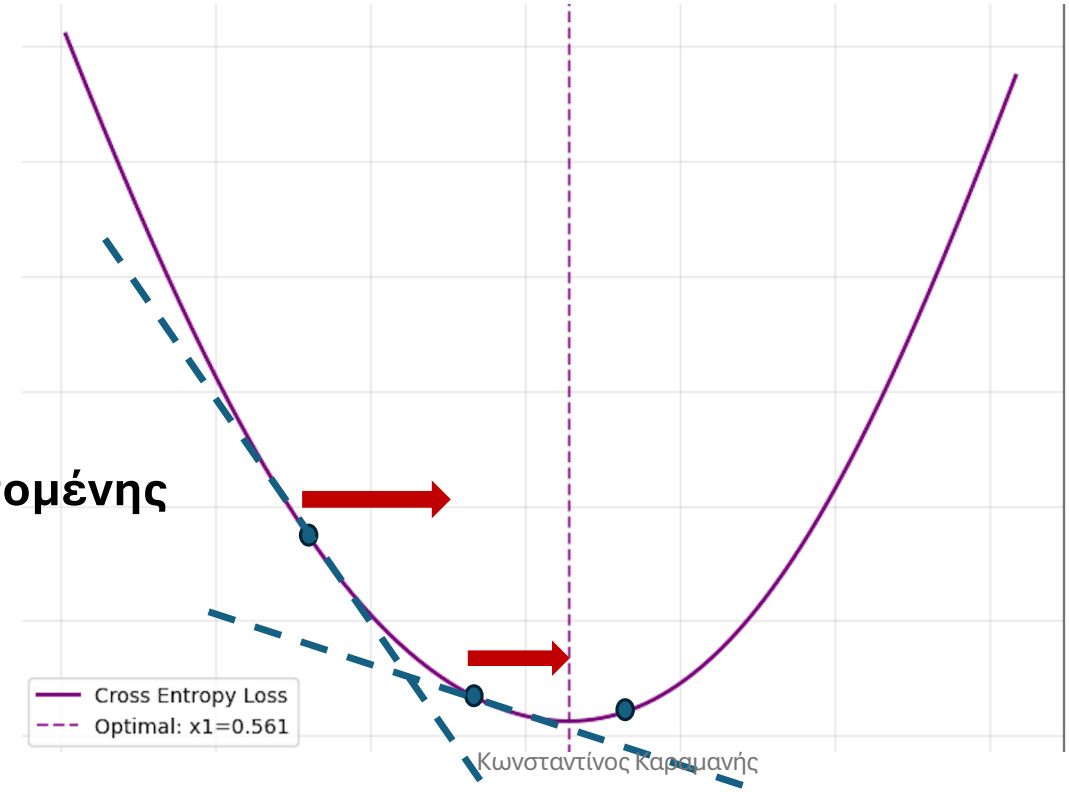
Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**



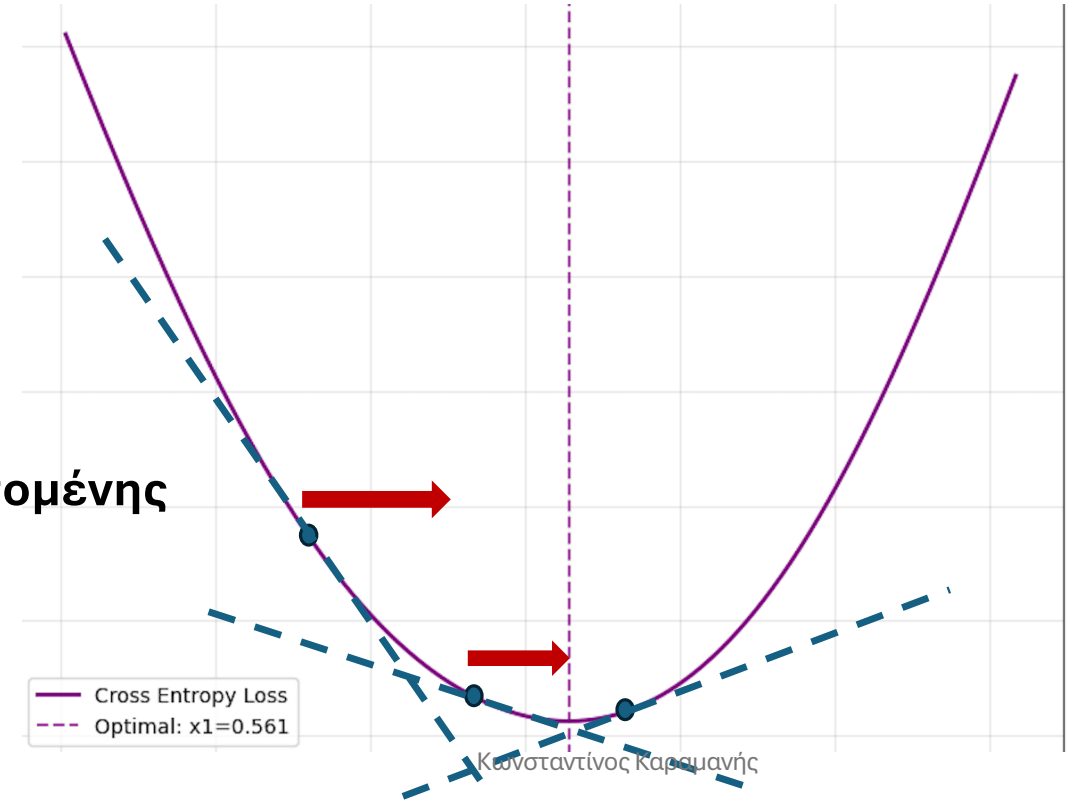
Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**



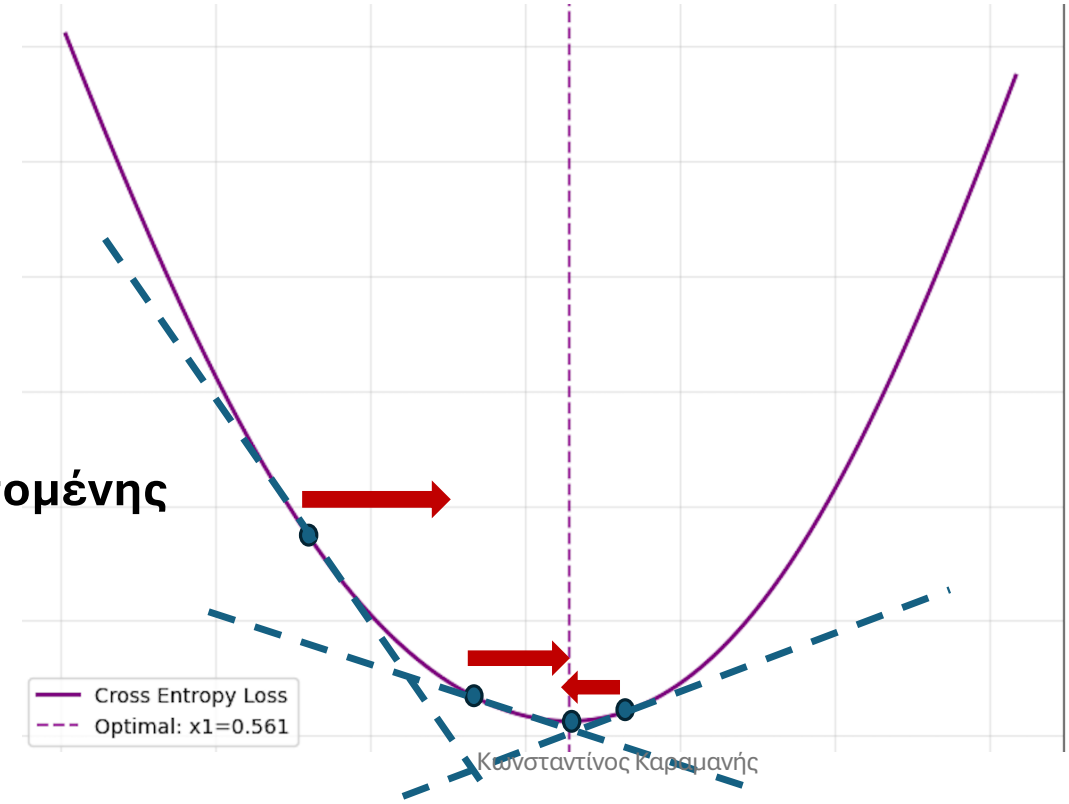
Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**



Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

κλίση της εφαπτομένης
= η **παράγωγος**

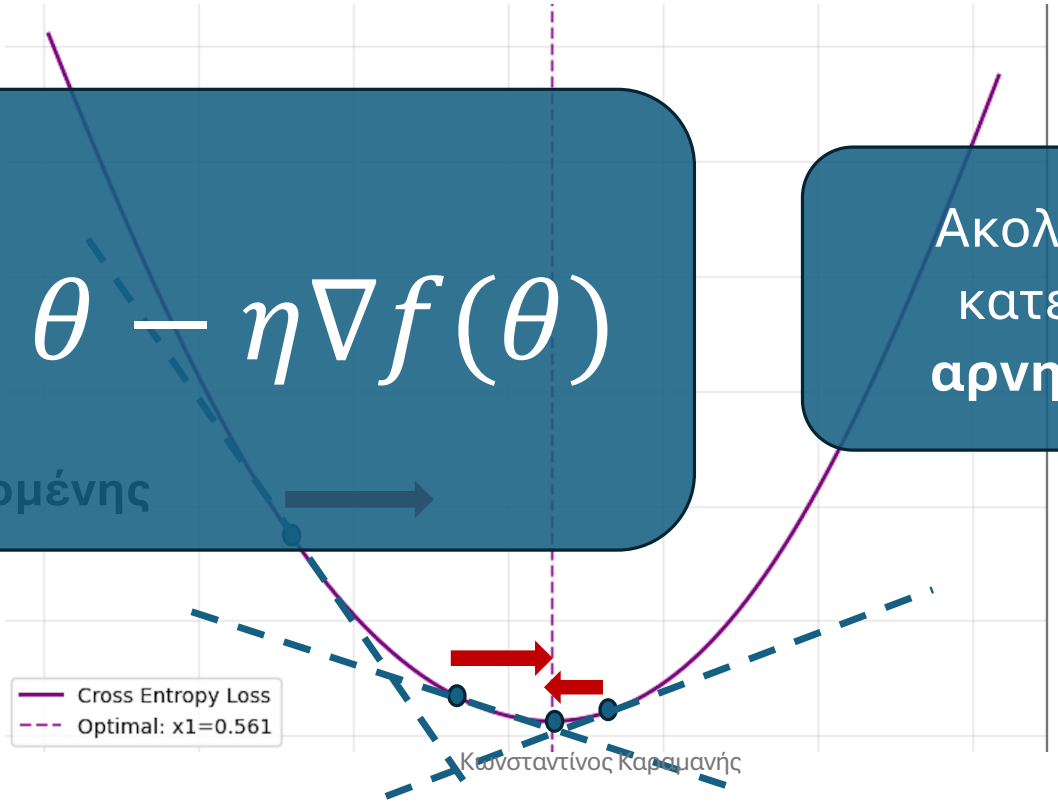


Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)

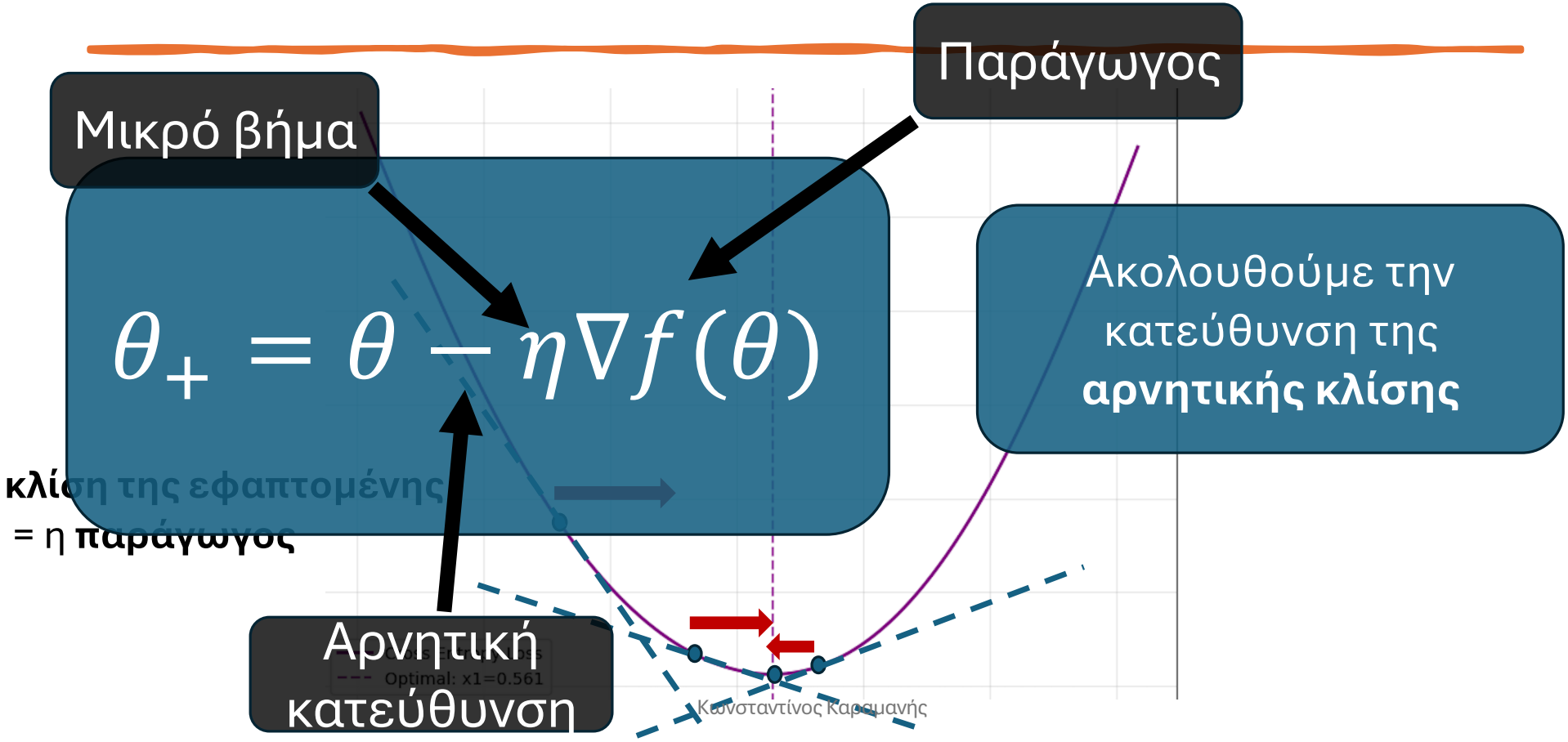
$$\theta_+ = \theta - \eta \nabla f(\theta)$$

κλίση της εφαπτομένης
= η **παράγωγος**

Ακολουθούμε την κατεύθυνση της **αρνητικής κλίσης**



Μέθοδος: Gradient Descent (Κάθοδος Κλίσης)



Η Μέθοδος:
Gradient Descent
(Κάθοδος Κλίσης)

$$\theta_+ = \theta - \eta \nabla f(\theta)$$

Εδώ αρχίζει μια πλούσια και πολυμελετημένη περιοχή των εφαρμοσμένων μαθηματικών: η θεωρία και οι αλγόριθμοι βελτιστοποίησης. (Optimization Theory and Algorithms)

Η Μέθοδος:
Gradient Descent
(Κάθοδος Κλίσης)

$$\theta_+ = \theta - \eta \nabla f(\theta)$$

Θεμελιώδεις ερωτήσεις:
Ποια χαρακτηριστικά της συνάρτησης $f(\theta)$ καθορίζουν πόσα *βήματα* της μεθόδου απαιτούνται για να προσεγγίσουμε το βέλτιστο σημείο θ^*

Βάση αρχίζει μια πλούσια και πολυμελετημένη θεωρία των εφαρμοσμένων μαθηματικών:

Η Θεωρία και Αλγόριθμοι της Βελτιστοποίησης
(Optimization Theory and Algorithms)

Η Μέθοδος:
Gradient Desce
(Κάθοδος Κλίσ

$$\theta_+ = \theta - \eta \nabla f(\theta)$$

Θεμελιώδεις ερωτήσεις:

Π ης

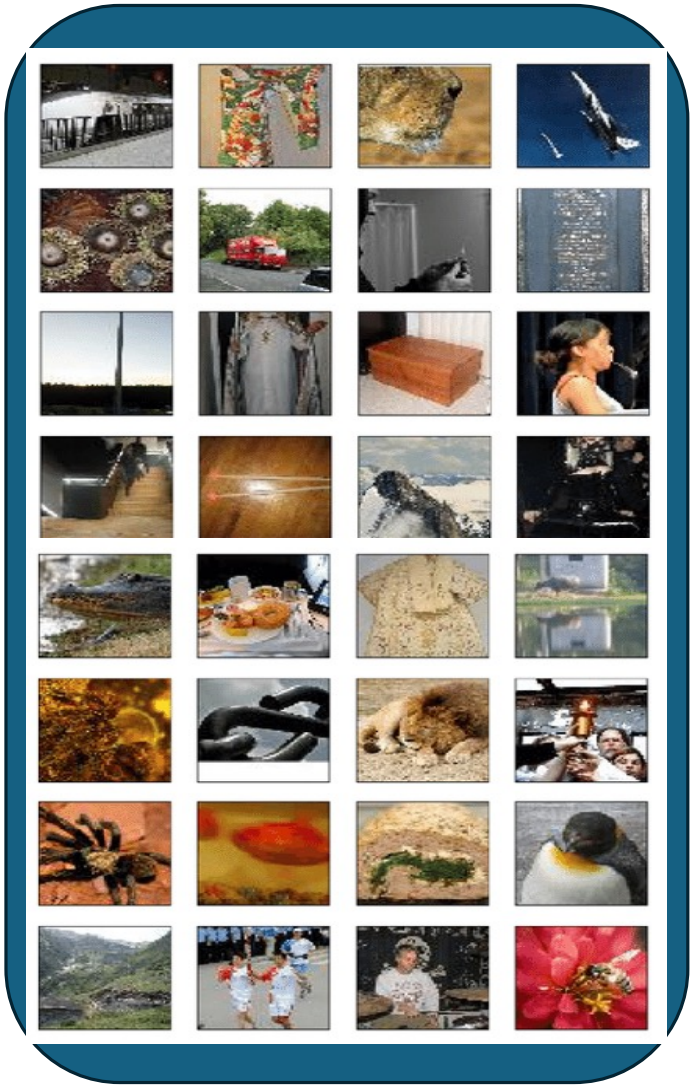
π μένη
ών:
θ*

οίησης

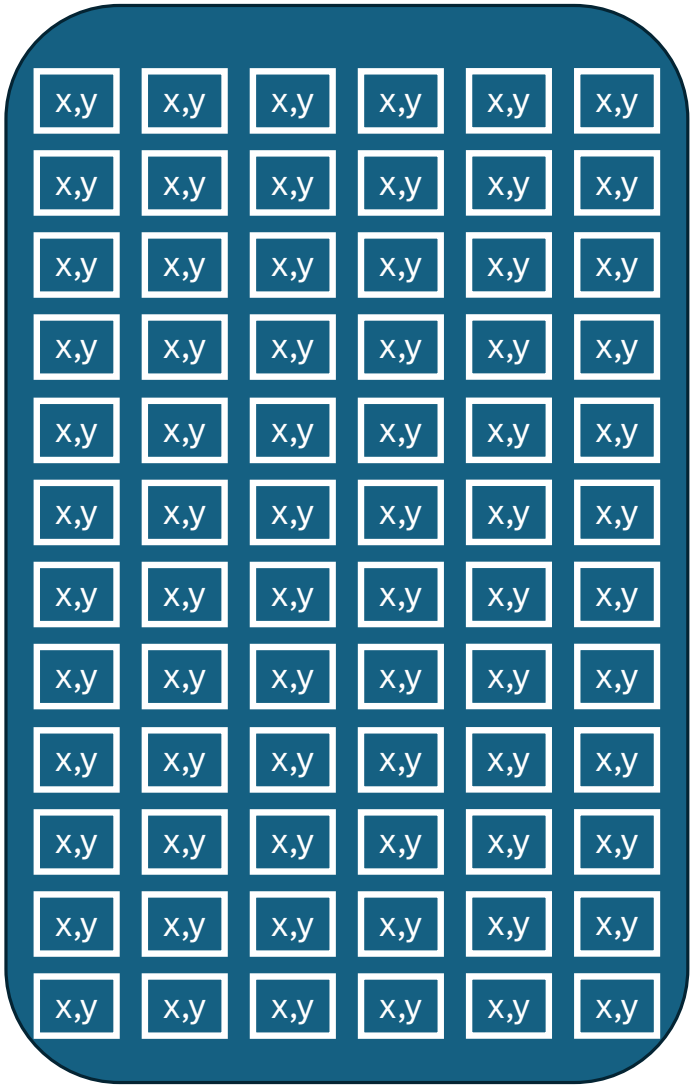
Πώς εφαρμόζεται ή χρησιμοποιείται στην εκπαίδευση και στη χρήση των νευρωνικών δικτύων;

Gradient Descent
&
Νευρωνικά Δίκτυα

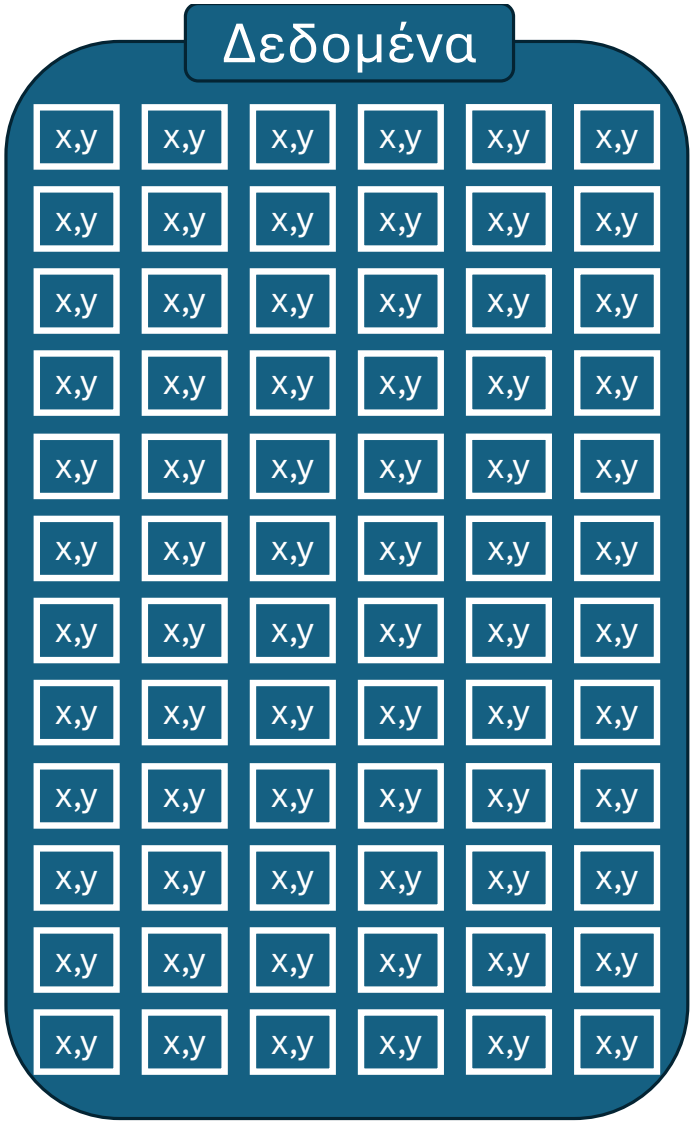
Κωνσταντίνος Καραμανής

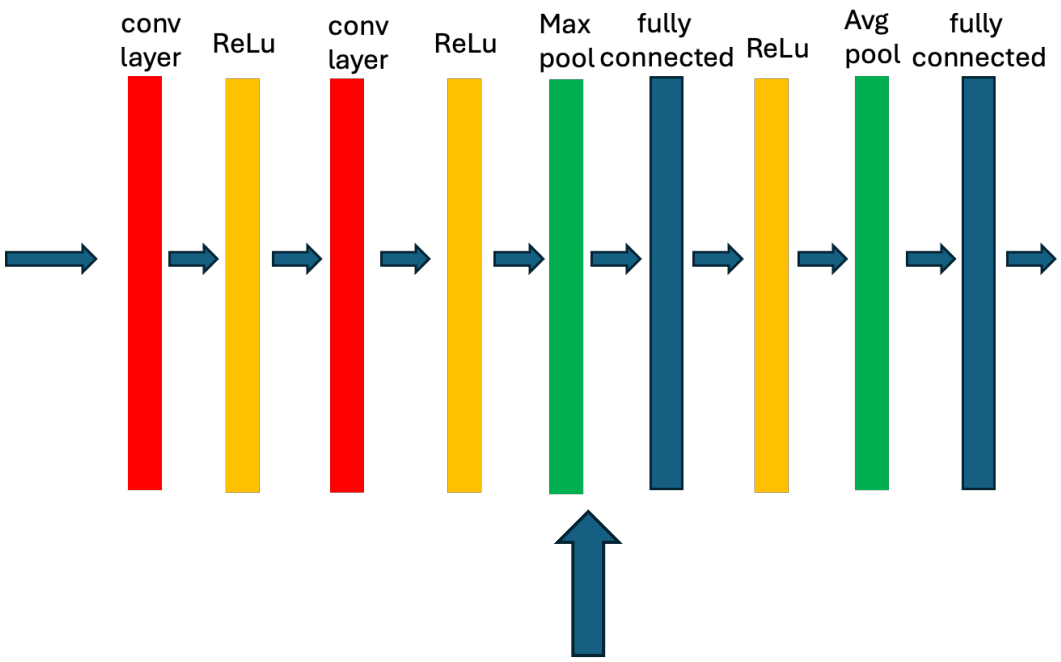


Δεδομένα

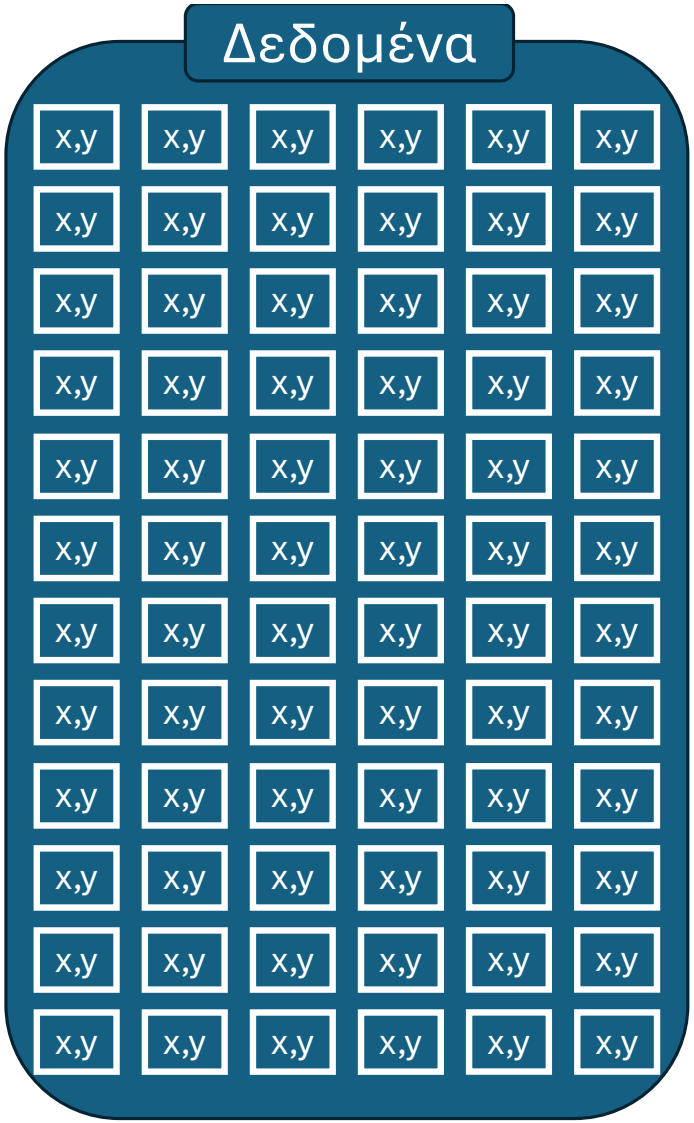


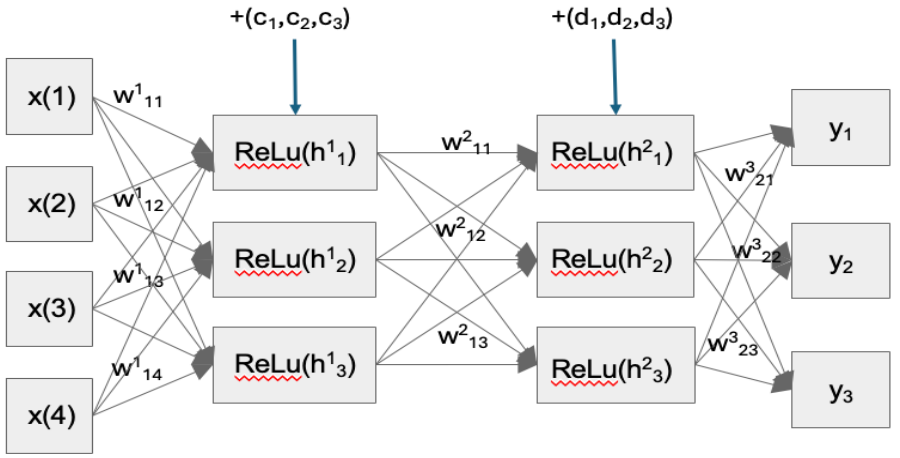
Δεδομένα


Δεδομένα
x,y


conv layer
ReLu
conv layer
ReLu
Max pool
fully connected
ReLu
Avg pool
fully connected

Νευρωνικό


Δεδομένα
x,y

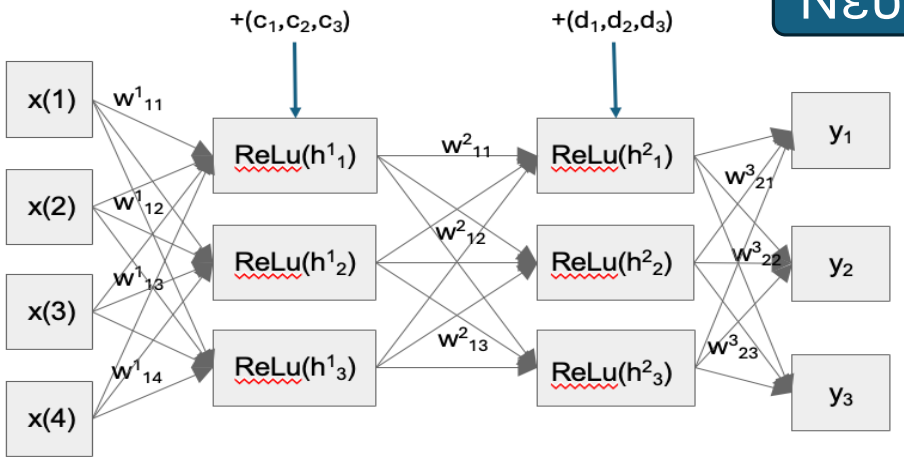

+(c1,c2,c3)
+(d1,d2,d3)
x(1)
x(2)
x(3)
x(4)
w1 11
w1 12
w1 13
w1 14
ReLu(h1 1)
ReLu(h1 2)
ReLu(h1 3)
w2 11
w2 12
w2 13
ReLu(h2 1)
ReLu(h2 2)
ReLu(h2 3)
w3 21
w3 22
w3 23
y1
y2
y3

Νευρωνικό

Δεδομένα

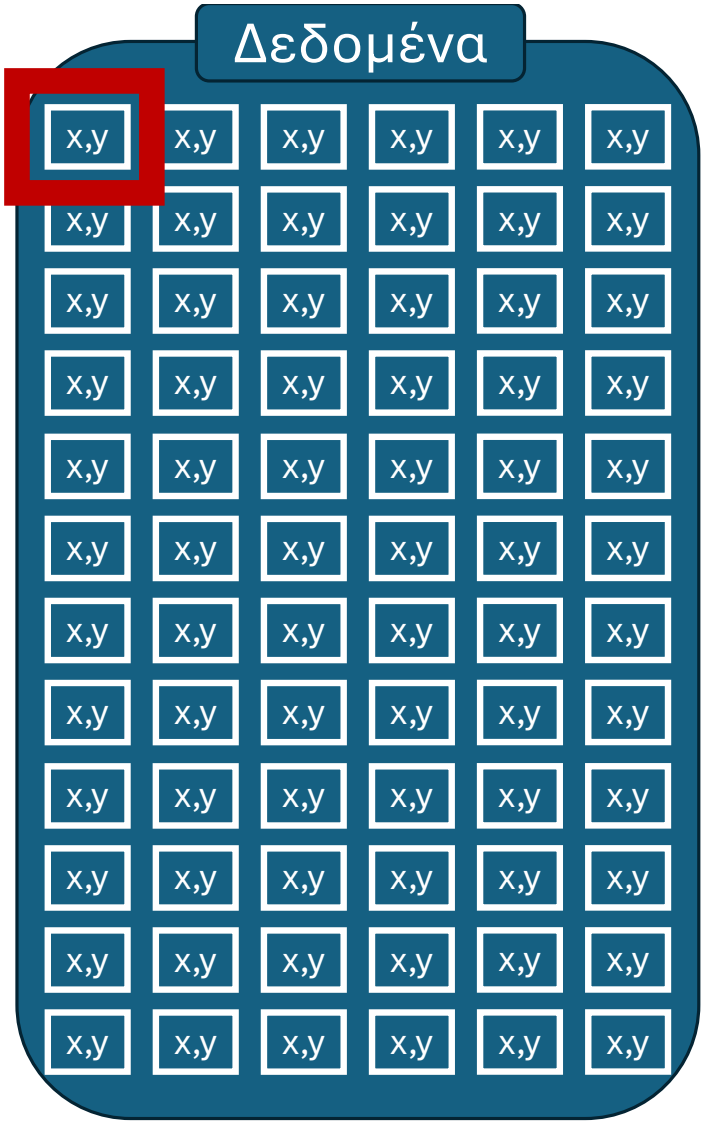
x,y

Νευρωνικό

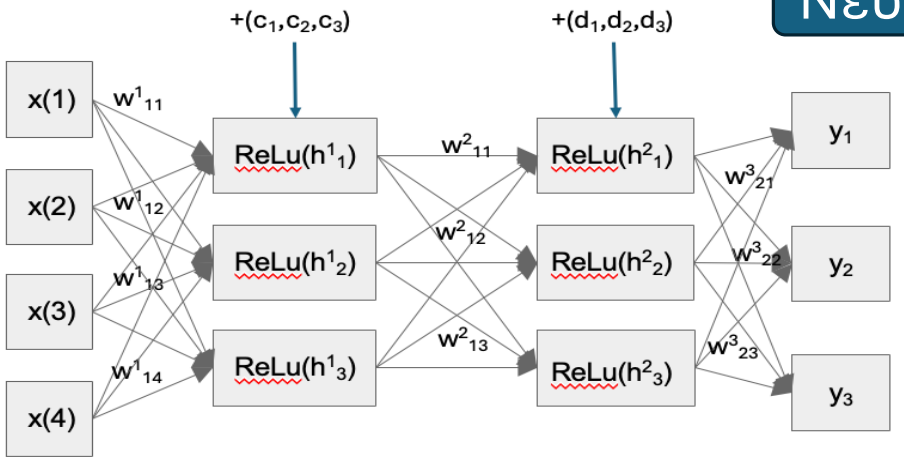


$$\theta = (W^1, c, W^2, d, W^3)$$

Παράμετροι του μοντέλου



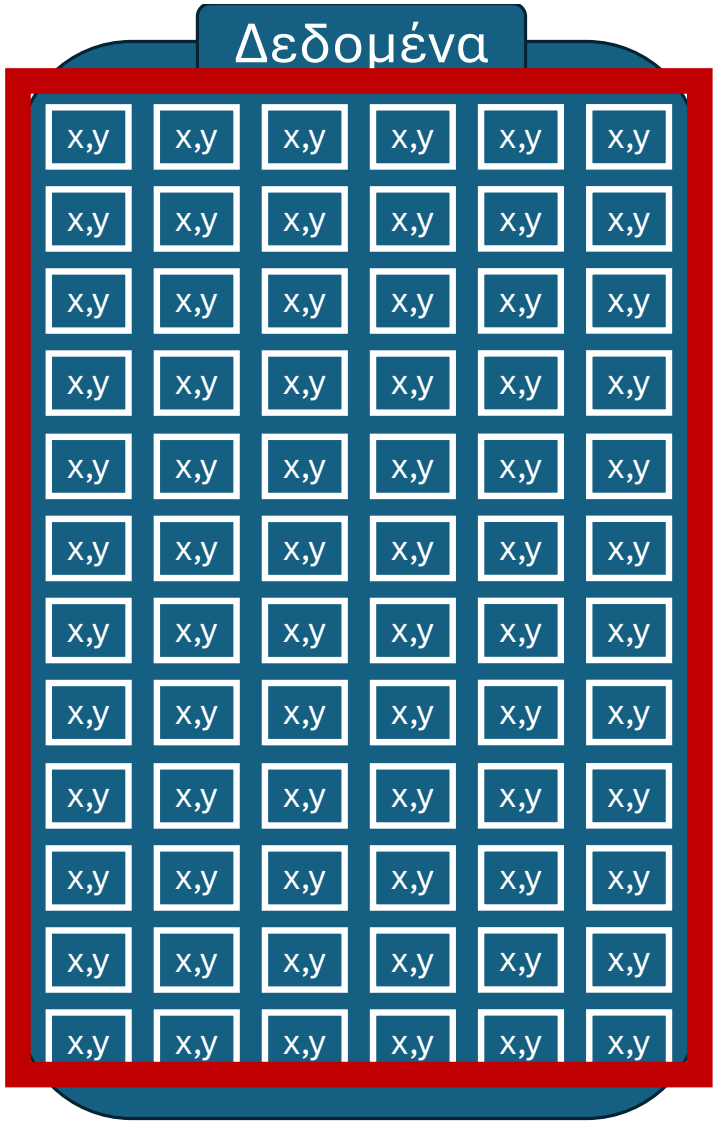
Νευρωνικό



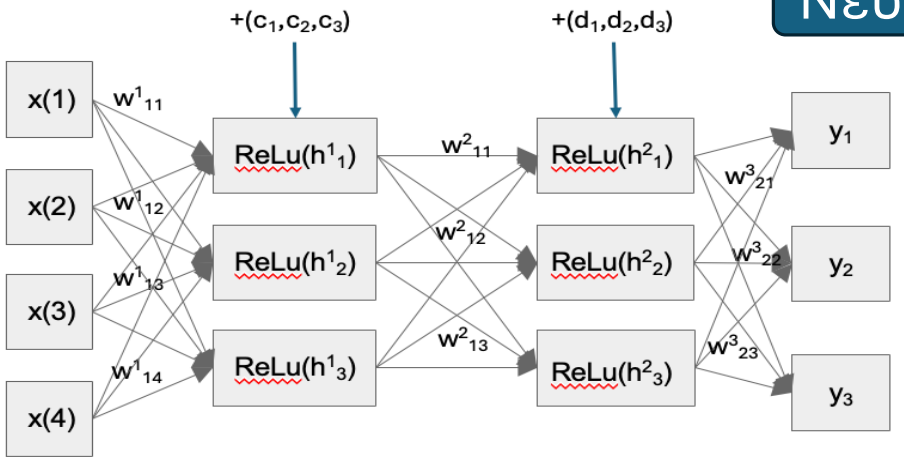
$$\theta = (W^1, c, W^2, d, W^3)$$

$$Loss(y_i, x_i, \theta) = Loss(y_i, x_i, W^1, c, W^2, d, W^3)$$

Απώλεια σε ένα ζευγάρι δεδομένων



Νευρωνικό



$$\theta = (W^1, c, W^2, d, W^3)$$

$$Loss(y_i, x_i, \theta) = Loss(y_i, x_i, W^1, c, W^2, d, W^3)$$

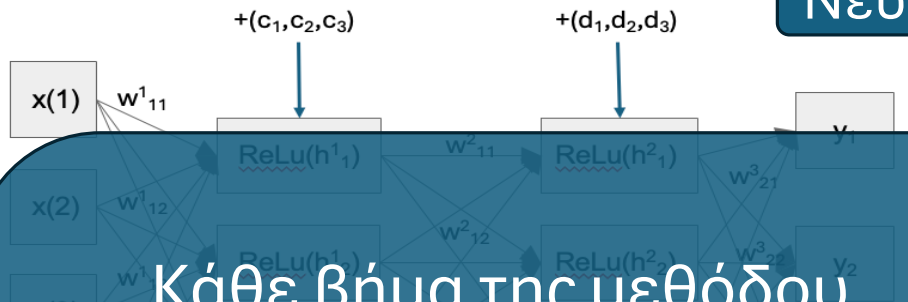
$$\mathrm{L}(\theta) = \sum Loss(y_i, x_i, \theta)$$

Συνολική απώλεια

Δεδομένα

x,y

Νευρωνικό



Κάθε βήμα της μεθόδου απαιτεί **άθροισμα πάνω σε όλα τα δεδομένα:**

$$\theta_+ = \theta - \eta \nabla L(\theta)$$

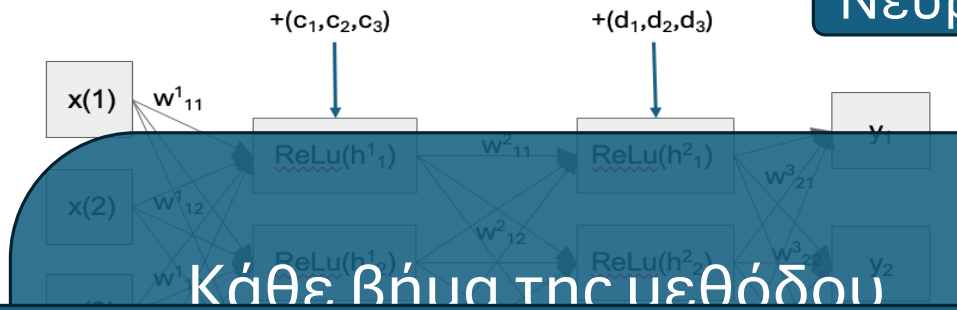
$$= \theta - \eta \sum_i \nabla Loss(x_i, y_i, \theta)$$

Συνολική απώλεια

Δεδομένα

x,y

Νευρωνικό



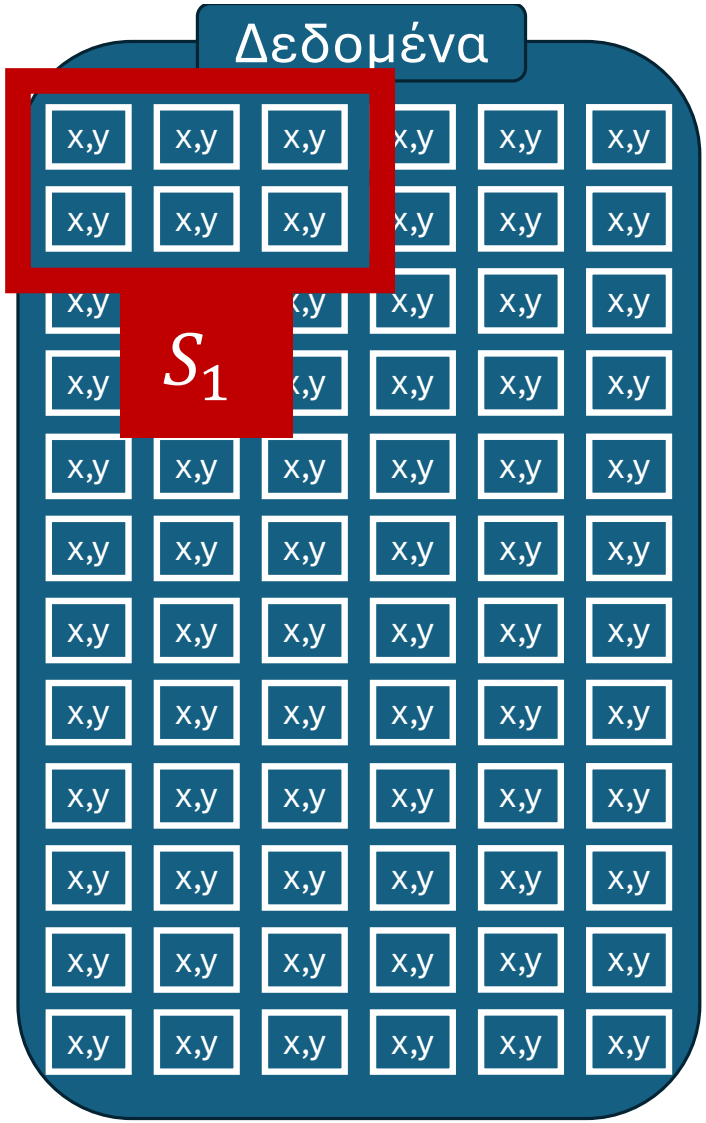
Κάθε βήμα της μεθόδου

Gradient Descent + Minibatch

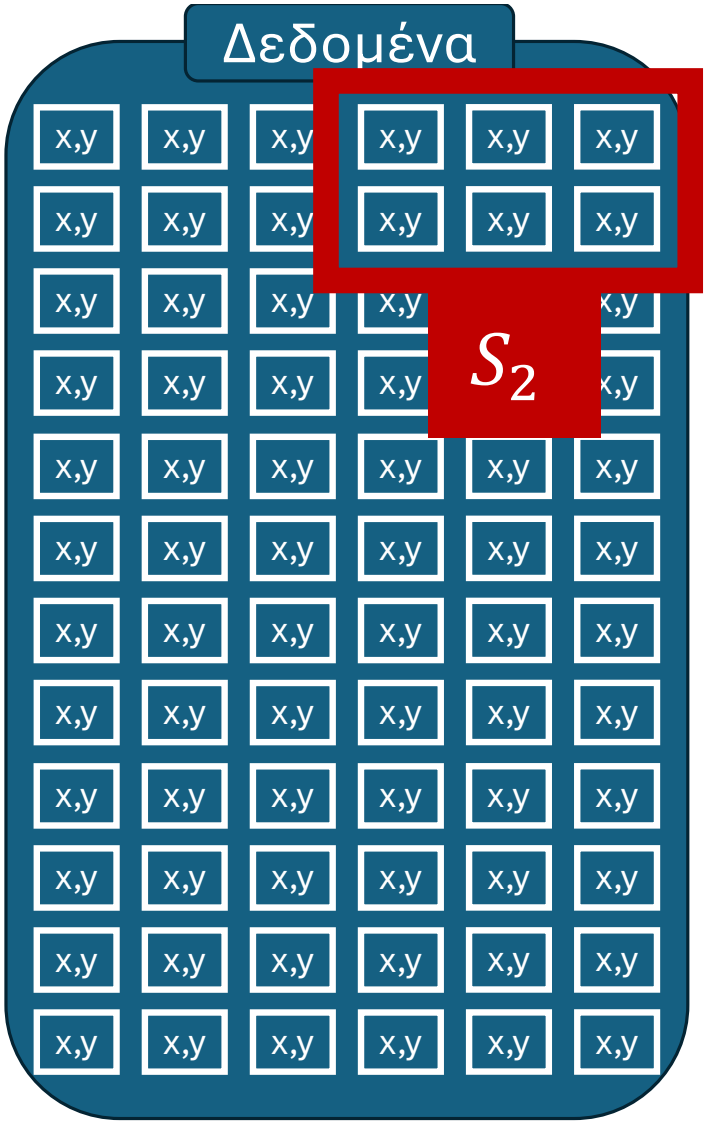
(batch_size)

→ Stochastic Gradient Descent

Συνολική απώλεια

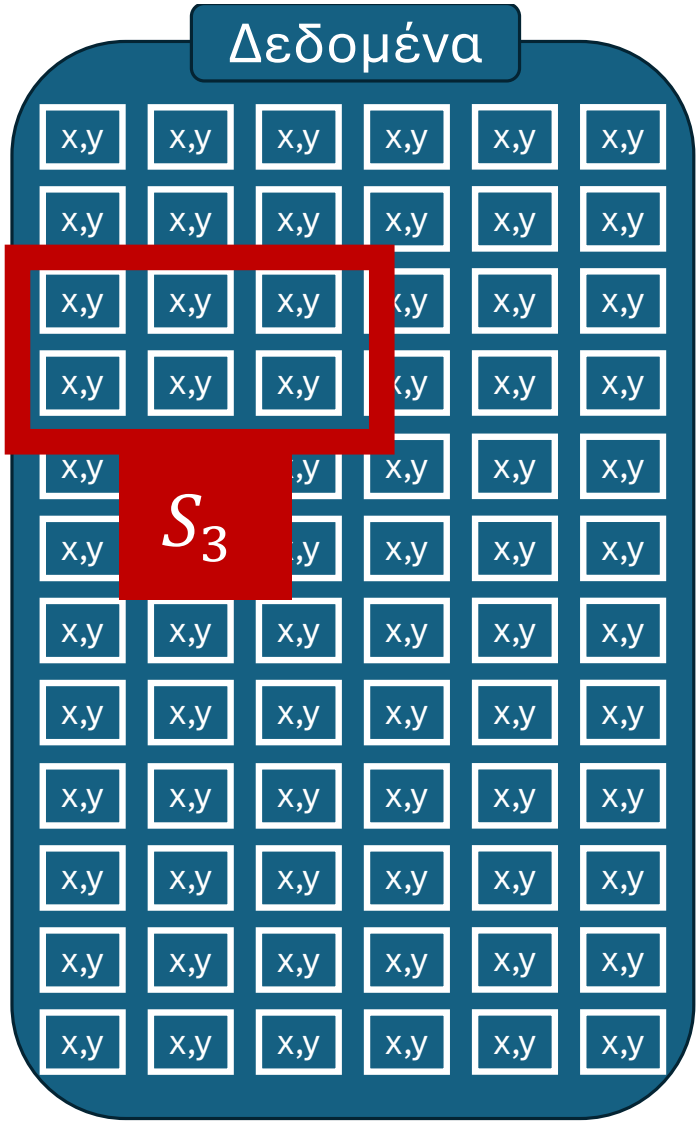


$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S_1} \nabla Loss(x_i, y_i, \theta_t)$$



$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S_1} \nabla Loss(x_i, y_i, \theta_t)$$

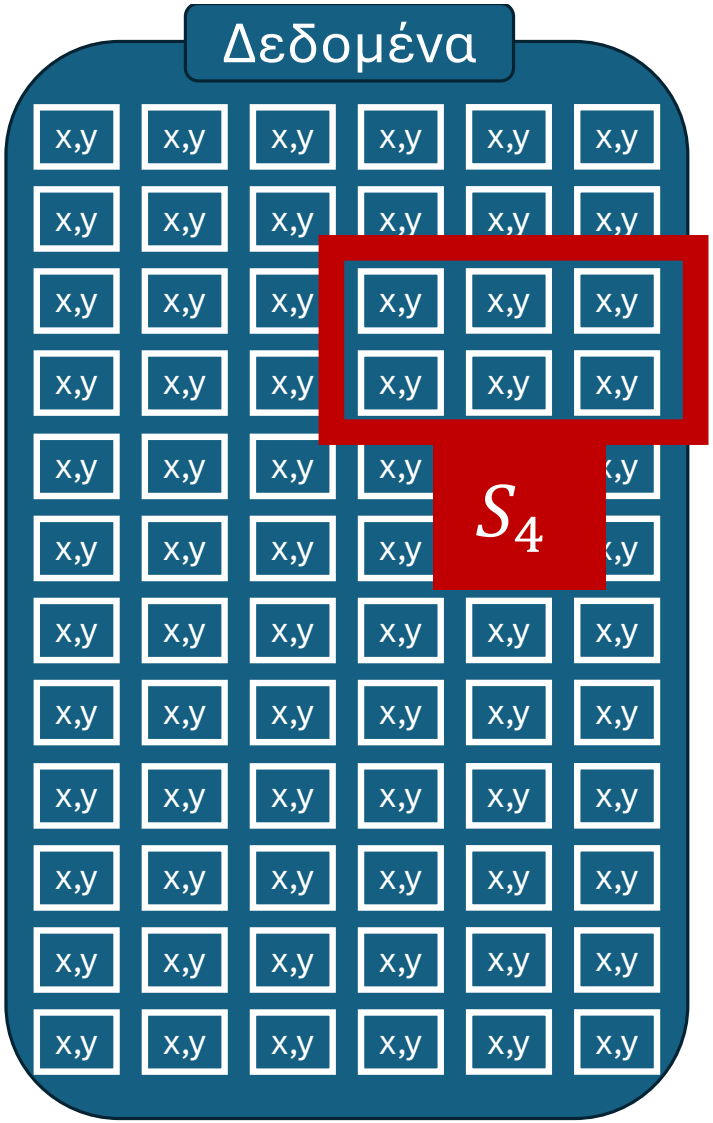
$$\theta_{t+2} = \theta_{t+1} - \eta \sum_{i \in S_2} \nabla Loss(x_i, y_i, \theta_{t+1})$$



$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S_1} \nabla Loss(x_i, y_i, \theta_t)$$

$$\theta_{t+2} = \theta_{t+1} - \eta \sum_{i \in S_2} \nabla Loss(x_i, y_i, \theta_{t+1})$$

$$\theta_{t+3} = \theta_{t+2} - \eta \sum_{i \in S_3} \nabla Loss(x_i, y_i, \theta_{t+2})$$



$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S_1} \nabla Loss(x_i, y_i, \theta_t)$$

$$\theta_{t+2} = \theta_{t+1} - \eta \sum_{i \in S_2} \nabla Loss(x_i, y_i, \theta_{t+1})$$

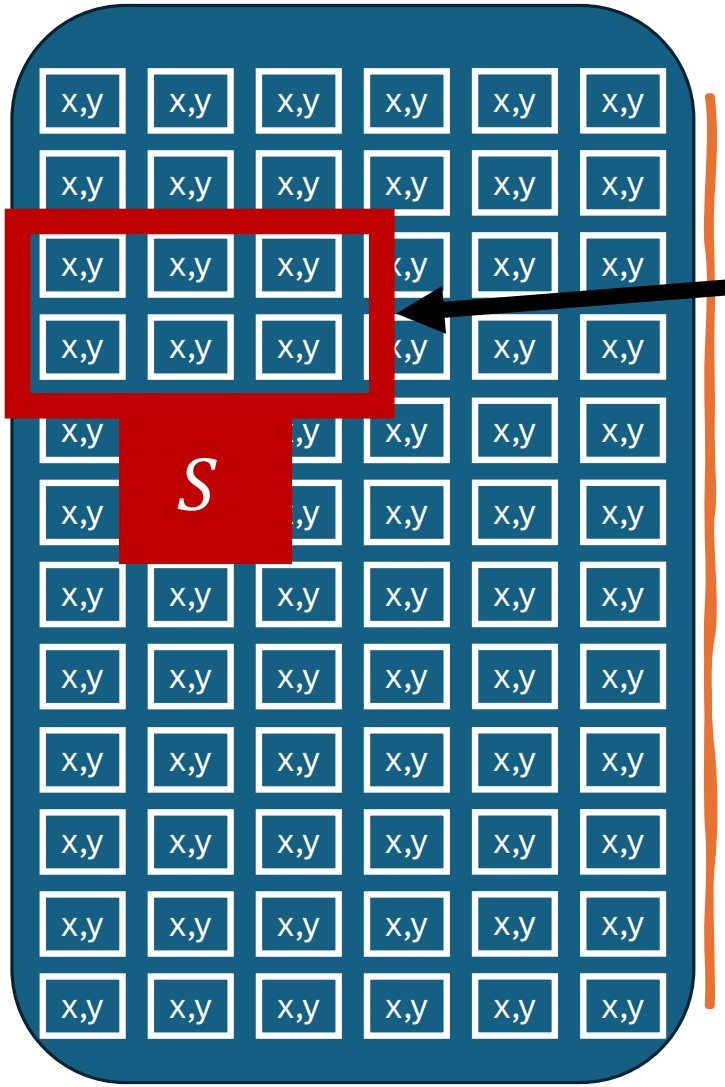
$$\theta_{t+3} = \theta_{t+2} - \eta \sum_{i \in S_3} \nabla Loss(x_i, y_i, \theta_{t+2})$$

$$\theta_{t+4} = \theta_{t+3} - \eta \sum_{i \in S_3} \nabla Loss(x_i, y_i, \theta_{t+3})$$

Η Μέθοδος *Stochastic* Gradient Descent σε κώδικα Python

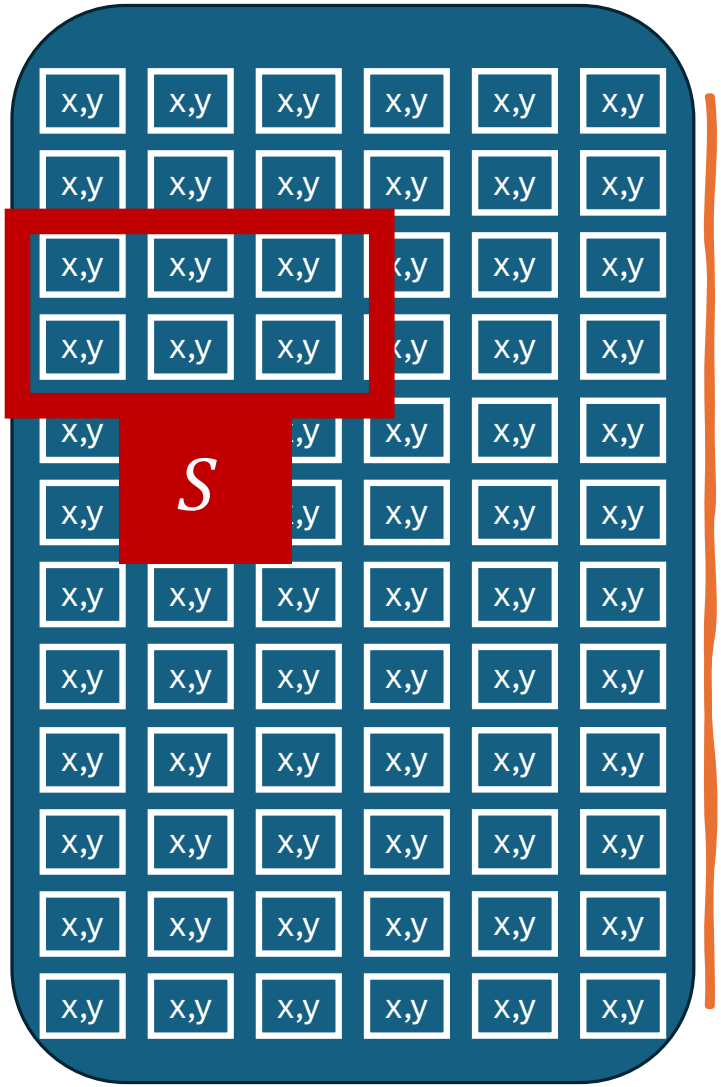
```python
# dataloaders
train_loader = DataLoader(train_dataset, batch_size=64,
                          shuffle=True, num_workers=2)
```

```python
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```




```
# dataloaders
train_loader = DataLoader(train_dataset, batch_size=64,
                          shuffle=True, num_workers=2)
```

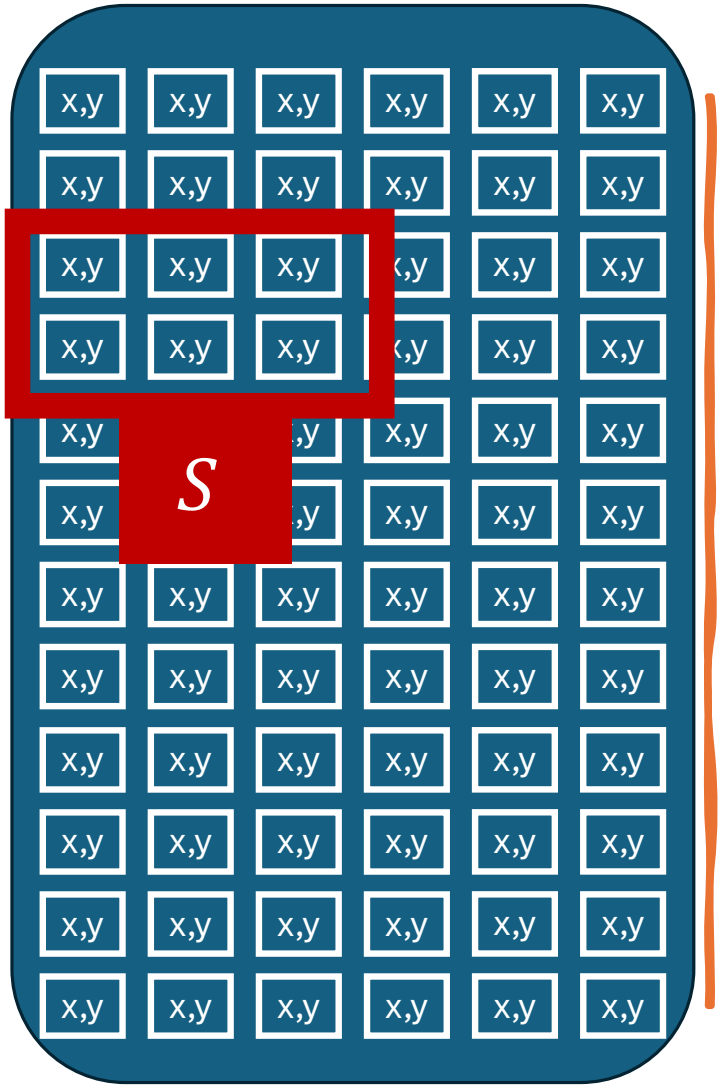
```
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```




```
# dataloaders
train_loader = DataLoader(train_dataset, batch_size=64,
                          shuffle=True, num_workers=2)
```

```
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

$$\sum_{i \in S} \nabla Loss(x_i, y_i, \theta_t)$$




```python
# dataloaders
train_loader = DataLoader(train_dataset, batch_size=64,
                          shuffle=True, num_workers=2)
```

```python
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

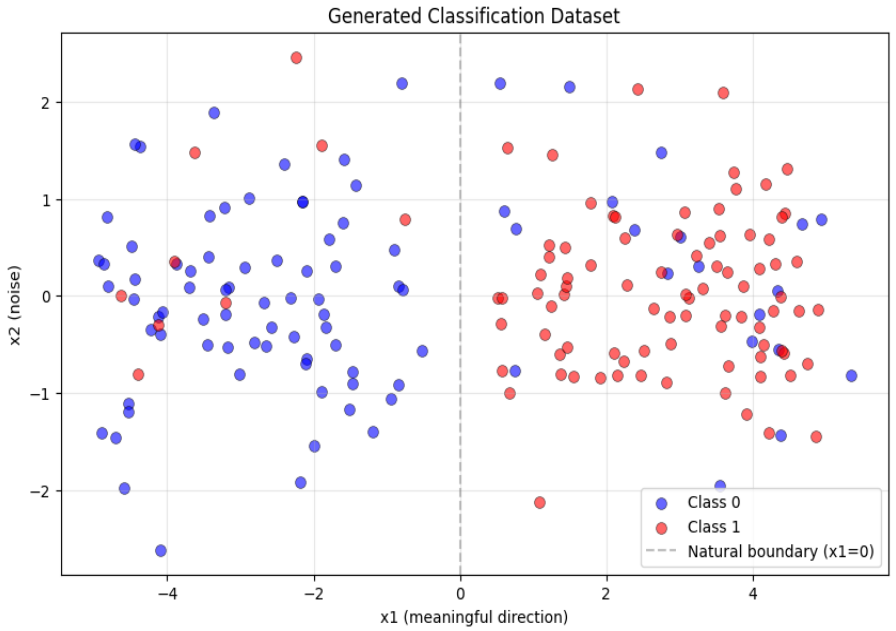
$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S} \nabla Loss(x_i, y_i, \theta_t)$$

Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των δεδομένων.

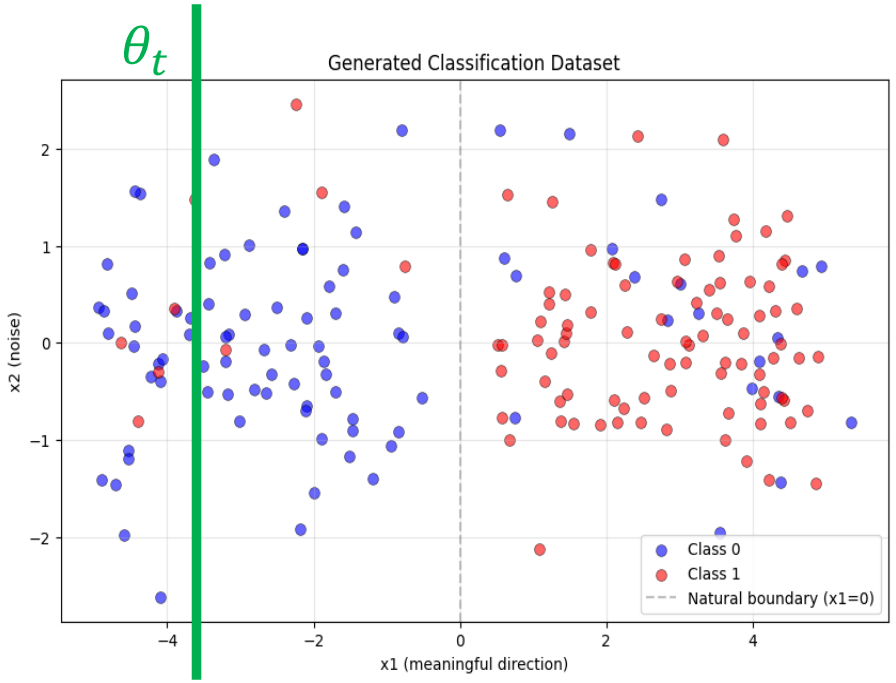


Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των **παραμέτρων** και των δεδομένων.

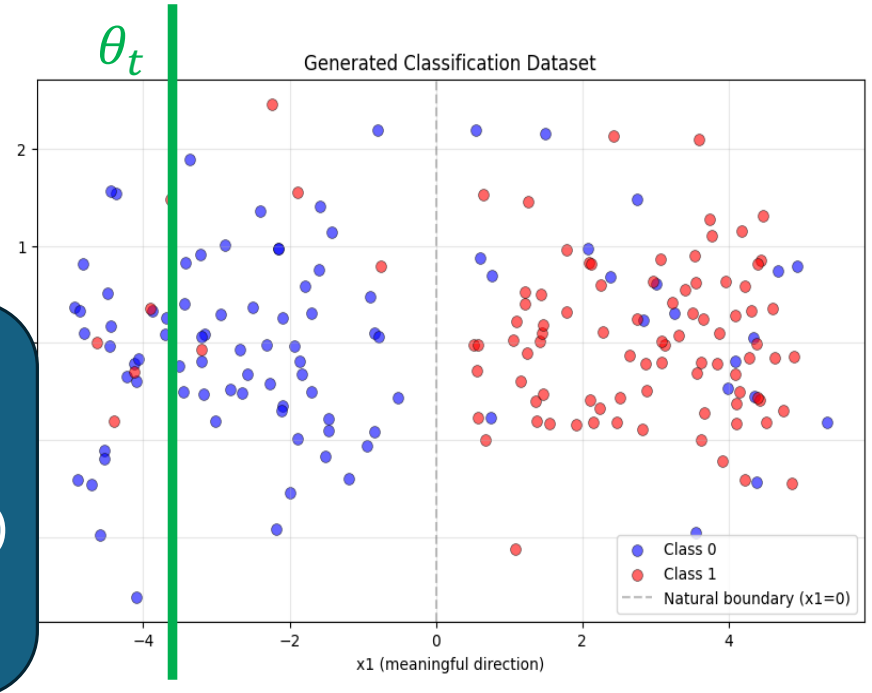


Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των **παραμέτρων** και των δεδομένων.



Πού είναι το θ_{t+1};

$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S} \nabla Loss(x_i, y_i, \theta_t)$$

Η Μέθοδος: Stochastic Gradient Descent

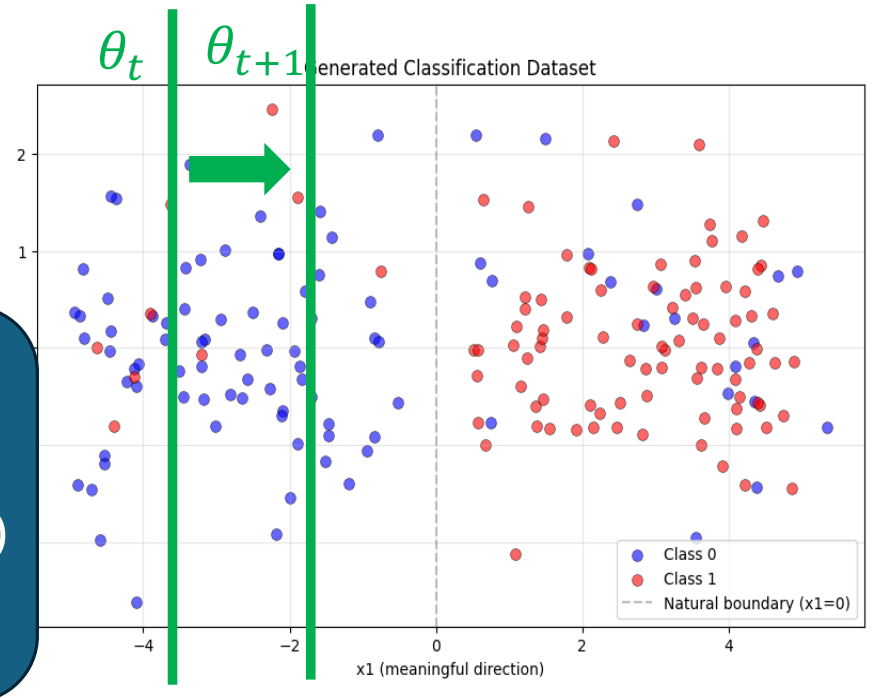
Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των **παραμέτρων** και των δεδομένων.

Πού είναι το θ_{t+1};

$$\theta_{t+1} = \theta_t - \eta \sum_{i \in S} \nabla Loss(x_i, y_i, \theta_t)$$

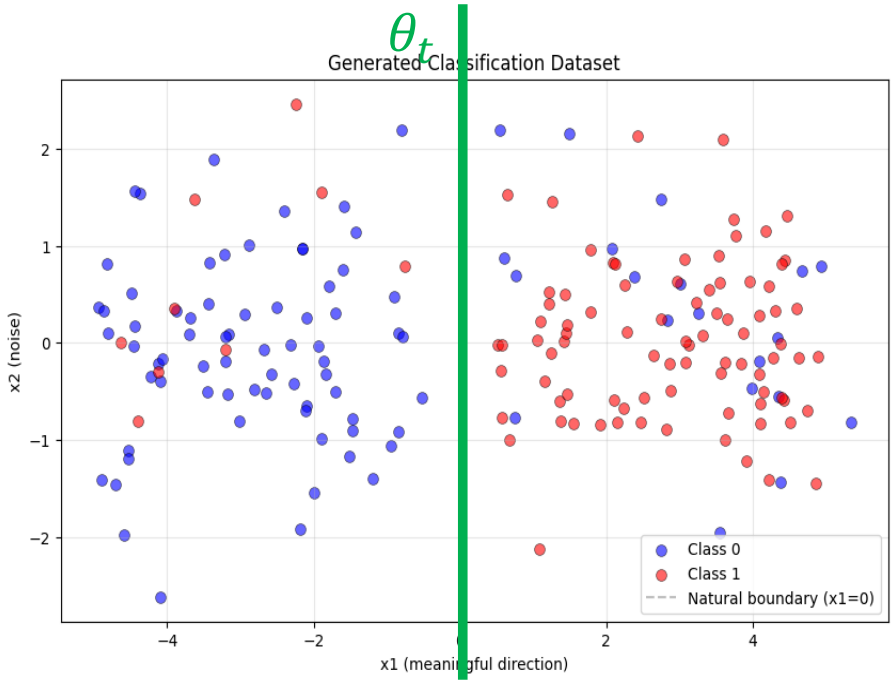


Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.

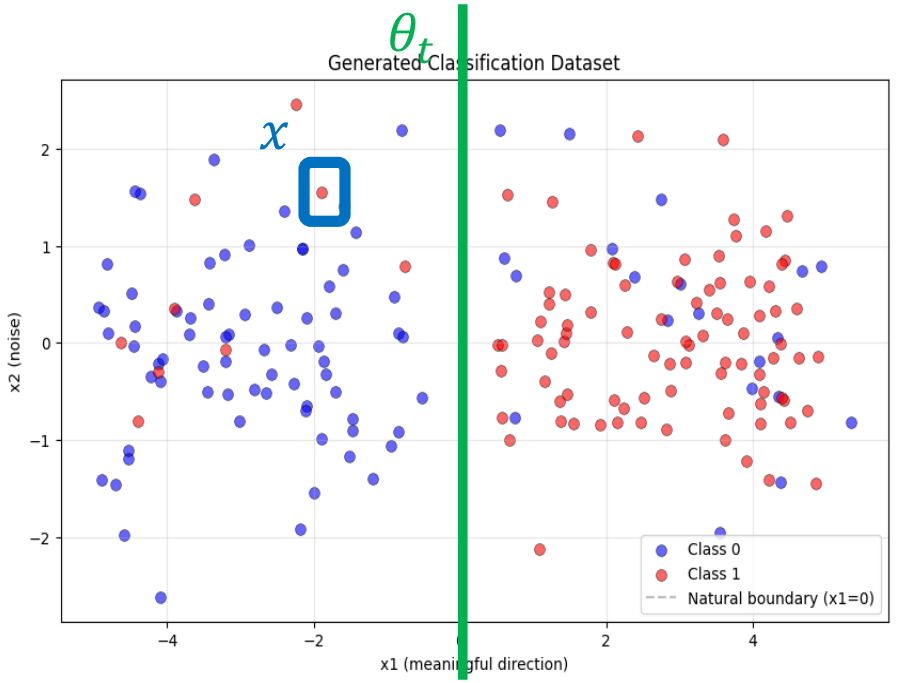


Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η Μέθοδος: Stochastic Gradient Descent

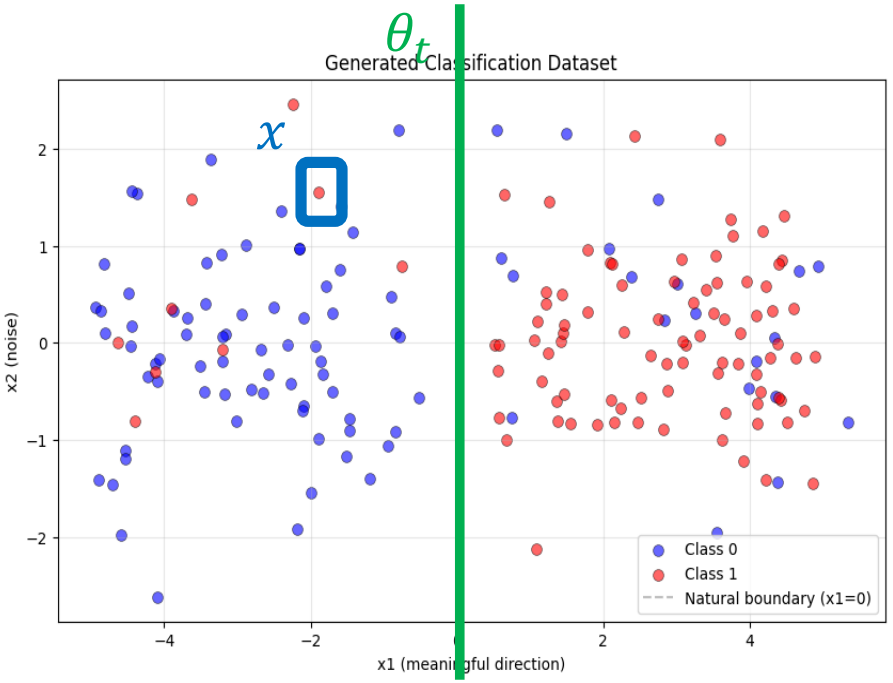
Πού είναι το x_+;

$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η Μέθοδος: Stochastic Gradient Descent

Πού είναι το x_+;

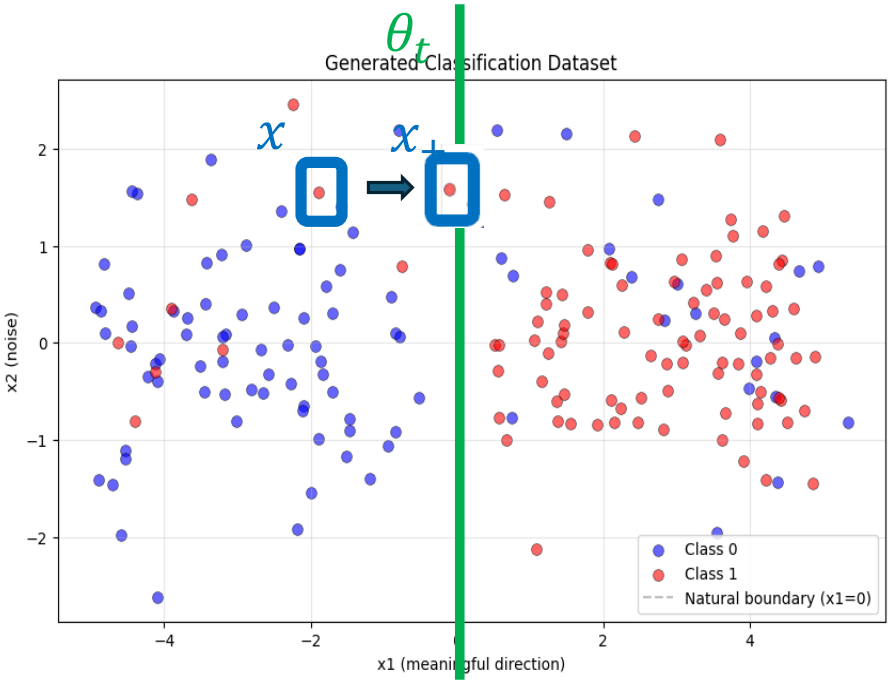
$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Μειώνουμε την απώλεια

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.

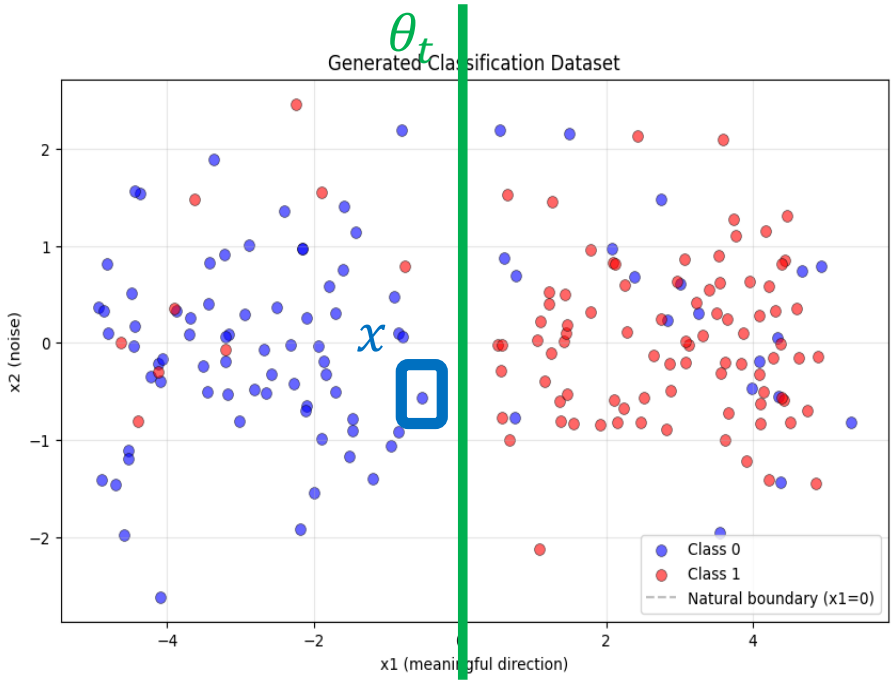


Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



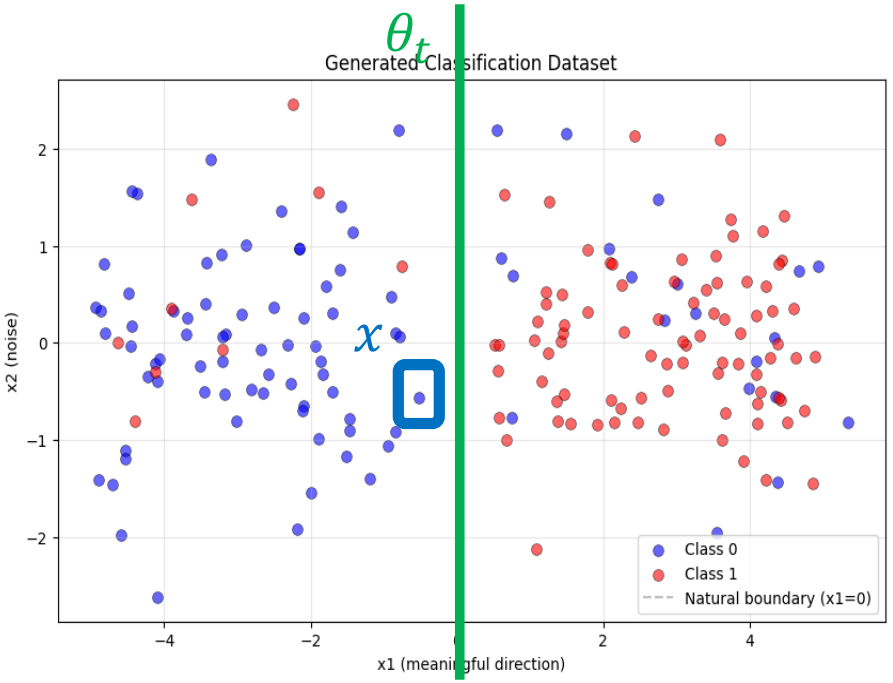
Η Μέθοδος: Stochastic Gradient Descent

Τι πρέπει να κάνουμε για να **αυξήσουμε** την απώλεια του σημείου;

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η Μέθοδος: Stochastic Gradient Descent

Αυξάνουμε την απώλεια

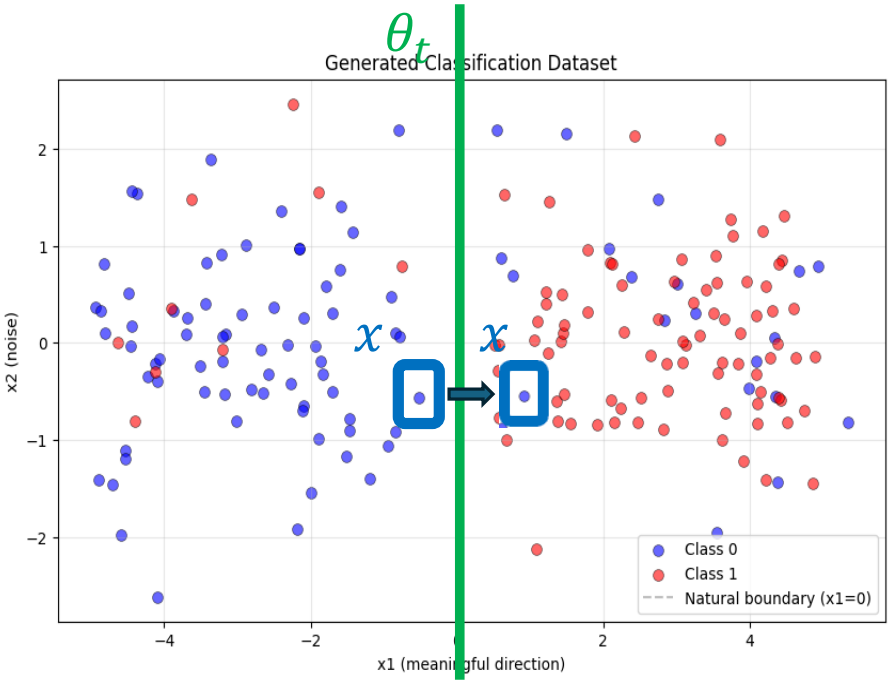
$$x_+ = x + \eta \nabla_x Loss(x, y, \theta_t)$$

Ακολουθούμε την κλίση

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η Μέθοδος: Stochastic Gradient Descent

Μειώνουμε
~~Αυξάνουμε~~ την απώλεια

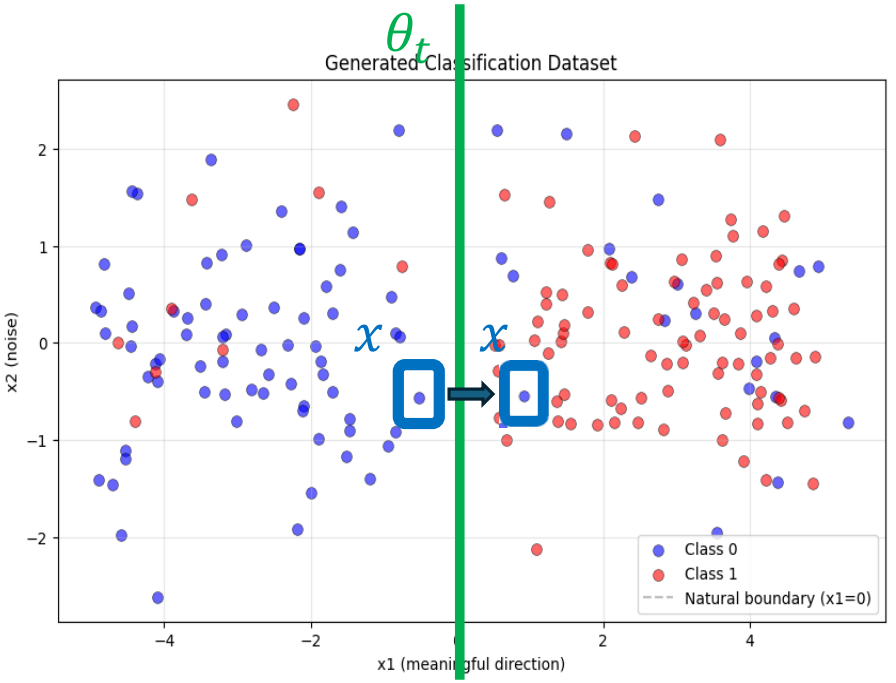
$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Ακολουθούμε την αρνητική κλίση

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η Μέθοδος: Stochastic Gradient Descent

Αλλάζουμε το y

Μειώνουμε ~~Αυξάνουμε~~ την απώλεια

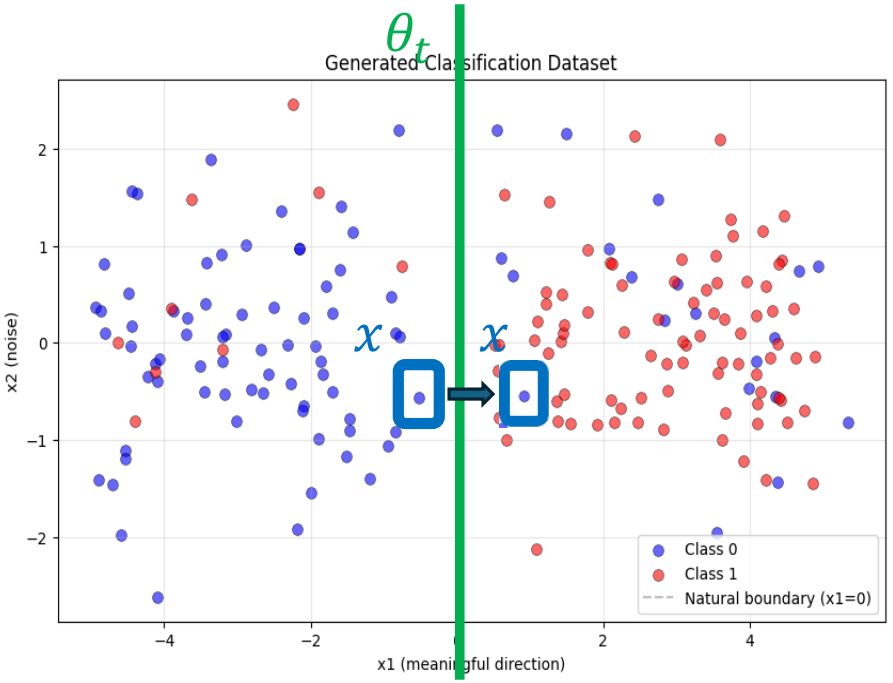
$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Ακολουθούμε την αρνητική κλίση

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.



Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.

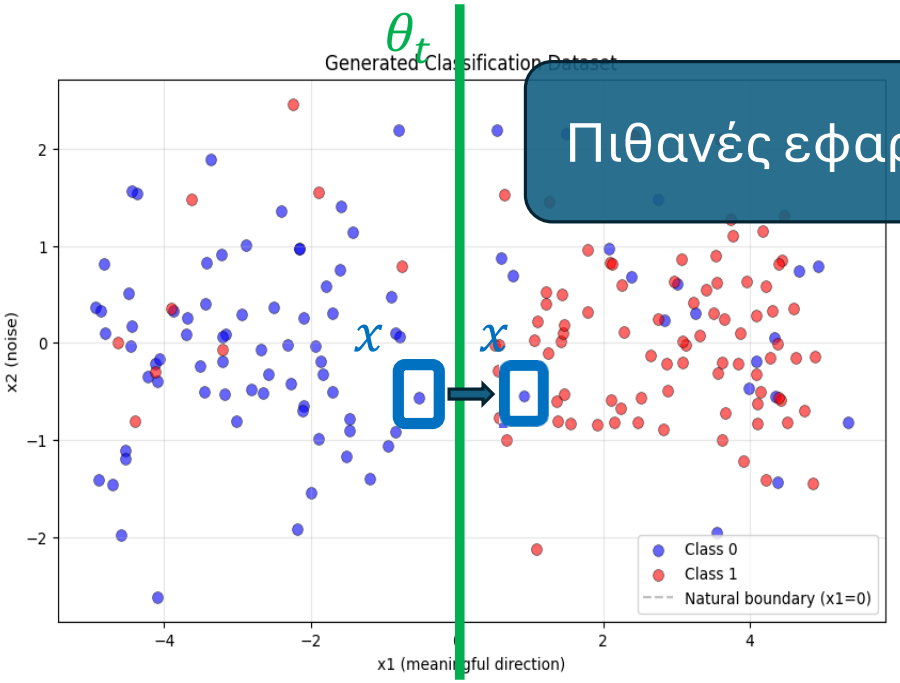
Η Μέθοδος: Stochastic Gradient Descent

Αλλάζουμε το y

Μειώνουμε ~~**Αυξάνουμε**~~ **την απώλεια**

$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Ακολουθούμε την αρνητική κλίση



Πιθανές εφαρμογές;

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των **δεδομένων**.

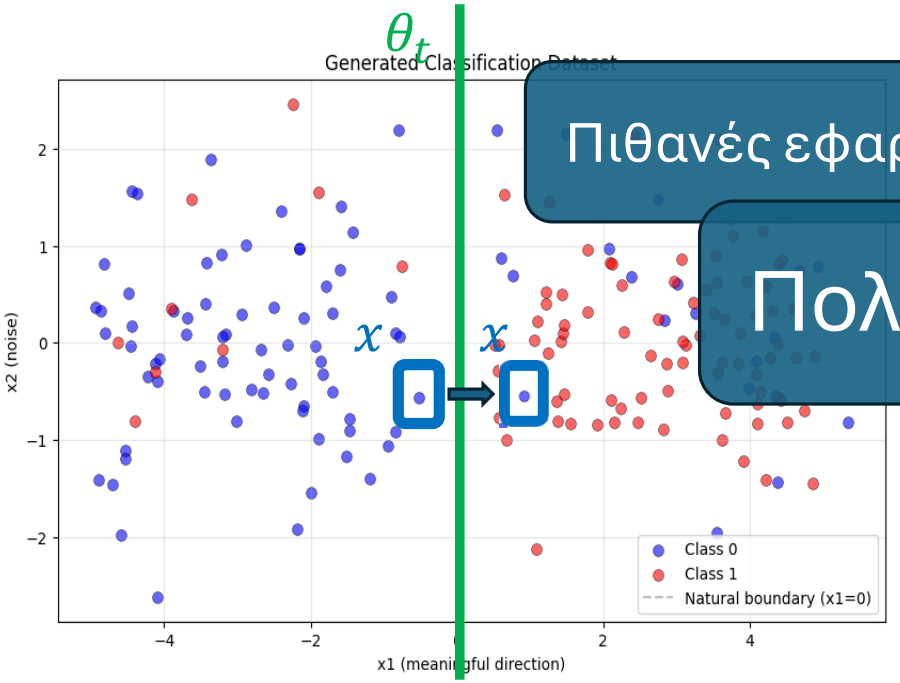
Η Μέθοδος: Stochastic Gradient Descent

Αλλάζουμε το y

Μειώνουμε ~~Αυξάνουμε~~ την απώλεια

$$x_+ = x - \eta \nabla_x Loss(x, y, \theta_t)$$

Ακολουθούμε την αρνητική κλίση



Πιθανές εφαρμογές;

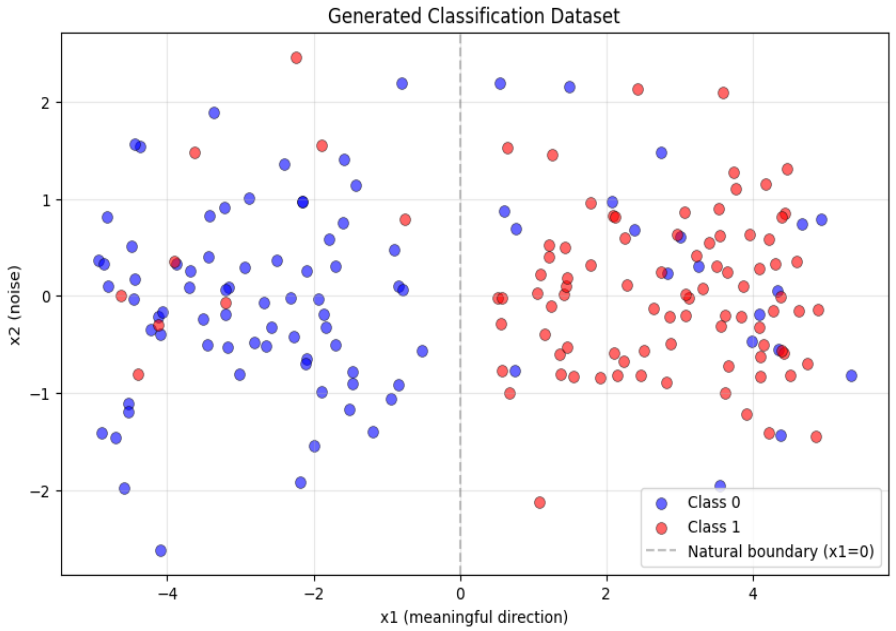
Πολλές!

Η Μέθοδος: Stochastic Gradient Descent

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των δεδομένων.



Όταν **εκπαιδεύουμε** βρίσκουμε παραμέτρους που ελαχιστοποιούν την απώλεια

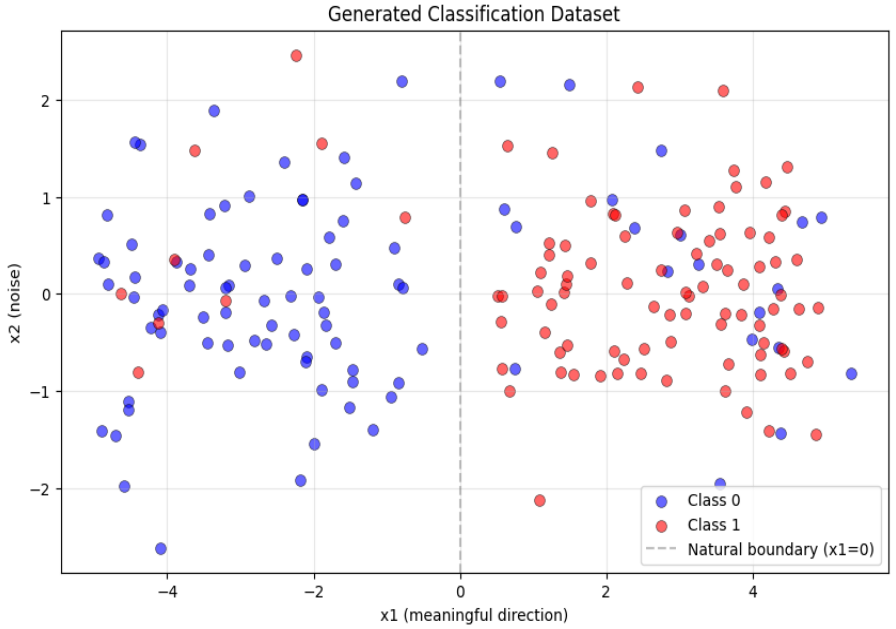
Stochastic Gradient Descent

$$\theta_+ = \theta - \eta \sum_{i \in S} \nabla Loss(x_i, y_i, \theta)$$

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των δεδομένων.



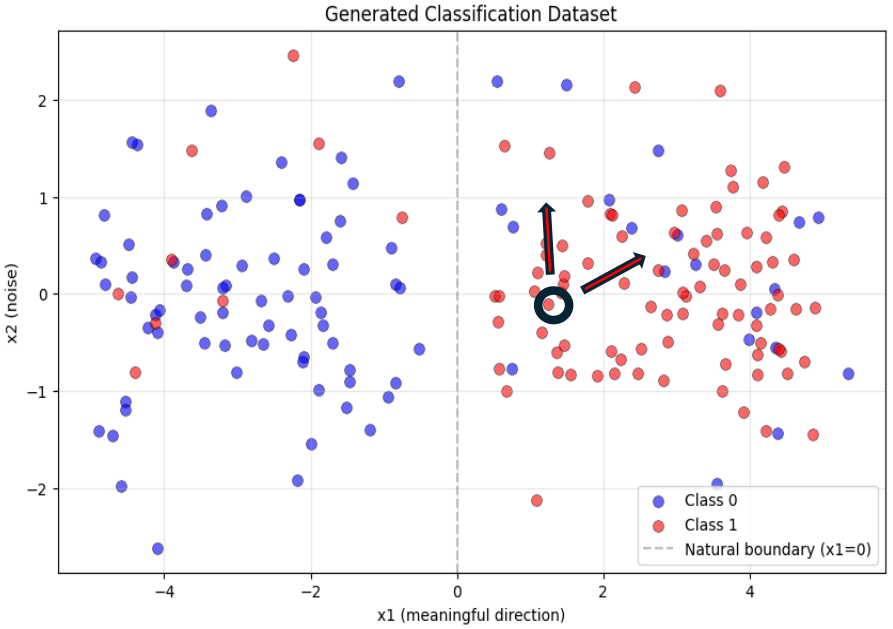
Πότε θα θέλαμε να μετακινήσουμε ένα σημείο, ώστε να αυξήσουμε ή να μειώσουμε την απώλειά του, χωρίς να πειράξουμε ή να αλλάξουμε το νευρωνικό δίκτυο;

$$x_+ = x \pm \eta \nabla_x Loss(x, \theta)$$

Η συνολική απώλεια:

$$\sum Loss(x_i, y_i, \theta)$$

Είναι συνάρτηση των παραμέτρων και των δεδομένων.



Η συνολική απώλεια:

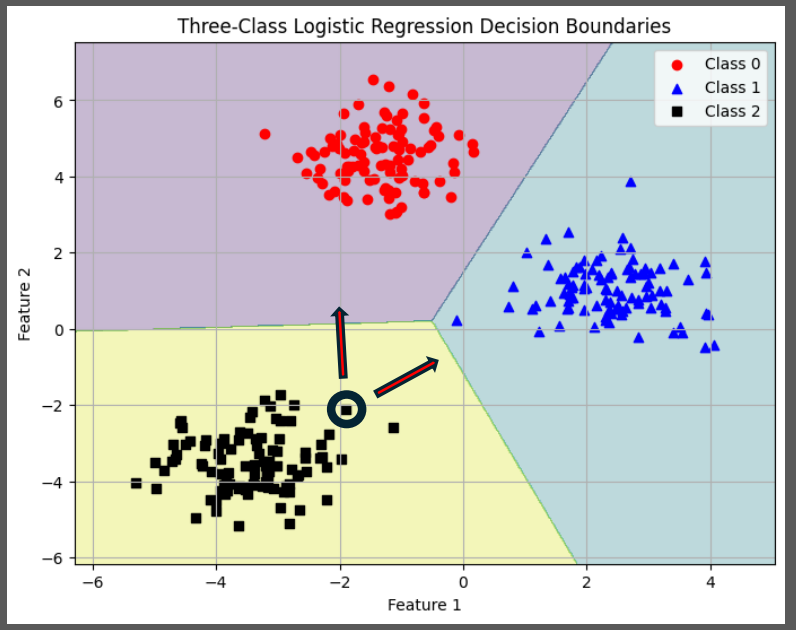
Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

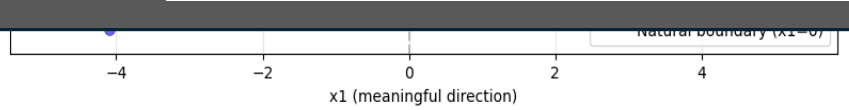
“Adversarial Attacks”

x – μαύρο σημείο

$$x^+ = x - \eta \nabla_x L(x, y, \theta)$$

$$x^+ = x - \eta \nabla_x L(x, y, \theta)$$



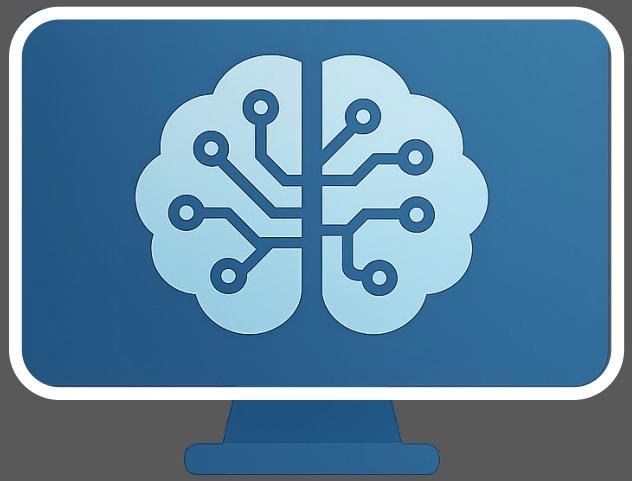


Η συνολική απώλεια:

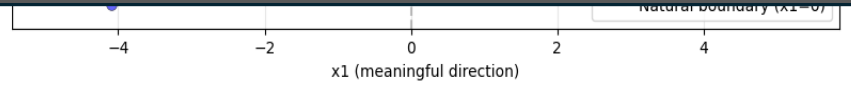
Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

“Adversarial Attacks”

resnet18



Golden retriever: 50%
Dog: 99.96%

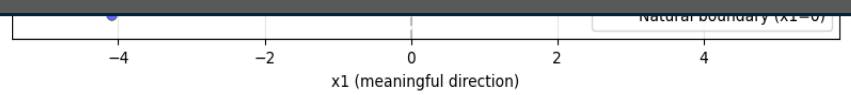


Η συνολική απώλεια:

Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

“Adversarial Attacks”

$$- \eta \nabla_x L(x_{\text{dog}}, y_{\text{cat}}, \theta)$$

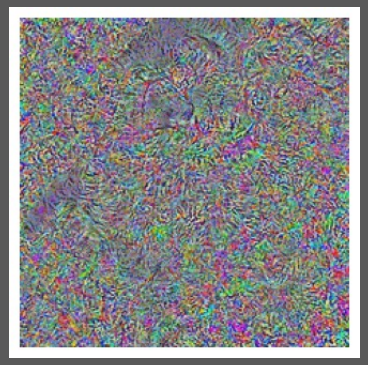


Η συνολική απώλεια:

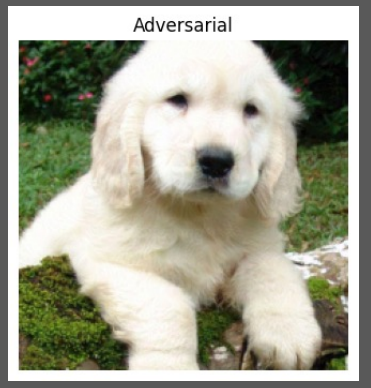
Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

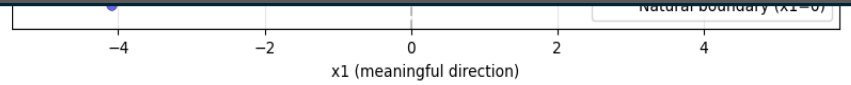
“Adversarial Attacks”

$$-\ \eta$$



$$=$$



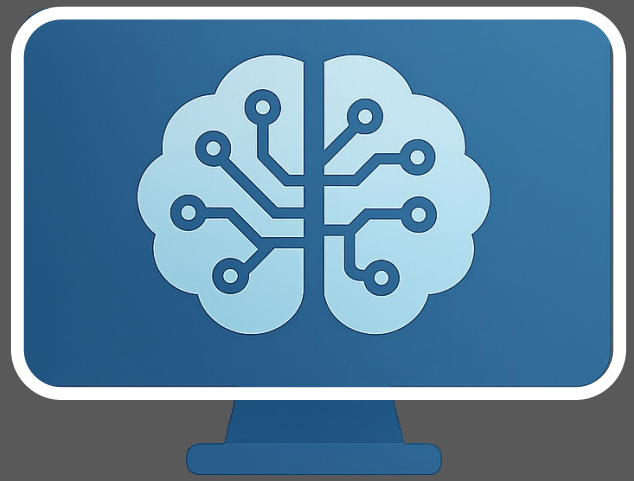


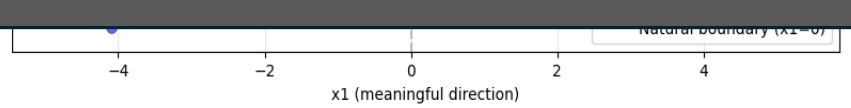
Η συνολική απώλεια:

Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

“Adversarial Attacks”

resnet18



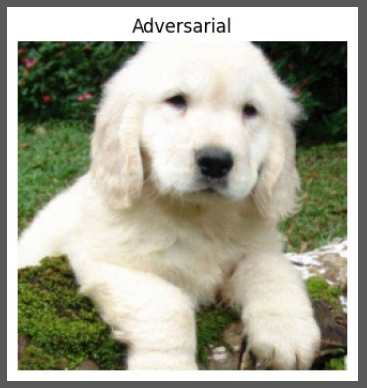


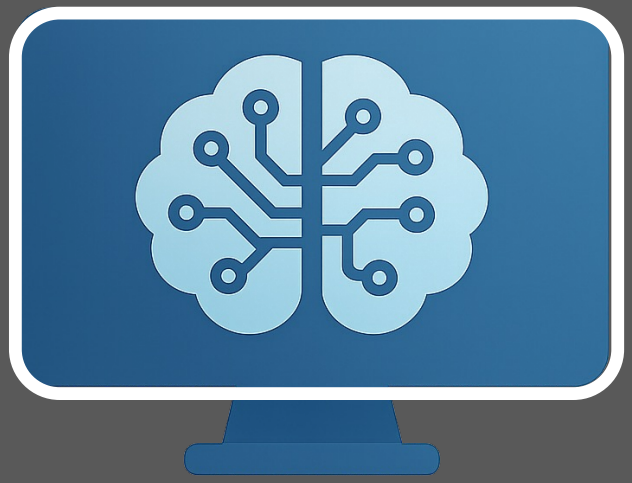
Η συνολική απώλεια:

Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

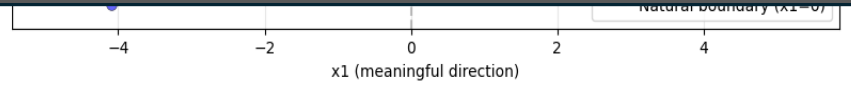
“Adversarial Attacks”

resnet18





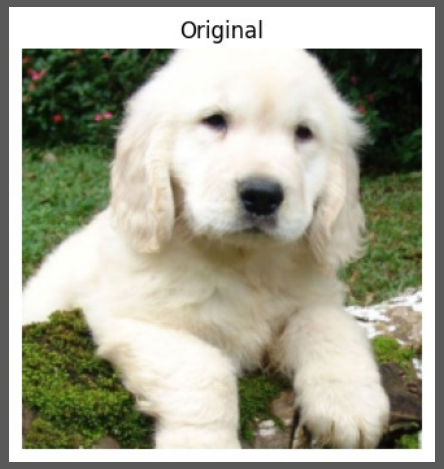
Siamese Cat: 100%

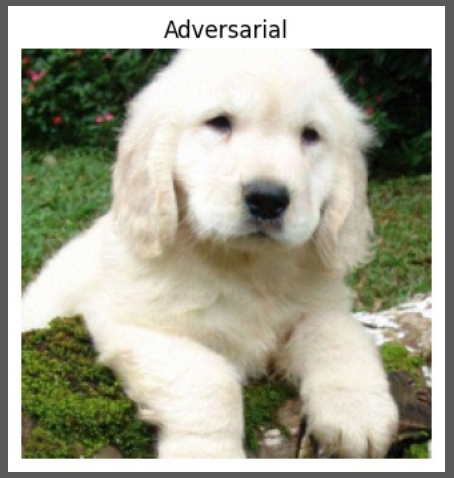


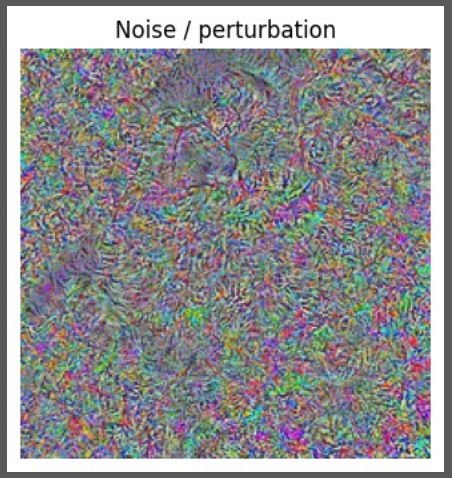
Η συνολική απώλεια:

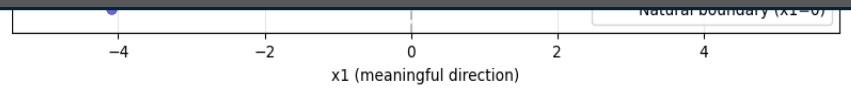
Κακόβουλη επίθεση ενάντια στο νευρωνικό δίκτυο

“Adversarial Attacks”









Εκπαίδευση: Αλλάζουμε το μοντέλο (θ)

Stochastic Gradient Descent

$$\theta_+ = \theta - \eta \sum_{i \in S} \nabla Loss(x_i, y_i, \theta)$$

Επίθεση: Αλλάζουμε το σημείο (x)

$$x_+ = x \pm \eta \nabla_x Loss(x, y, \theta)$$