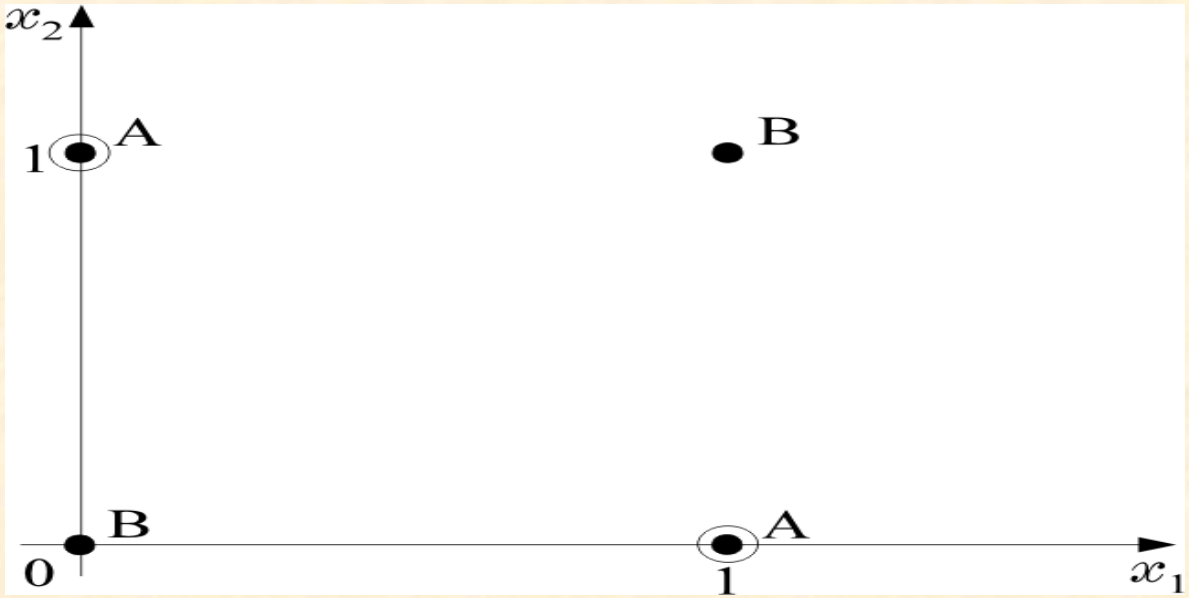
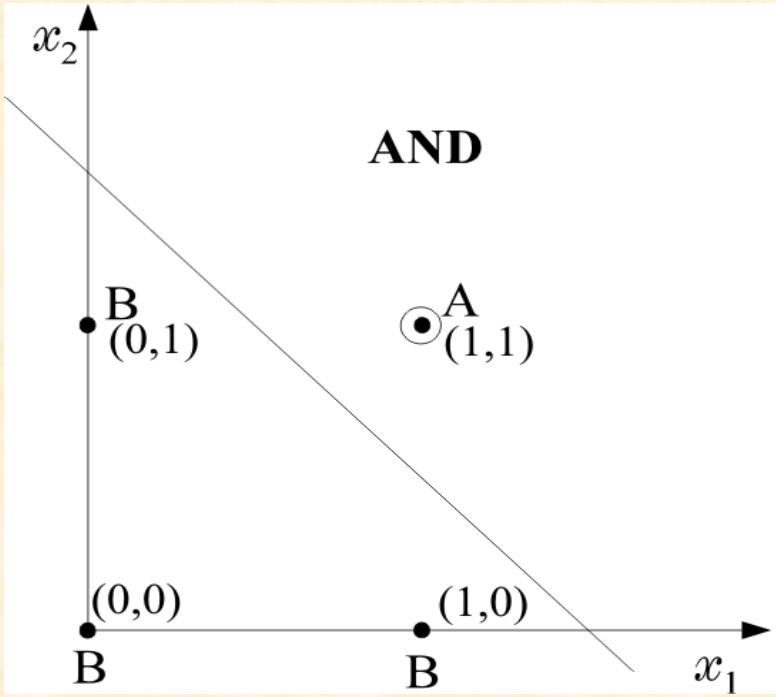
Μη γραμμικοί ταξινομητές

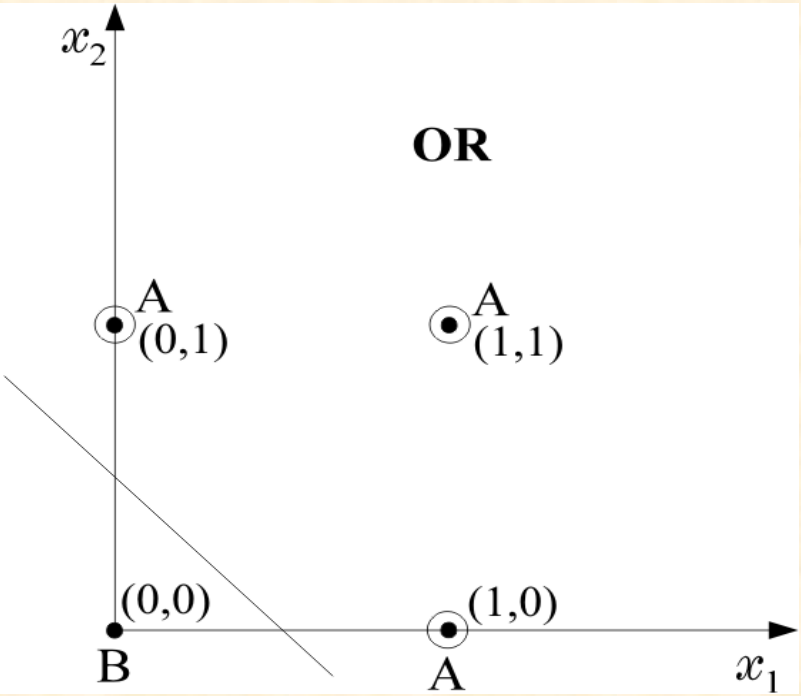
- Το πρόβλημα XOR

x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B



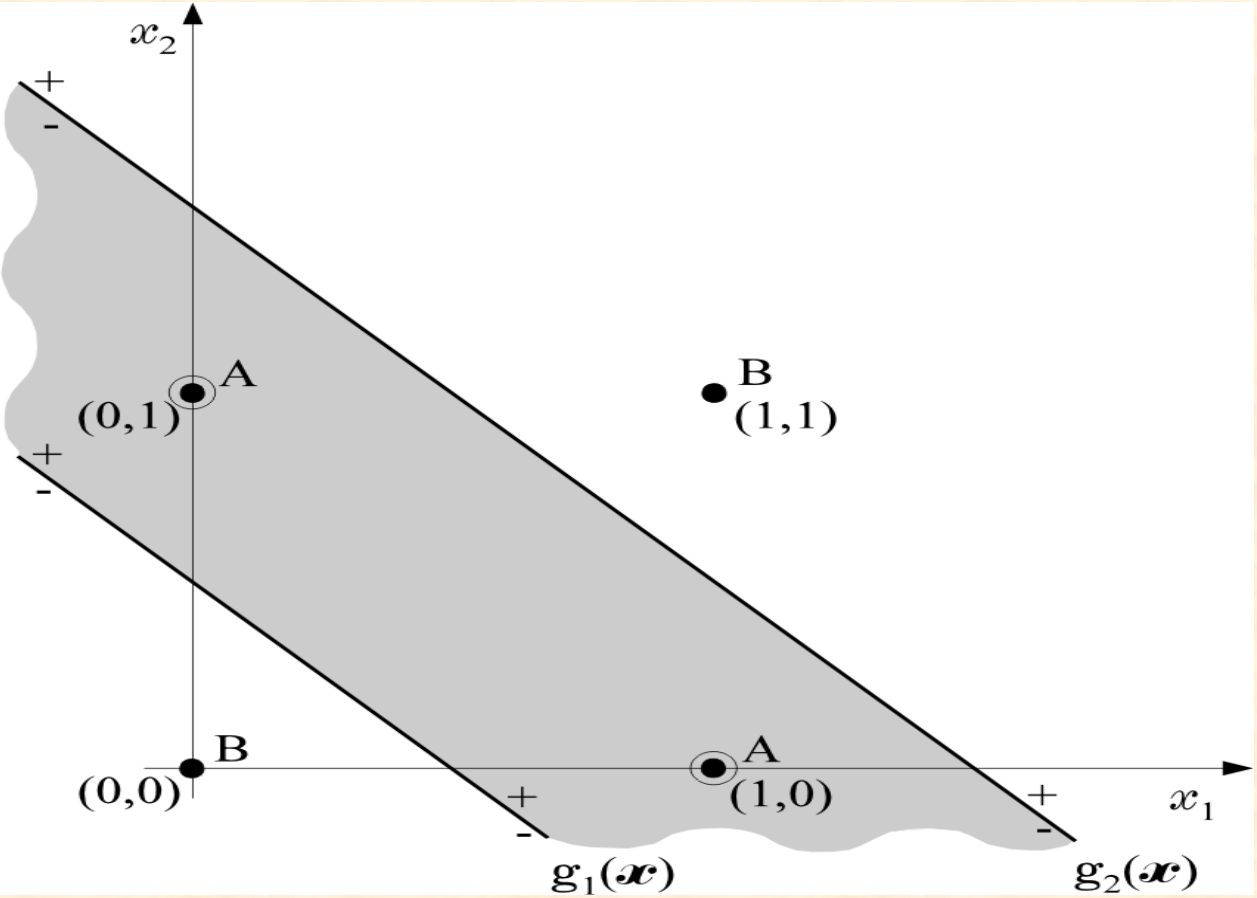
- Δεν υπάρχει μία γραμμή (υπερεπίπεδο) που να διαχωρίζει την κλάση Α από την κλάση Β. Αντίθετα, οι λειτουργίες AND και OR είναι γραμμικώς διαχωρίσιμα προβλήματα





❖ Το δι-επίπεδο (Two-Layer) Perceptron

- Για το πρόβλημα XOR, ζωγράφισε δύο, αντί, για μία γραμμή



- Τότε, η κλάση Β βρίσκεται εκτός της σκιασμένης περιοχής ενώ η κλάση Α εντός αυτής. Πρόκειται για μία σχεδίαση δύο φάσεων (two-phase design).
  - Φάση 1: χάραξε δύο γραμμές (υπερεπίπεδα)

    $$g_1(\underline{x}) = g_2(\underline{x}) = 0$$

    Καθένα από αυτά υλοποιείται από ένα perceptron. Οι έξοδοι των perceptrons θα είναι

    $$y_i = f(g_i(\underline{x})) = \begin{cases} 0 \\ 1 \end{cases} \quad i = 1, 2$$

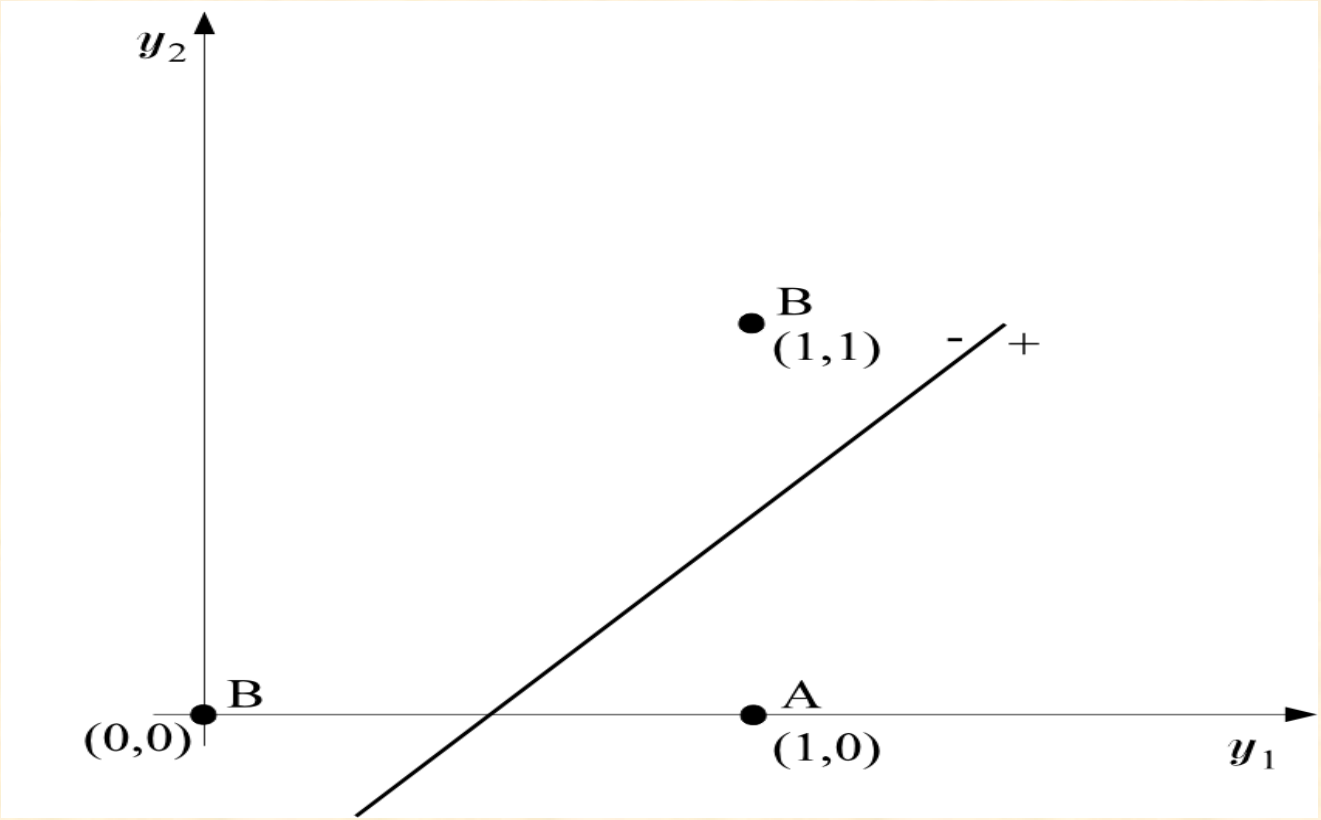
    ανάλογα με τη θέση του $\underline{x}$.

  - Φάση 2: Βρες τη θέση του $\underline{x}$ $\omega\varsigma$ $\pi\rho o\varsigma$ αμφότερες τις γραμμές, με βάση τις τιμές των y_1, y_2.

1st phase				2nd phase
x_1	x_2	y_1	y_2	
0	0	0(-)	0(-)	B(0)
0	1	1(+)	0(-)	A(1)
1	0	1(+)	0(-)	A(1)
1	1	1(+)	1(+)	B(0)

- Ισοδύναμα: Οι υπολογισμοί της πρώτης φάσης υλοποιούν μία απεικόνιση $\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$

Η απόφαση παίρνεται τώρα με βάση τα μετασχηματισμένα δεδομένα. $\underline{y}$

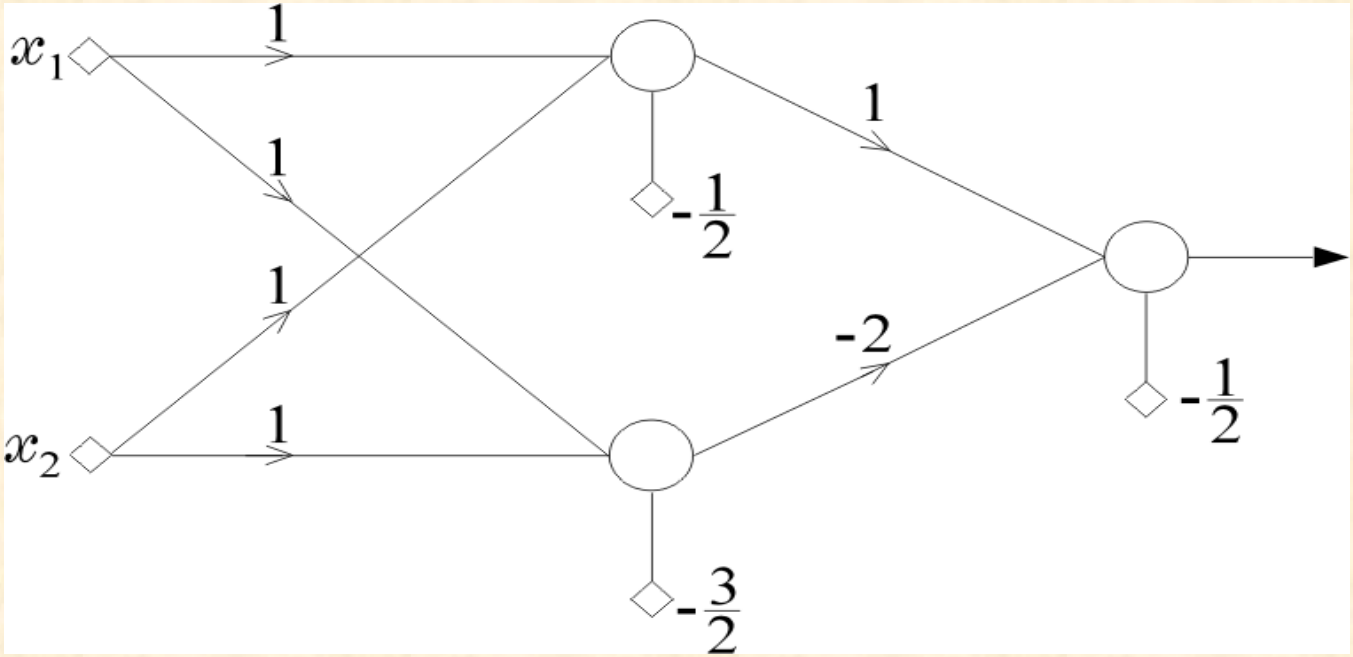


$g(\underline{y}) = 0$

Αυτό μπορεί να γίνει μέσω μίας δεύτερης γραμμής, η οποία μπορεί επίσης να υλοποιηθεί από ένα perceptron.

- Οι υπολογισμοί της πρώτης φάσης πραγματοποιούν μία απεικόνιση που μετασχηματίζει το μη γραμμικώς διαχωρίσιμο πρόβλημα σε ένα γραμμικώς διαχωρίσιμο.

- Η αρχιτεκτονική



- Αυτή είναι γνωστή ως perceptron δύο επιπέδων με ένα κρυφό (hidden) και ένα επίπεδο εξόδου (output layer). Οι συναρτήσεις ενεργοποίησης (activation functions) είναι

$$f(.) = \begin{cases} 0 \\ 1 \end{cases}$$

- Οι νευρώνες (κόμβοι) του σχήματος υλοποιούν τις ακόλουθες γραμμές (υπερέπίπεδα)

$$g_1(\underline{x}) = x_1 + x_2 - \frac{1}{2} = 0$$

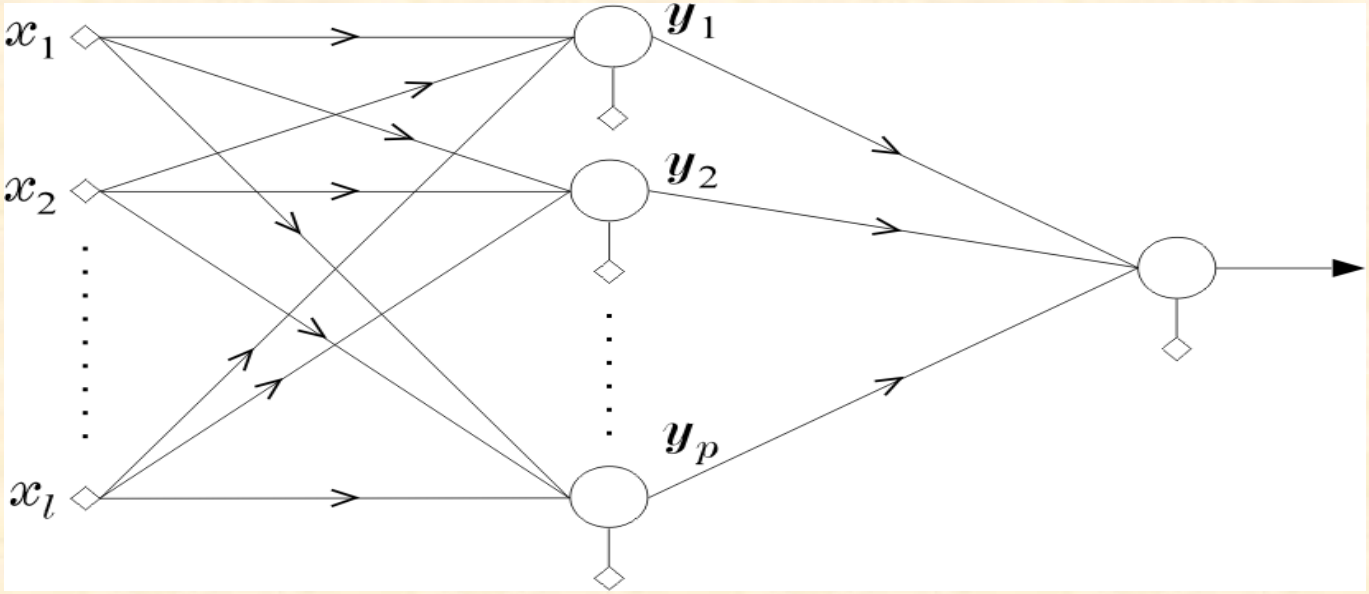
$$g_2(\underline{x}) = x_1 + x_2 - \frac{3}{2} = 0$$

$$g(\underline{y}) = y_1 - 2y_2 - \frac{1}{2} = 0$$

❖ Δυνατότητες ταξινόμησης δικτύου perceptron δύο επιπέδων

- Η απεικόνιση που πραγματοποιείται από τους νευρώνες του 1ου επιπέδου είναι πάνω στις κορυφές του τετραγώνου πλευράς 1, e.g., (0, 0), (0, 1), (1, 0), (1, 1).

- Η πιο γενική περίπτωση,



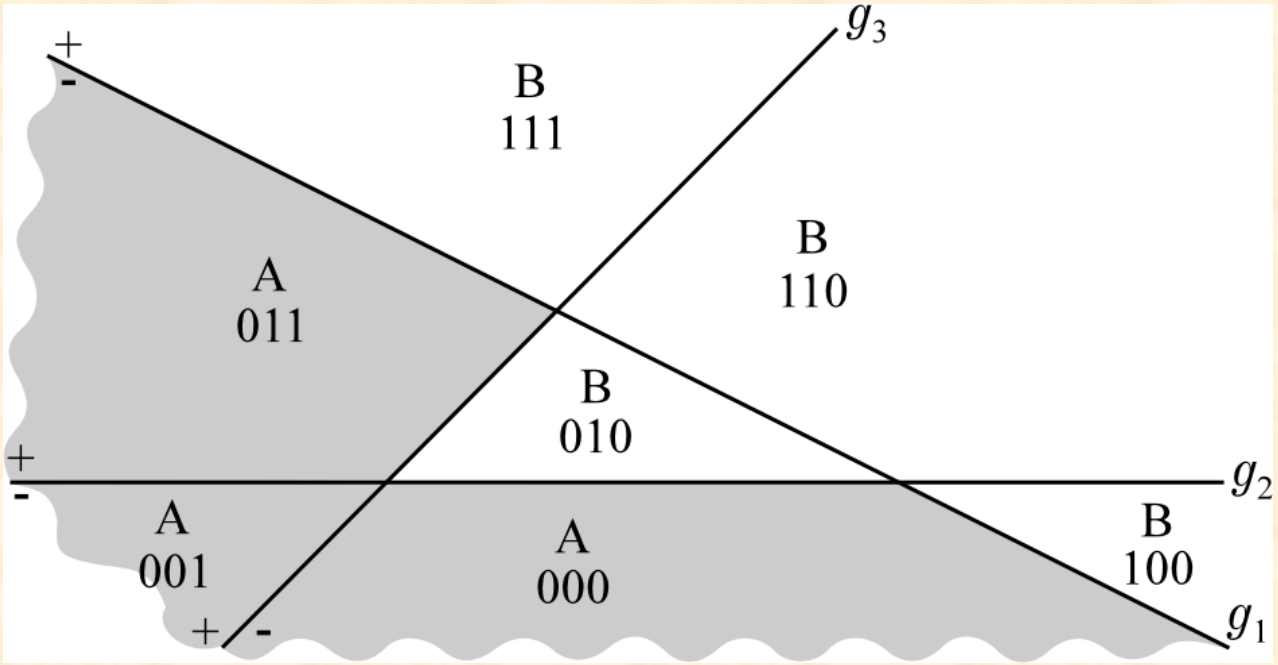
$$\underline{x} \in R^l$$

$$\underline{x} \rightarrow \underline{y} = [y_1, \ldots y_p]^T, \; y_i \in \{0,1\} \; i = 1, 2, \ldots p$$

πραγματοποιεί μία απεικόνιση ενός διανύσματος στις κορυφές του υπερκύβου H_p μοναδιαίας ακμής.

- Η απεικόνιση πραγματοποιείται με p νευρώνες καθένας από τους οποίους υλοποιεί ένα υπερεπίπεδο. Η έξοδος καθενός από αυτούς τους νευρώνες είναι 0 ή 1 ανάλογα με τη σχετική θέση του $\underline{x}$ ως προς το υπερεπίπεδο.

- Τομές αυτών των υπερεπιπέδων ορίζουν περιοχές στον l-διάστατο χώρο. Κάθε περιοχή αντιστοιχεί σε μία κορυφή του μοναδιαίου υπερκύβου H_p.

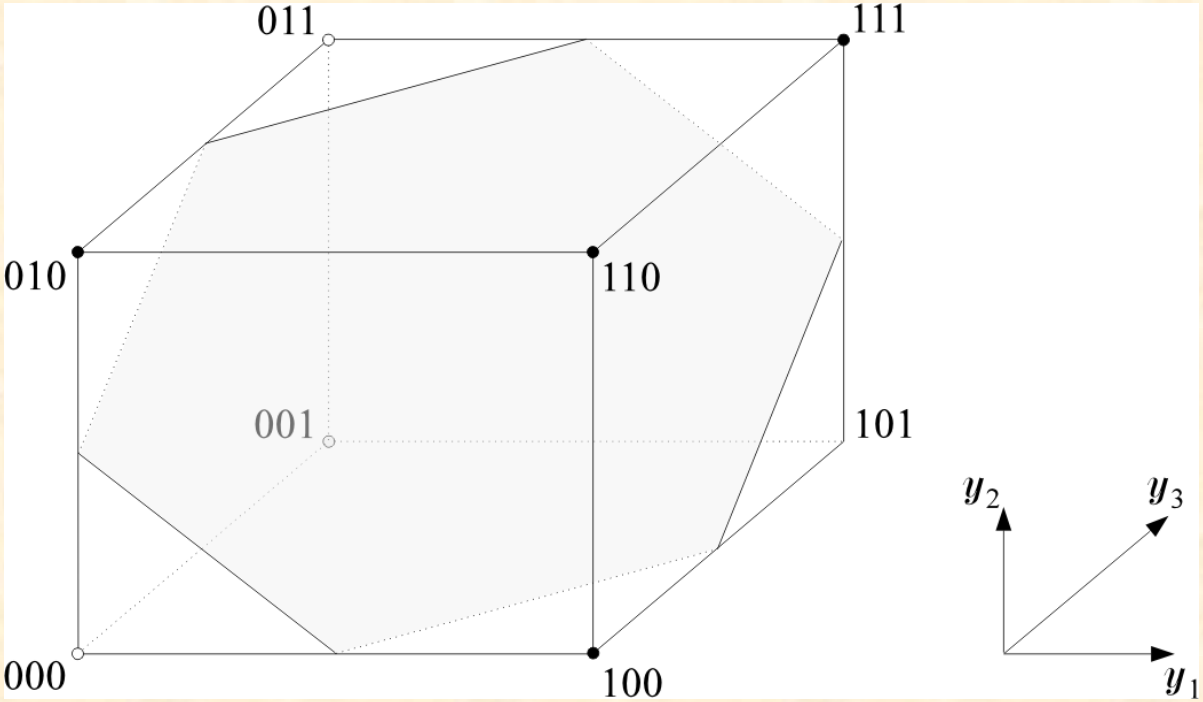


Για παράδειγμα, η κορυφή 001 αντιστοιχεί στην περιοχή που βρίσκεται

στην (-) πλευρά του $g_1(\underline{x})=0$

στην (-) πλευρά του $g_2(\underline{x})=0$

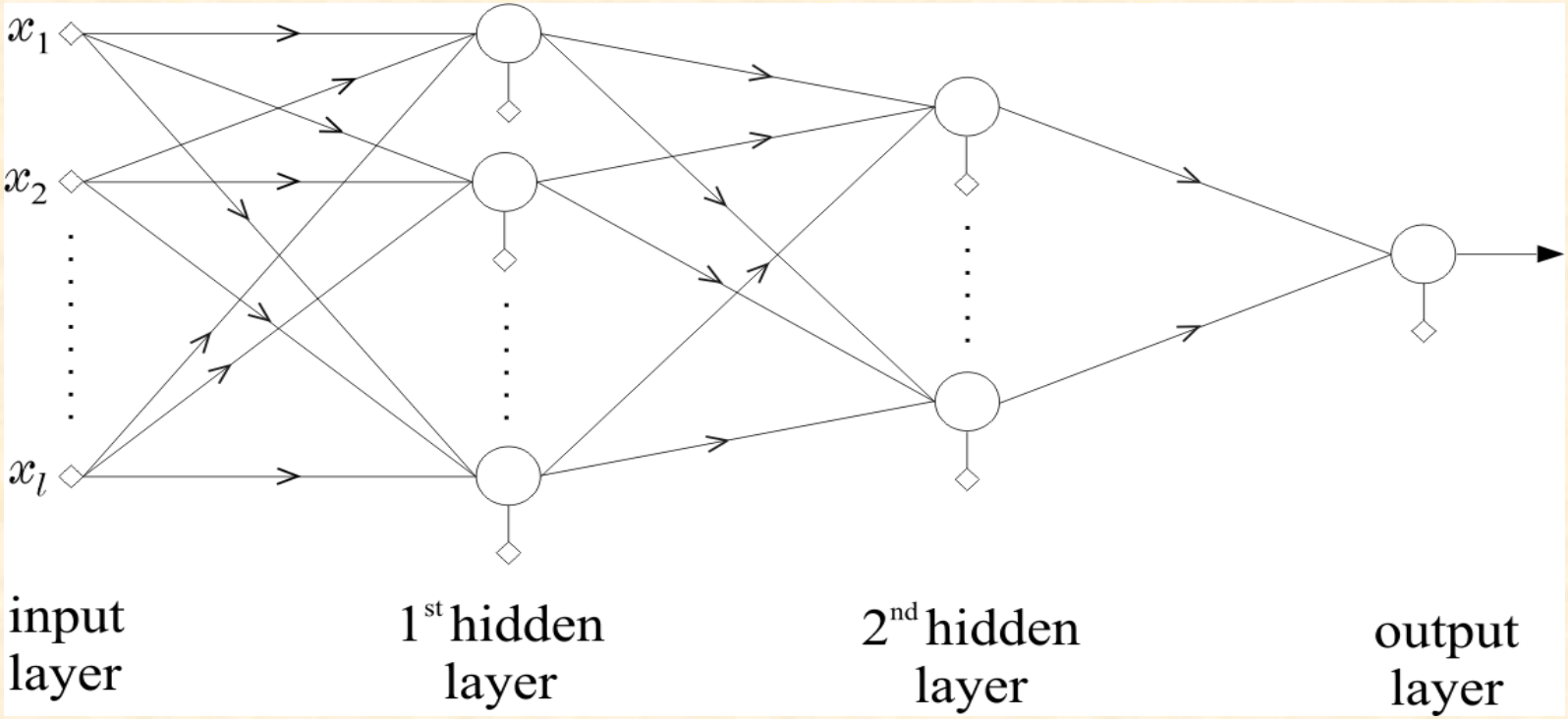
στην (+) πλευρά του $g_3(\underline{x})=0$



- Ο νευρώνας εξόδου υλοποιεί ένα υπερεπίπεδο στο μετασχηματισμένο χώρο, ο οποίος διαχωρίζει μερικές κορυφές από μερικές άλλες. Έτσι, το δίκτυο perceptron δύο επιπέδων έχει τη δυνατότητα να διαχωρίζει **κλάσεις που αποτελούνται από ενώσεις πολυεδρικών περιοχών**. Αλλά **ΟΧΙ ΟΠΟΙΕΣΔΗΠΟΤΕ** ενώσεις. Εξαρτάται από τη σχετική θέση των αντίστοιχων κορυφών.

❖ Δίκτυα perceptron τριών επιπέδων

- Η δομή τους



- Αυτά είναι ικανά να διαχωρίζουν κλάσεις οι οποίες αποτελούνται από **ΟΠΟΙΑΔΗΠΟΤΕ** ένωση πολυεδρικών περιοχών.
- Η ιδέα είναι παρόμοια μ' αυτή του προβλήματος XOR. Το δίκτυο υλοποιεί περισσότερα του ενός υπερεπίπεδα στο χώρο $\underline{y} \in R^p$

- Η λογική
  - Για κάθε κορυφή, που αντιστοιχεί, ας πούμε, στην κλάση Α κατασκεύασε ένα υπερεπίπεδο που αφήνει ΑΥΤΗ την κορυφή στην θετική πλευρά του (+) and ΟΛΕΣ τις άλλες κορυφές στην αρνητική πλευρά του (-).
  - Ο νευρώνας εξόδου υλοποιεί μία πύλη OR

- Συνολικά:

Το πρώτο επίπεδο του δικτύου ορίζει τα υπερεπίπεδα, το δεύτερο επίπεδο ορίζει τις περιοχές και ο νευρώνας εξόδου ορίζει τις κλάσεις.

❖ Σχεδιάζοντας perceptrons πολλαπλών επιπέδων

- Ένας τρόπος είναι να υιοθετήσουμε την παραπάνω λογική.

Στην πράξη όμως σπάνια γνωρίζουμε ακριβώς τα όρια των κλάσεων

- Ο άλλος τρόπος είναι να επιλέξουμε μία δομή και να υπολογίσουμε τα συναπτικά βάρη της έτσι ώστε να βελτιστοποιείται μία συνάρτηση κόστους.

❖ Ο αλγόριθμος οπισθοδρομικής διάδοσης (Backpropagation Algorithm)

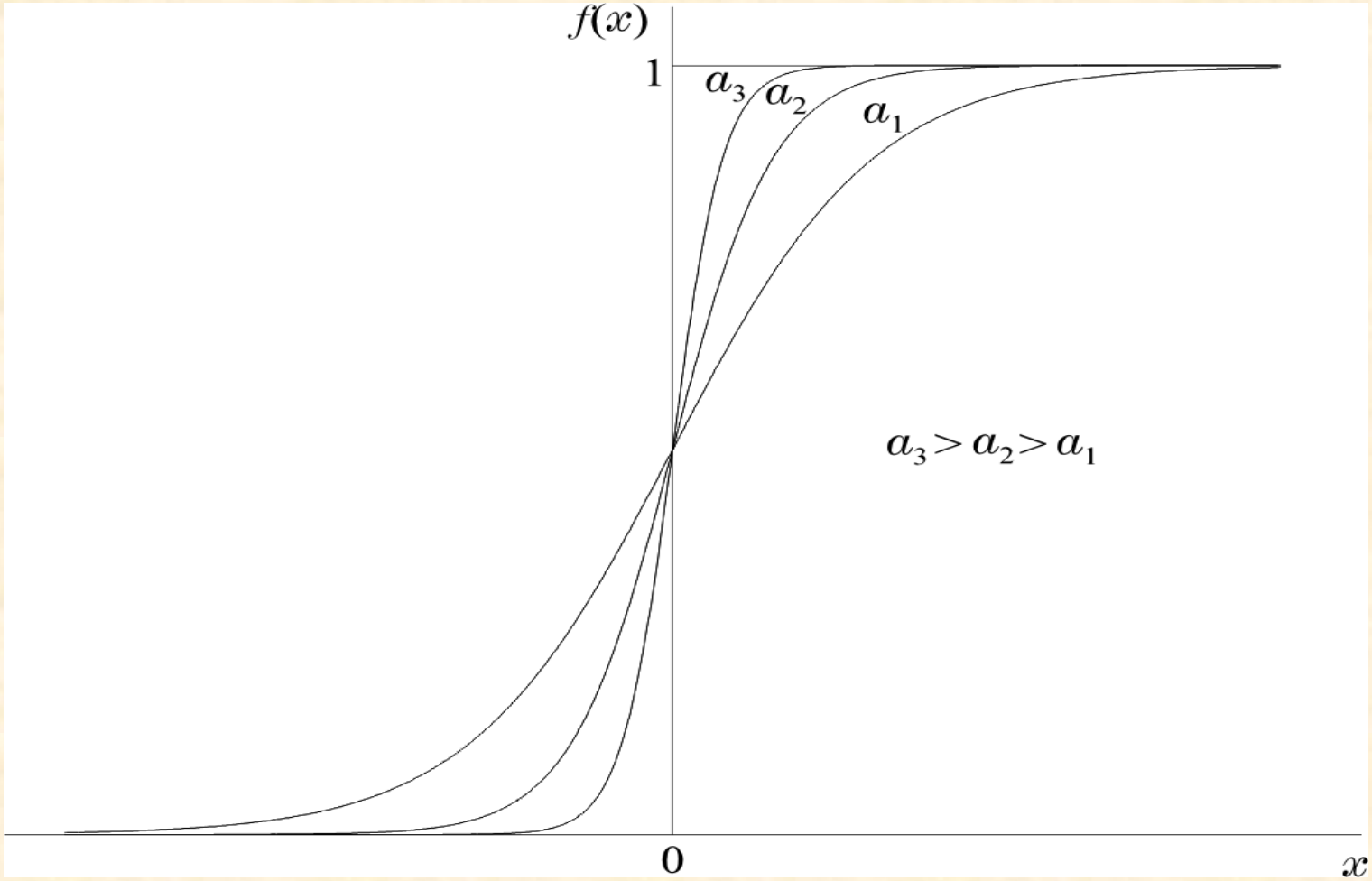
- Πρόκειται για μία αλγοριθμική διαδικασία που υπολογίζει τα συναπτικά βάρη επαναληπτικά, έτσι ώστε να ελαχιστοποιείται (βελτιστοποιείται) μία επιλεγμένη συνάρτηση κόστους.

- Ο υπολογισμός των παραγώγων εμπλέκεται σε ένα μεγάλο αριθμό διαδικασιών βελτιστοποίησης. Έτσι, οι μη συνεχείς συναρτήσεις ενεργοποίησης (εξόδου) δημιουργούν πρόβλημα, δηλαδή,

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

- Υπάρχει πάντα μία έξοδος κινδύνου!!! Η logistic συνάρτηση $f(x) = \dfrac{1}{1+\exp(-ax)}$

είναι ένα παράδειγμα. Άλλες συναρτήσεις είναι επίσης δυνατές και, μερικές φόρες, πιο επιθυμητές.


f(x)
1
a3
a2
a1
a3 > a2 > a1
0
x

- Τα βήματα:
  - Υιοθέτησε μία συνάρτηση κόστους για βελτιστοποίηση, δηλ.,
    - Σφάλμα ελαχίστων τετραγώνων (Least Squares Error)
    - Σχετική εντροπία (Relative Entropy)

    ανάμεσα σε επιθυμητές και πραγματικές αποκρίσεις του δικτύου για τα διαθέσιμα δεδομένα εκπαίδευσης. Δηλαδή, από εδώ και πέρα θα ζήσουμε με λάθη. Προσπαθούμε μόνο να τα ελαχιστοποιήσουμε, χρησιμοποιώντας ορισμένα κριτήρια.

  - Υιοθέτησε μία αλγοριθμική διαδικασία για τη βελτιστοποίηση της συνάρτησης κόστους ως προς τα συναπτικά βάρη, π.χ.,
    - Αλγόριθμος απότομης κατάδυσης (Gradient descent)
    - Αλγόριθμος του Newton
    - Αλγόριθμος συζυγών διευθύνσεων (Conjugate gradient algorithm)

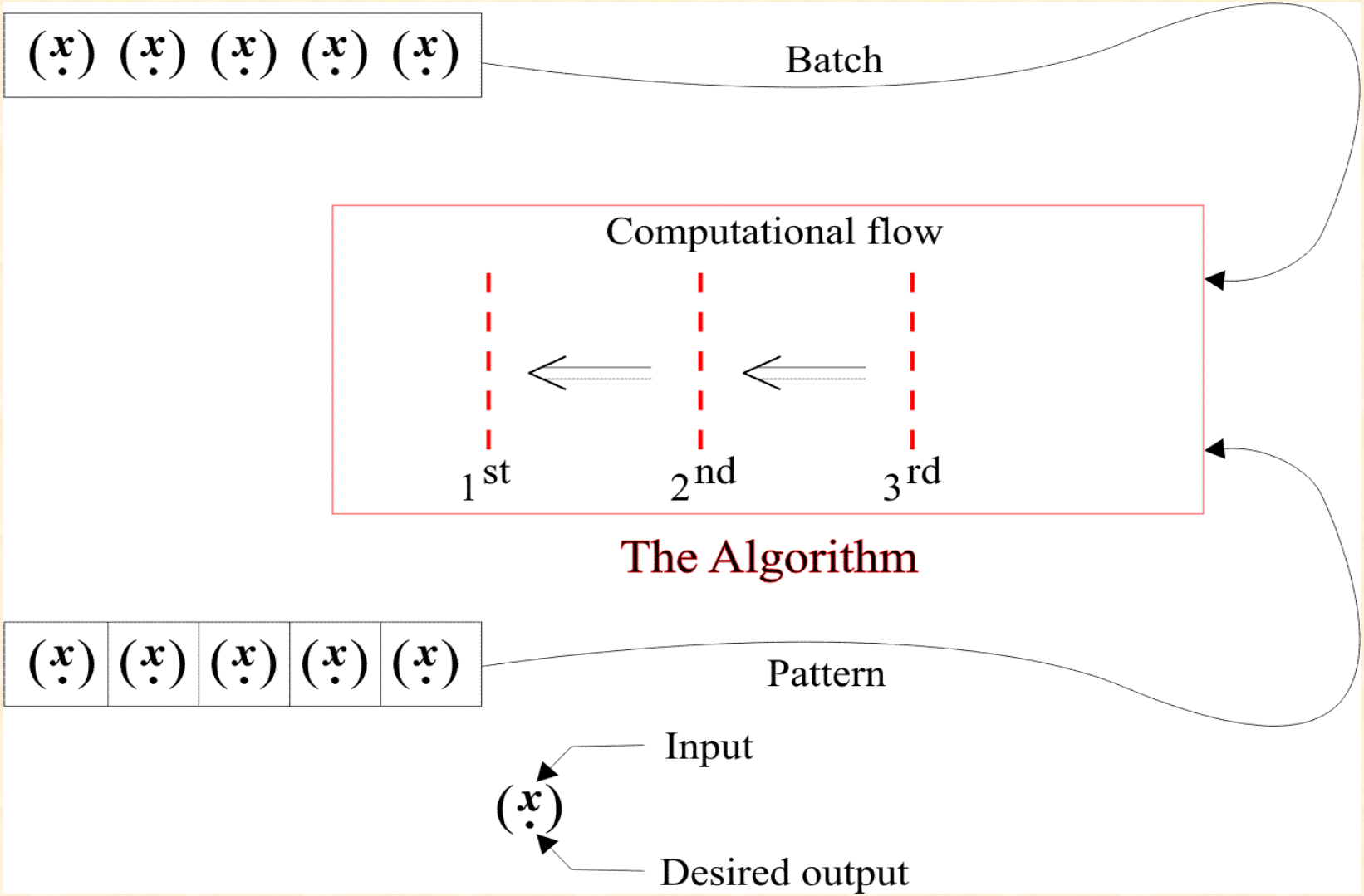
- Η διαδικασία βελτιστοποίησης είναι **μη γραμμική**. Για τη μέθοδο απότομής κατάδυσης (gradient descent)

$$\underline{w}_1^r(\text{new}) = \underline{w}_1^r(\text{old}) + \Delta\underline{w}_1^r$$

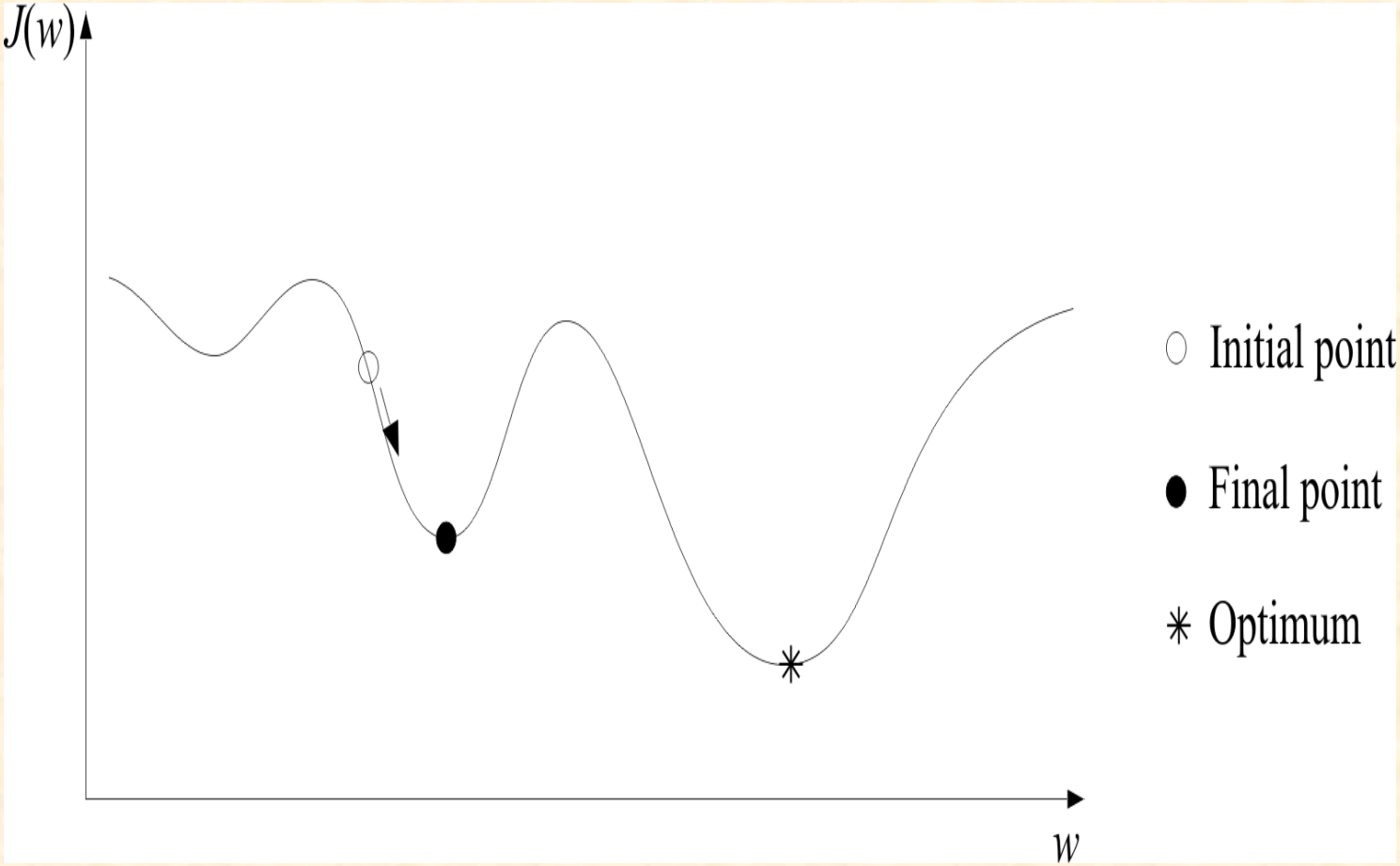
$$\Delta\underline{w}_1^r = -\mu\frac{\partial J}{\partial w_1^r}$$

- Η διαδικασία:
  - Αρχικοποίησε τυχαία τα άγνωστα βάρη με μικρές τιμές.
  - Υπολόγισε τους όρους του gradient προς τα πίσω, αρχίζοντας από τα βάρη του τελευταίου (3rd) επιπέδου και κινούμενος προς το πρώτο.
  - Ενημέρωσε τα βάρη.
  - Επανέλαβε τη διαδικασία έως ότου ικανοποιηθεί ένα κριτήριο τερματισμού.

- Δύο κύριες φιλοσοφίες:
  - Επεξεργασία κατά συρροή (Batch mode): Τα gradients του τελευταίου επιπέδου υπολογίζονται αφού παρουσιαστούν στον αλγόριθμο ΟΛΑ τα δεδομένα εκπαίδευσης, δηλ., αθροίζοντας όλους τους όρους λάθους.
  - Επεξεργασία κατά πρότυπο (Pattern mode): Τα gradients υπολογίζονται κάθε φορά που εμφανίζεται ένα νέο διάνυσμα εκπαίδευσης. Έτσι, τα gradients βασίζονται σε διαδοχικά μεμονωμένα σφάλματα.


Batch
Computational flow
1st
2nd
3rd
The Algorithm
Pattern
Input
Desired output

- Ένα σημαντικό πρόβλημα: Ο αλγόριθμος μπορεί να συγκλίνει σε ένα τοπικό ελάχιστο.



- Η επιλογή της συνάρτησης κόστους

  Παραδείγματα:

  - Η συνάρτηση ελαχίστων τετραγώνων (Least Squares)

$$J = \sum_{i=1}^{N} E(i)$$

$$E(i) = \sum_{m=1}^{k} e_m^2(i) = \sum_{m=1}^{k} (y_m(i) - \hat{y}_m(i))^2$$

$$i = 1,2,\ldots, N$$

$y_m(i) \rightarrow$ Επιθυμητή απόκριση του m^{th} νευρώνα εξόδου (1 or 0) για $\underline{x}(i)$

$\hat{y}_m(i) \rightarrow$ Πραγματική απόκριση του m^{th} νευρώνα εξόδου, στο διάστημα [0, 1], για είσοδο $\underline{x}(i)$

- Η συνάρτηση cross-entropy

$$J = \sum_{i=1}^{N} E(i)$$

$$E(i) = \sum_{m=1}^{k} \{ y_m(i) \ln \hat{y}_m(i) + (1 - y_m(i)) \ln(1 - \hat{y}_m(i)) \}$$

Αυτή προϋποθέτει την ερμηνεία των y and $\hat{y}$ ως **πιθανότητες.**

- Ποσοστό σφάλματος ταξινόμησης (Classification error rate). Είναι επίσης γνωστό ως_discriminative learning. Οι περισσότερες από αυτές τις τεχνικές χρησιμοποιούν μία εξομαλυσμένη έκδοση του σφάλματος ταξινόμησης.

- Σχόλιο 1: Ένα κοινό χαρακτηριστικό των παραπάνω συναρτήσεων είναι ο κίνδυνος σύγκλισης σε κάποιο τοπικό ελάχιστο. Οι **"καλώς ορισμένες"** ("**Well formed**") συναρτήσεις κόστους εγγυώνται σύγκλιση σε μία "καλή" λύση, δηλαδή σε μία λύση που να ταξινομεί σωστά ΟΛΑ τα δεδομένα εκπαίδευσης, υπό την προϋπόθεση ότι υπάρχει μία τέτοια λύση. Η cross-entropy συνάρτηση κόστους **είναι** καλώς ορισμένη. Η συνάρτηση ελαχίστων τετραγώνων **δεν είναι**.

- Σχόλιο 2: Αμφότερες οι συναρτήσεις κόστους ελαχίστων τετραγώνων και cross entropy οδηγούν σε τιμές εξόδου $\hat{y}_m(i)$ που προσεγγίζουν κατά βέλτιστο τρόπο τις εκ των υστέρων πιθανότητες για κάθε κλάση (Optimally class a-posteriori probabilities)!!!

$$\hat{y}_m(i) \cong P(\omega_m | \underline{x}(i))$$

Δηλ., την πιθανότητα της κλάσης ω_m δοθέντος του $\underline{x}(i)$.

Πρόκειται για ένα πολύ ενδιαφέρον αποτέλεσμα. **Δεν** εξαρτάται από τις κατανομές των κλάσεων. Είναι ένα χαρακτηριστικό **ορισμένων** συναρτήσεων κόστους. Η ποιότητα της προσέγγισης εξαρτάται από το μοντέλο που υιοθετήθηκε. Επιπλέον, ισχύει μόνο στο ολικό ελάχιστο.

❖Παραλλαγές του αλγορίθμου Backpropagation

- Αλγόριθμος backpropagation με όρο ορμής (momentum term)
  - Προστατεύει τον αλγόριθμο από περιπτώσεις ταλάντωσης και, κατά συνέπεια, αργής σύγκλισης
  - Εξισώσεις

$$\Delta \boldsymbol{w}_j^r(\text{new}) = \alpha \Delta \boldsymbol{w}_j^r(\text{old}) - \mu \sum_i^N \delta_j^r(i) \boldsymbol{y}^{r-1}(i)$$

$$\boldsymbol{w}_j^r(\text{new}) = \boldsymbol{w}_j^r(\text{old}) + \Delta \boldsymbol{w}_j^r(\text{new})$$

- Προσαρμοστικός (adaptive) αλγόριθμος backpropagation
  - Επιταχύνει ή επιβραδύνει ανάλογα με το είδος της περιοχής του landscape της συνάρτησης κόστους που βρίσκεται η τρέχουσα εκτίμηση
  - Εξισώσεις

$$\frac{J(t)}{J(t-1)} < 1, \quad \mu(t) = r_i \mu(t-1)$$

$$\frac{J(t)}{J(t-1)} > c, \quad \mu(t) = r_d \mu(t-1)$$

$$1 \leq \frac{J(t)}{J(t-1)} \leq c, \quad \mu(t) = \mu(t-1)$$

- **Επιλογή του μεγέθους του δικτύου.**

  Πόσο μεγάλο μπορεί να είναι;; Πόσα επίπεδα νευρώνων και πόσοι νευρώνες ανά επίπεδο;;

  Υπάρχουν δύο κύριες κατευθύνσεις

  - Τεχνικές «κλαδέματος» (Pruning Techniques)**:** Αυτές ξεκινούν με ένα δίκτυο μεγάλου μεγέθους και απομακρύνουν επαναληπτικά βάρη και/ή νευρώνες, σύμφωνα με ένα κριτήριο.

—Μέθοδοι που βασίζονται στην ευαισθησία των παραμέτρων

$$\delta J = \sum_i g_i \delta w_i + \frac{1}{2} \sum_i h_{ii} \delta w_i^2 + \frac{1}{2} \sum_i \sum_j h_{ij} \delta w_i \delta w_j$$

+ όροι υψηλής τάξης

όπου

$$g_i = \frac{\partial J}{\partial w_i}, \; h_{ij} = \frac{\partial^2 J}{\partial w_i \partial w_j}$$

Κοντά σ' ένα ελάχιστο και υποθέτοντας ότι

$$\delta J \cong \frac{1}{2} \sum_i h_{ii} \delta w_i^2$$

Το κλάδεμα τώρα γίνεται ως ακολούθως:

- ✓ Εκπαίδευσε το δίκτυο χρησιμοποιώντας Backpropagation για έναν αριθμό βημάτων

- ✓ Υπολόγισε τις «προεξοχές» (saliencies)

$$s_i = \frac{h_{ii} w_i^2}{2}$$

- ✓ Απομάκρυνε τα βάρη με μικρά s_i.
- ✓ Επανέλαβε τη διαδικασία

— Μέθοδοι που βασίζονται στην κανονικοποίηση συνάρτησης (function regularization)

$$J = \sum_{i=1}^{N} E(i) + aE_p(\underline{w})$$

Ο δεύτερος όρος ευνοεί μικρές τιμές για τα βάρη, π.χ.,

$$E_p(\underline{\omega}) = \sum_k h(w_k^2)$$

$$h(w_k^2) = \frac{w_k^2}{w_0^2 + w_k^2}$$

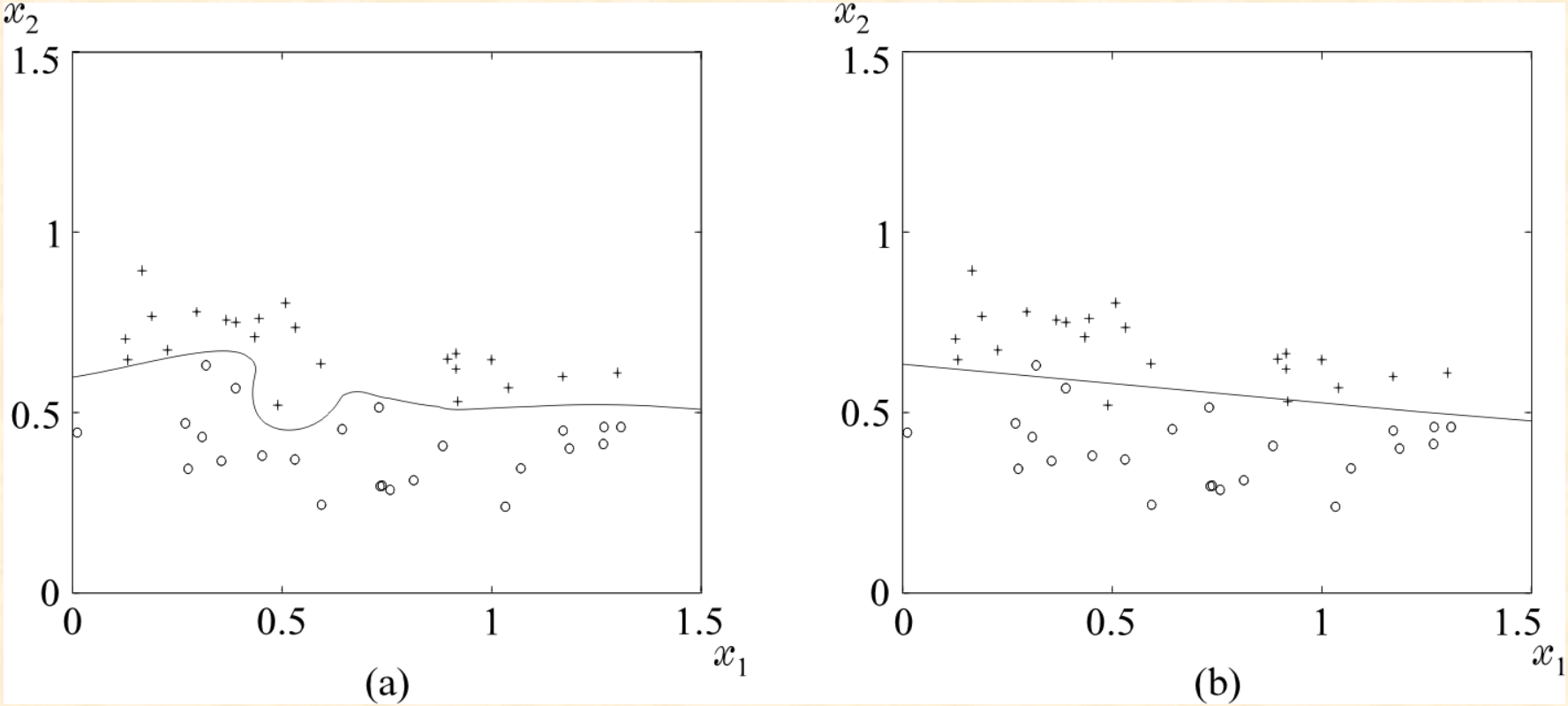
όπου $w_0 \cong 1$

Μετά από μερικά βήματα εκπαίδευσης, τα βάρη με μικρές τιμές απομακρύνονται.

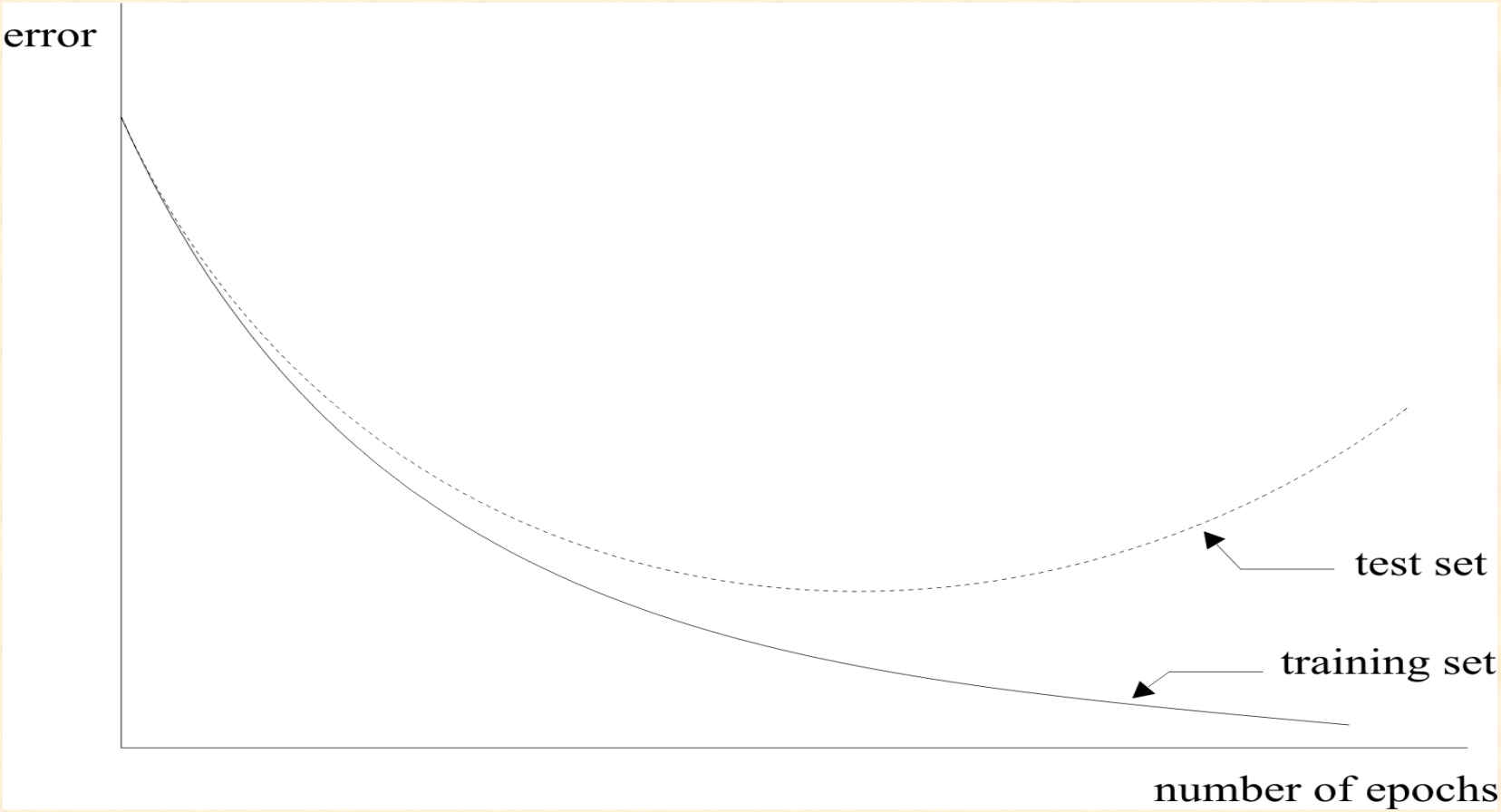
- Κατασκευαστικές τεχνικές (Constructive techniques): Ξεκινούν με ένα δίκτυο μικρού μεγέθους και το μεγαλώνουν, σύμφωνα με μία προκαθορισμένη διαδικασία και κριτήριο.

- Σχόλιο: Γιατί να μην ξεκινήσουμε με ένα δίκτυο μεγάλου μεγέθους και να αφήσουμε τον αλγόριθμο να αποφασίσει ποια βάρη είναι μικρά;; Η προσέγγιση αυτή είναι απλοϊκή. Παραβλέπει το γεγονός ότι οι ταξινομητές πρέπει να έχουν καλή δυνατότητα **γενίκευσης (generalization)**. Ένα μεγάλο δίκτυο μπορεί να δώσει μικρά σφάλματα για το σύνολο εκπαίδευσης, αφού μπορεί να μάθει τις συγκεκριμένες λεπτομέρειές του. Από την άλλη μεριά, δεν αναμένεται να παρουσιάζει καλή απόδοση για δεδομένα στα οποία δεν εκπαιδεύτηκε. Το μέγεθος του δικτύου πρέπει να είναι:
  - Αρκούντως μεγάλο για να μπορεί να μάθει τι κάνει όμοια τα δεδομένα της ίδιας κλάσης και τι κάνει ανόμοια τα δεδομένα διαφορετικών κλάσεων.
  - Αρκούντως μικρό για να μην μπορεί να μάθει τις διαφορές μεταξύ δεδομένων της ίδιας κλάσης. Το τελευταίο οδηγεί στο λεγόμενο υπερ-ταίριασμα (overfitting).

Παράδειγμα:



- Υπερεκπαίδευση (Overtraining) είναι μία άλλη όψη του ίδιου νομίσματος, δηλ. το δίκτυο προσαρμόζεται στις ιδιαιτερότητες του συνόλου δεδομένων.



❖ Γενικευμένοι γραμμικοί ταξινομητές

- Ας θεωρήσουμε το πρόβλημα XOR. Η απεικόνιση

$$\underline{x} \to \underline{y} = \begin{bmatrix} f(g_1(\underline{x})) \\ f(g_2(\underline{x})) \end{bmatrix}$$

$f(.) \to$ Η συνάρτηση ενεργοποίησης μετασχηματίζει το μη γραμμικό πρόβλημα σε γραμμικό.

- Στη γενικότερη περίπτωση:

  - Έστω $\underline{x} \in R^l$ και ένα μη γραμμικό πρόβλημα.

$$f_i(.), i = 1,2,\ldots,k$$

- Υπάρχουν κατάλληλες συναρτήσεις και κατάλληλο k, έτσι ώστε η απεικόνιση

$$\underline{x} \rightarrow \underline{y} = \begin{bmatrix} f_1(\underline{x}) \\ \dots \\ f_k(\underline{x}) \end{bmatrix}$$

να μετασχηματίζει σε γραμμικό το πρόβλημα στο χώρο $\underline{y} \in R^k$?

- Αν αυτό ισχύει, τότε υπάρχει υπερεπίπεδο $\underline{w} \in R^k$ έτσι ώστε

$$\text{Αν} \quad w_0 + \underline{w}^T \underline{y} > 0, \quad \underline{x} \in \omega_1$$

$$w_0 + \underline{w}^T \underline{y} < 0, \quad \underline{x} \in \omega_2$$

- Σε μία τέτοια περίπτωση αυτό είναι ισοδύναμο με την προσέγγιση της μη γραμμικής συνάρτησης $g(\underline{x})$, εκπεφρασμένη ως γραμμικός συνδυασμός των $f_i(\underline{x})$, δηλ.,

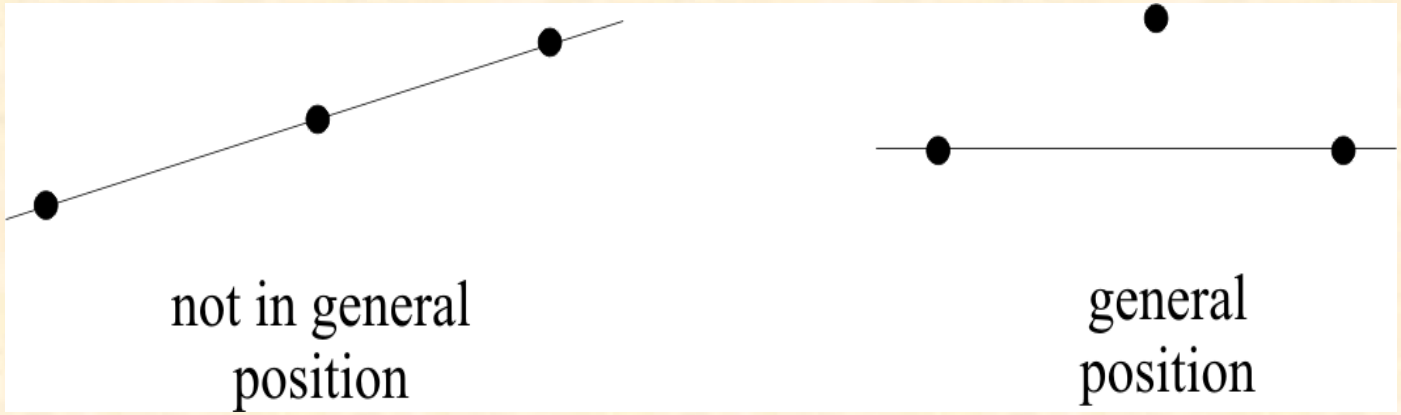
$$g(\underline{x}) \cong w_0 + \sum_{i=1}^{k} w_i f_i(\underline{x}) \quad (><) \quad 0$$

- Δοθέντων των $f_i(\underline{x})$, η διαδικασία υπολογισμού των βαρών είναι γραμμική.

- Πόσο λογικό είναι αυτό;;
  - Από τη σκοπιά της αριθμητικής ανάλυσης αυτό δικαιολογείται αν οι $f_i(\underline{x})$ είναι συναρτήσεις παρεμβολής.
  - Από τη σκοπιά της αναγνώρισης προτύπων, αυτό δικαιολογείται από το θεώρημα του Cover.

- Χωρητικότητα του l-διάστατου χώρου σε γραμμικές διχοτομήσεις
  - Έστω N σημεία στον R^l χώρο που υποθέτουμε ότι βρίσκονται σε γενική θέση, δηλαδή:

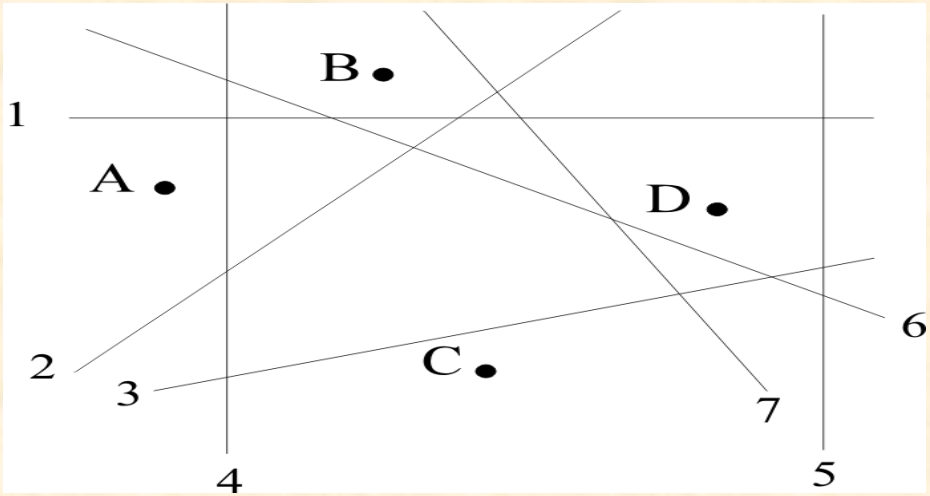
    Καμία ομάδα $\ell+1$ τέτοιων σημείων δεν βρίσκεται σε έναν $\ell-1$ διάστατο χώρο.



- Το θεώρημα του Cover λέει: Ο αριθμός των ομαδοποιήσεων που μπορούν να υλοποιηθούν από (l-1)-διάστατα υπερεπίπεδα προκειμένου να διαχωριστούν N σημεία σε δύο κλάσεις είναι

$$O(N,l) = 2\sum_{i=0}^{l}\binom{N-1}{i}, \qquad \binom{N-1}{i} = \frac{(N-1)!}{(N-1-i)!i!}$$

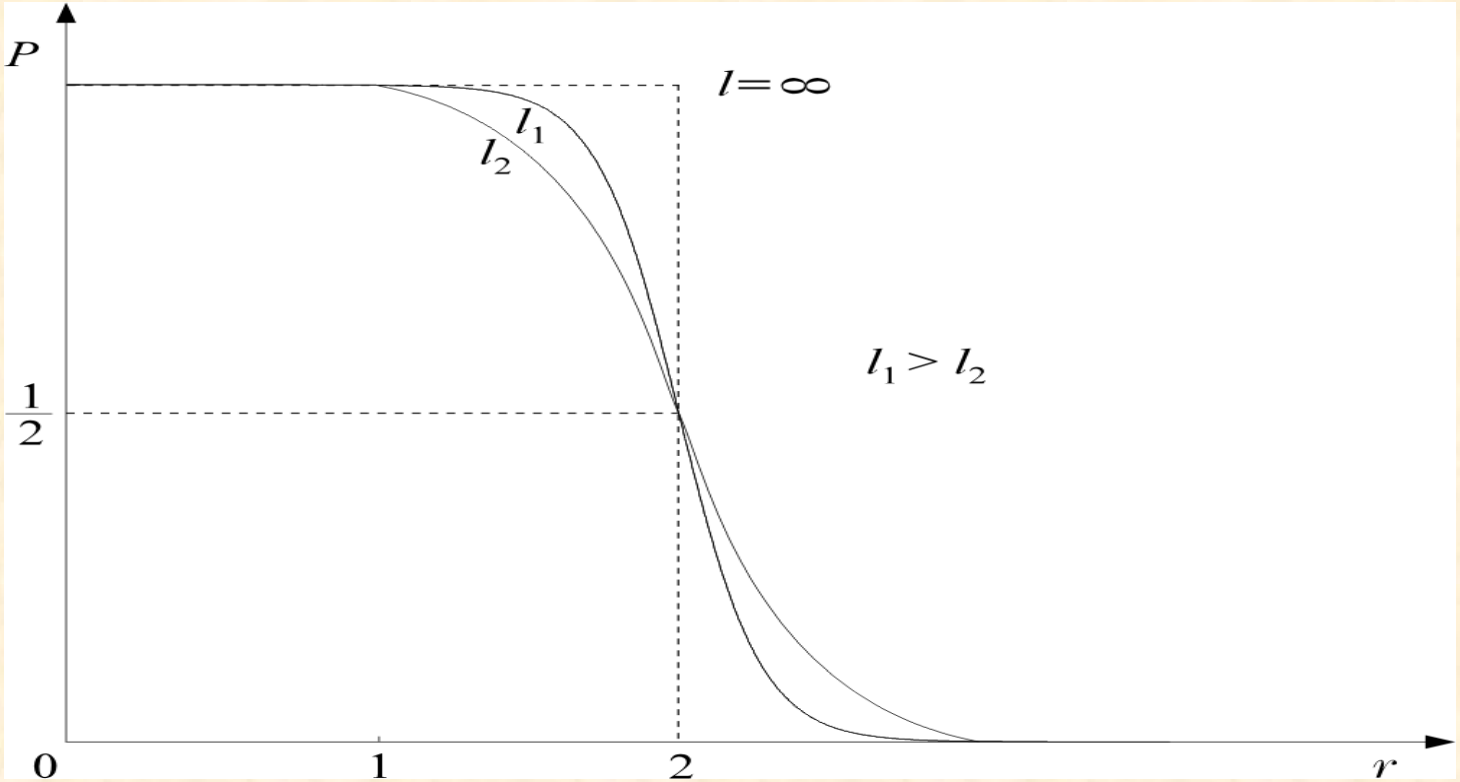
Example: N=4, l=2, $O(4,2)$=**14**



Σημείωση:Ο συνολικός αριθμός δυνατών ομαδοποιήσεων είναι 2^4=16

- Η πιθανότητα ομαδοποίησης N σημείων σε δύο γραμμικώς διαχωρίσιμες κλάσεις είναι

$$\frac{O(N,l)}{2^N} = P_N^l$$



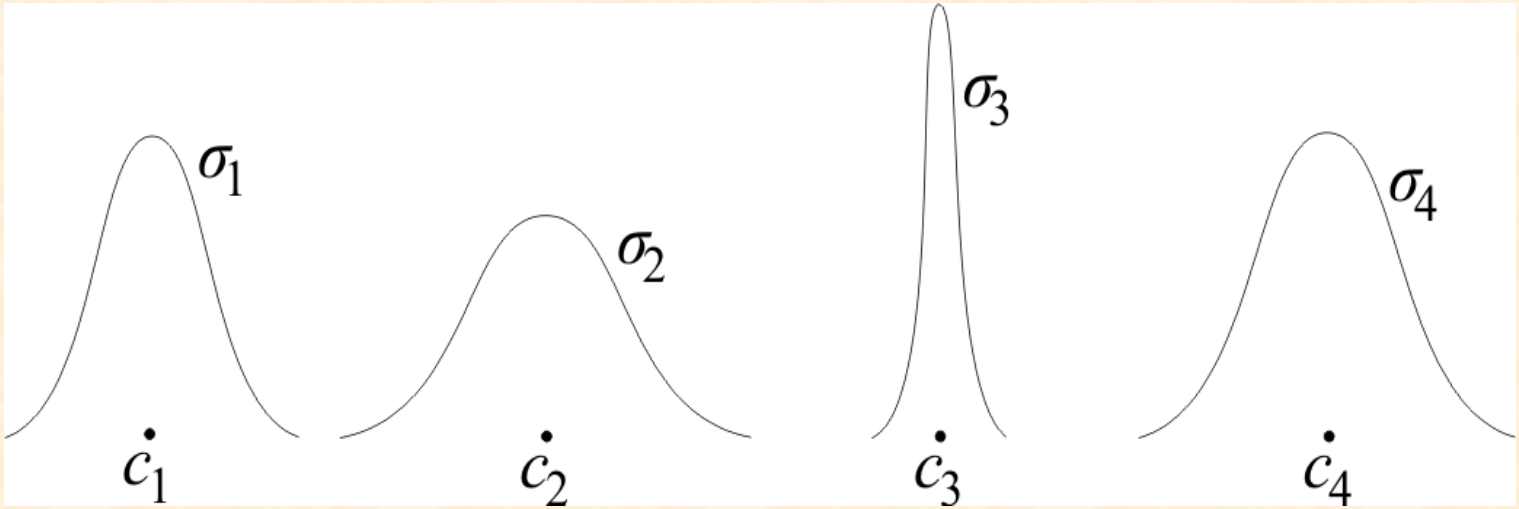
$$N = r(l+1)$$

Έτσι, η πιθανότητα να έχουμε N σημεία σε γραμμικώς διαχωρίσιμες κλάσεις τείνει στο 1, για μεγάλο l, υπό την προϋπόθεση ότι $N<2(l+1)$

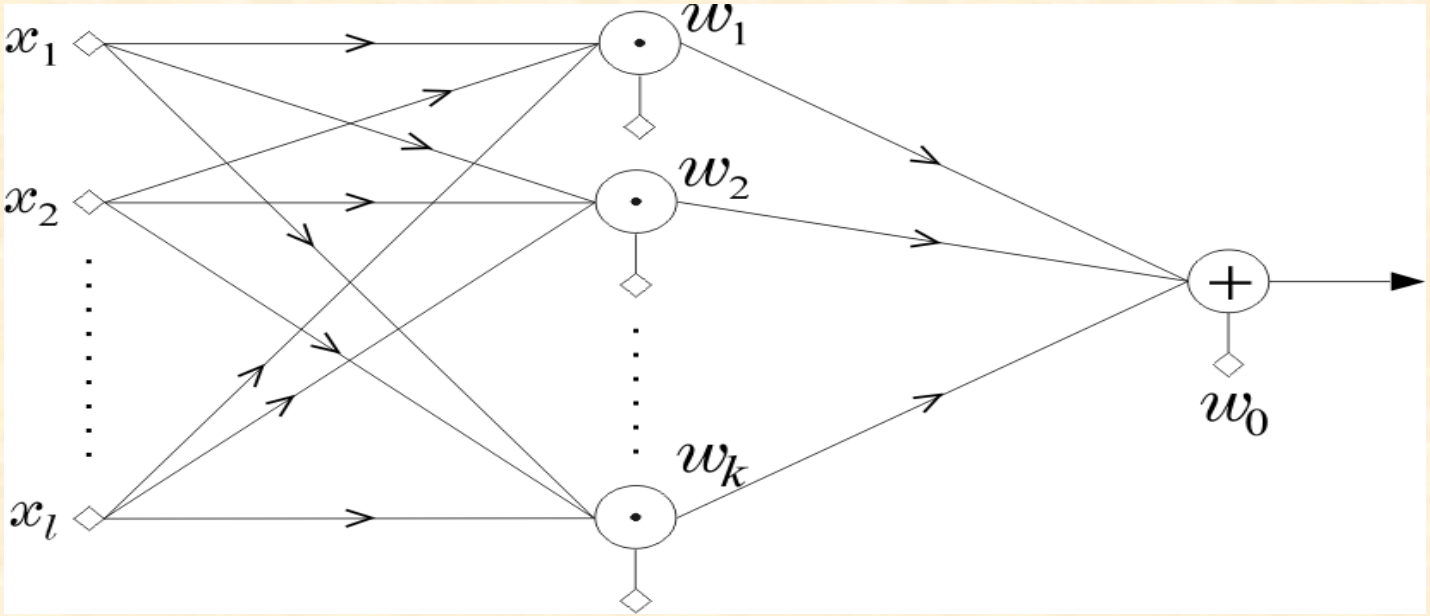
Έτσι, απεικονίζοντας σε χώρο υψηλότερης διάστασης, αυξάνουμε την πιθανότητα γραμμικού διαχωρισμού, υπό την προϋπόθεση ότι ο χώρος δεν παρουσιάζει μεγάλη πυκνότητα σε σημεία δεδομένων.

❖ Δίκτυα συνάρτησης ακτινικής βάσης (Radial Basis Function Networks - RBF)

- Επέλεξε



$$f_i(\underline{x}) = \exp\left(-\frac{\left\| x - \underline{c}_i \right\|^2}{2\sigma_i^2} \right)$$



Ισοδύναμο με ένα δίκτυο ενός κρυφού επιπέδου με RBF συναρτήσεις ενεργοποίησης και γραμμικό νευρώνα εξόδου.

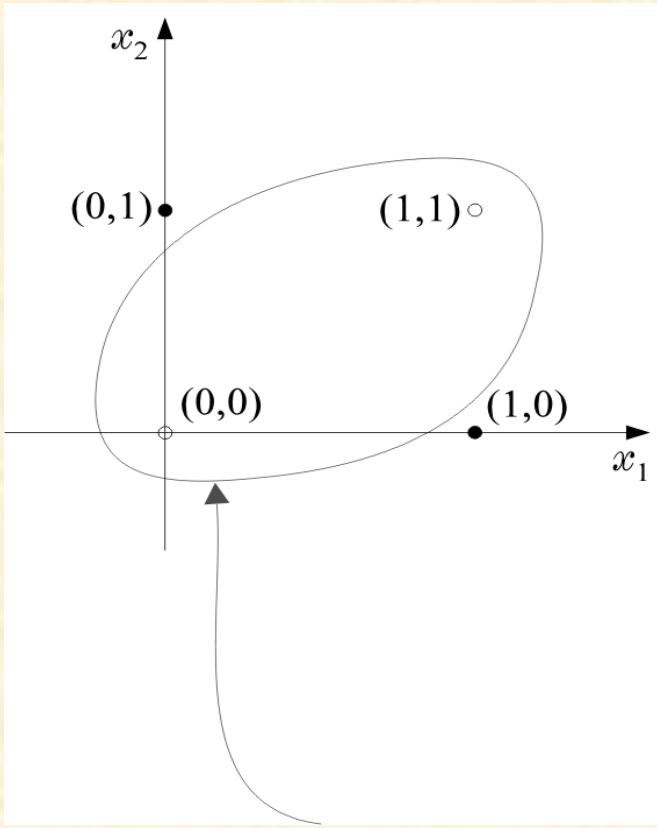
➢ Παράδειγμα: Το πρόβλημα XOR

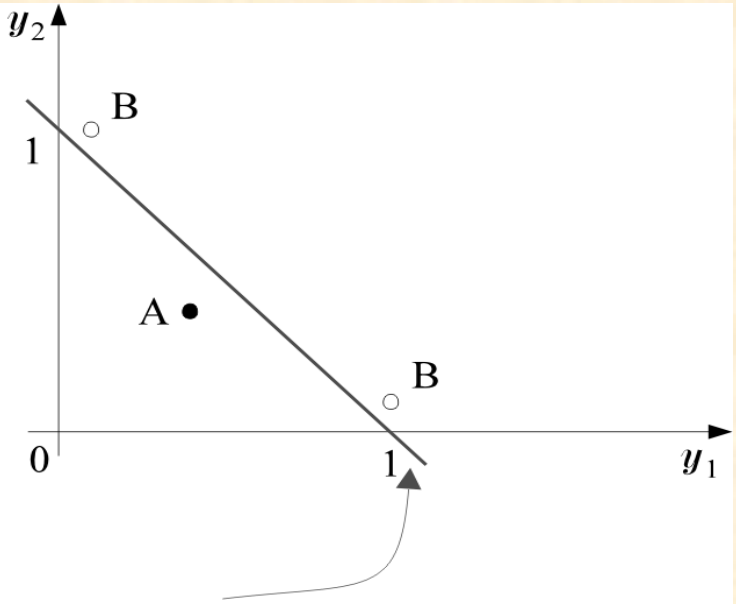
- Ορίζουμε:

$$\underline{c}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \underline{c}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \sigma_1 = \sigma_2 = \frac{1}{\sqrt{2}}$$

$$\underline{y} = \begin{bmatrix} \exp(-\|\underline{x} - \underline{c}_1\|^2) \\ \exp(-\|\underline{x} - \underline{c}_2\|^2) \end{bmatrix}$$

- $$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.135 \\ 1 \end{bmatrix}, \ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0.135 \end{bmatrix}$$
$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.368 \\ 0.368 \end{bmatrix}, \ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0.368 \\ 0.368 \end{bmatrix}$$





$$g(\underline{y}) = y_1 + y_2 - 1 = 0$$

$$g(\underline{x}) = \exp(-\|\underline{x} - \underline{c}_1\|^2) + \exp(-\|\underline{x} - \underline{c}_2\|^2) - 1 = 0$$

➢ Εκπαίδευση δικτύων RBF

- Σταθερά κέντρα: Επέλεξε τα κέντρα με τυχαίο τρόπο από το σύνολο δεδομένων. Επίσης, σταθεροποίησε τα σ_i's. Τότε

$$g(\underline{x}) = w_0 + \underline{w}^T \underline{y}$$

είναι ένα τυπικό πρόβλημα σχεδίασης γραμμικού ταξινομητή.

- Εκπαίδευση των κέντρων: Πρόκειται για ένα μη γραμμικό πρόβλημα βελτιστοποίησης

- Συνδυασμός τεχνικών εκπαίδευσης με επίβλεψη και χωρίς επίβλεψη.

- Το χωρίς επίβλεψη τμήμα αποκαλύπτει τάσεις ομαδοποίησης των δεδομένων και αντιστοιχεί τα κέντρα των RBF νευρώνων στους αντιπροσώπους των ομάδων (clusters)

❖ Μοντέλα καθολικής προσέγγισης (Universal Approximators)

Έχει αποδειχθεί ότι οποιαδήποτε μη γραμμική συνεχής συνάρτηση μπορεί να προσεγγιστεί αυθαίρετα κοντά, τόσο από ένα perceptron δύο επιπέδων, με νευρώνες σιγμοειδούς συναρτήσεων ενεργοποίησης, όσο και από ένα δίκτυο RBF, υπό την προϋπόθεση ότι χρησιμοποιείται αρκούντως μεγάλος αριθμός νευρώνων.

❖ Perceptrons πολλών επιπέδων vs. Δικτύων RBF

- MLP's περιλαμβάνουν ενεργοποιήσεις ολικής φύσης. Ένας κόμβος εδώ δίνει την ίδια απόκριση για όλα τα σημεία του επιπέδου $w^T \underline{x} = c$.

- Τα δίκτυα RBF παρουσιάζουν ενεργοποίηση τοπικής φύσης, λόγω της εκθετικής μείωσης καθώς απομακρυνόμαστε από το κέντρο ενός κόμβου.

- Τα MLP's μαθαίνουν πιο αργά αλλά παρουσιάζουν καλύτερες ικανότητες γενίκευσης

❖ Μηχανές διανυσματικής στήριξης: Η μη γραμμική περίπτωση

- Υπενθυμίζουμε ότι η πιθανότητα για γραμμικώς διαχωρίσιμες κλάσεις αυξάνει καθώς **η διάσταση** των διανυσμάτων χαρακτηριστικών **αυξάνει**. Θεωρείστε την απεικόνιση:

$$\underline{x} \in R^l \rightarrow \underline{y} \in R^k, \; k > l$$

Στη συνέχεια χρησιμοποιήστε SVM στον R^k

- Θυμηθείτε ότι στην περίπτωση αυτή το δυϊκό πρόβλημα θα είναι

$$\underset{\underline{\lambda}}{\text{maximize}} \; \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \underline{y}_i^T \underline{y}_j\right)$$

$$\text{where } \underline{y}_i \in R^k$$

Επίσης ο ταξινομητής θα είναι

$$g(\underline{y}) = \underline{w}^T \underline{y} + w_0$$

$$= \sum_{i=1}^{N_s} \lambda_i y_i \underline{y}_i \underline{y}$$

where $\underline{x} \to \underline{y} \in R^k$

Έτσι, εμπλέκονται εσωτερικά γινόμενα στο χώρο υψηλής διάστασης. Έτσι, έχουμε

- Υψηλή υπολογιστική πολυπλοκότητα

- Κάτι έξυπνο: Υπολόγισε τα εσωτερικά γινόμενα στον χώρο **υψηλής** διάστασης σαν συνάρτηση των εσωτερικών γινομένων στο χώρο **χαμηλής** διάστασης!!!

- Είναι αυτό ΔΥΝΑΤΟ?? Ναι. Να ένα παράδειγμα

$$\text{Έστω} \quad \underline{x} = \left[x_1,\ x_2 \right]^T \in R^2$$

$$\text{Έστω} \quad \underline{x} \to \underline{y} = \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} \in R^3$$

Τότε, είναι εύκολο να δείξουμε ότι

$$\underline{y}_i^T \underline{y}_j = (\underline{x}_i^T \underline{x}_j)^2$$

➢ Θεώρημα του Mercer

$$\text{Έστω } \underline{x} \to \underline{\Phi}(\underline{x}) \in H$$

Τότε, το εσωτερικό γινόμενο στο χώρο *H*

$$\sum_r \Phi_r(\underline{x})\Phi_r(\underline{y}) = K(\underline{x}, \underline{y})$$

όπου

$$\int K(\underline{x}, \underline{y}) g(\underline{x}) g(\underline{y}) d\underline{x} d\underline{y} \geq 0$$

για **οποιαδήποτε** $g(\underline{x})$, $\underline{x}$:

$$\int g^2(\underline{x}) d\underline{x} < +\infty,$$

$K(\underline{x},\underline{y})$ είναι συμμετρική συνάρτηση γνωστή ως πυρήνας (kernel).

- Ισχύει επίσης και το αντίθετο. Κάθε πυρήνας, με τις παραπάνω ιδιότητες, αντιστοιχεί σε εσωτερικό γινόμενο σε ΚΑΠΟΙΟ χώρο!!!

- Παραδείγματα πυρήνων
  - Συναρτήσεις ακτινικής βάσης (Radial basis Functions):

$$K(\underline{x},\underline{z}) = \exp\left(-\frac{\|\underline{x}-\underline{z}\|^2}{\sigma^2}\right)$$

  - Πολυωνυμικές:

$$K(\underline{x},\underline{z}) = (\underline{x}^T\underline{z}+1)^q,\ q>0$$

  - Συνάρτηση υπερβολικής εφαπτομένης:

$$K(\underline{x},\underline{z}) = \tanh(\beta\underline{x}^T\underline{z}+\gamma)$$

για κατάλληλες τιμές των $\beta,\ \gamma$.

- Διατύπωση του SVM
  - Βήμα 1: Επέλεξε κατάλληλο πυρήνα. Αυτή η ενέργεια υπονοεί την απεικόνιση σε ένα (άγνωστο) χώρο υψηλότερης διάστασης.

  - Βήμα 2:

$$\max_{\underline{\lambda}} \left(\sum_i \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j K(\underline{x}_i, \underline{x}_j) \right)$$

$$\text{subject to : } \; 0 \le \lambda_i \le C, \; i = 1,2,\ldots, N$$

$$\sum_i \lambda_i y_i = 0$$

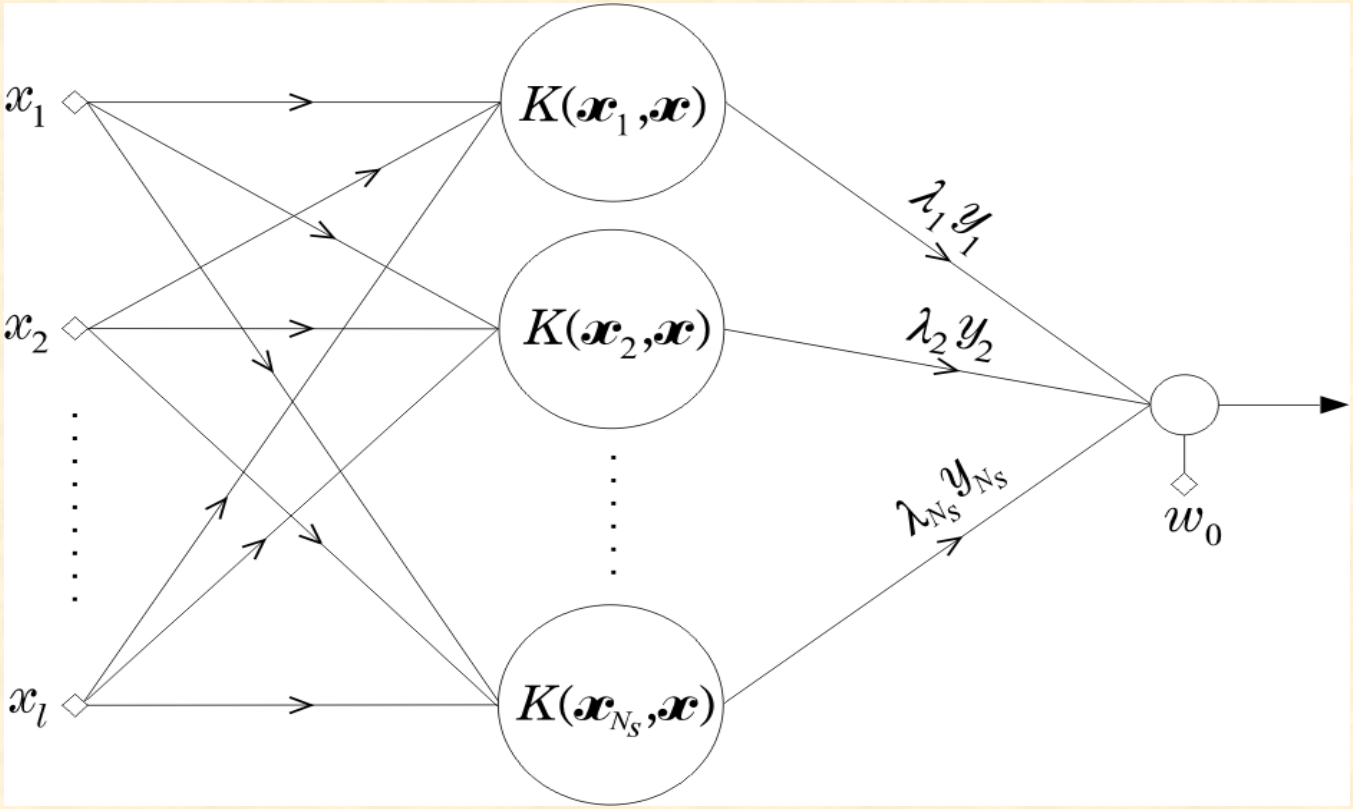
Αυτό καταλήγει σε έναν υπονοούμενο συνδυασμό

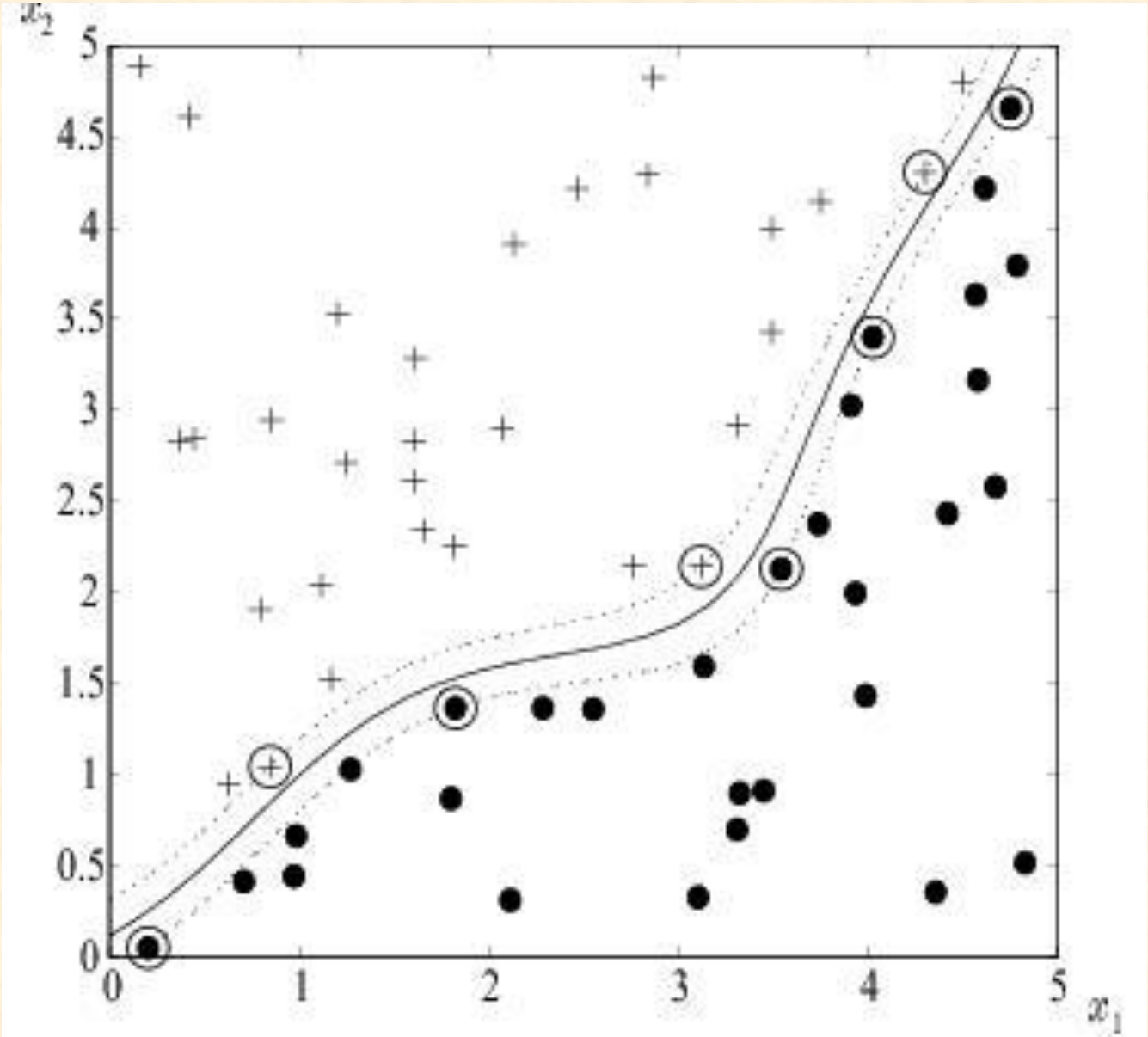
$$\underline{w} = \sum_{i=1}^{N_s} \lambda_i y_i \underline{\varphi}(\underline{x}_i)$$

- Βήμα 3: Καταχώρησε το $\underline{x}$ στην

$$\omega_1(\omega_2) \ \text{αν} \ g(\underline{x}) = \sum_{i=1}^{N_s} \lambda_i y_i K(\underline{x}_i, \underline{x}) + w_0 > (<) 0$$

- Η αρχιτεκτονική SVM




x_2
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
x_1