Modern Methods of Statistical Learning sf2935
Lecture 1:
Introduction to Learning Theory

Timo Koski

TK

2017-11-01

KTH Matematik

Overview of the Lecture

The methods of statistical learning provide a heterogeneous collection of points of view, tasks, algorithms, models and probability distributions. This lecture tries to give a few comprehensive concepts and presents the historically first instance of learning theory, perceptron and perceptron algorithm.

KTH VETENSKAP OCH KONST

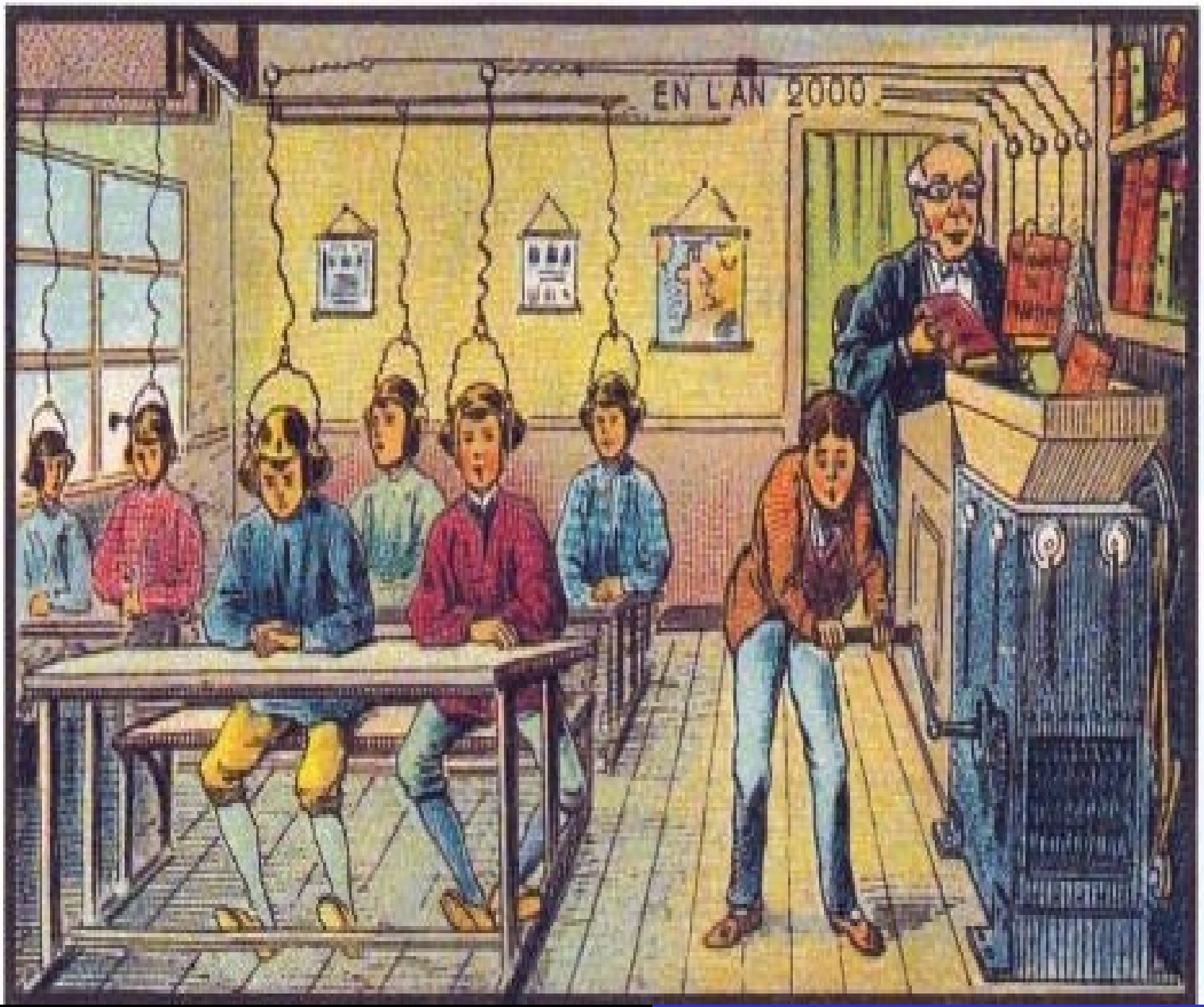
KTH Matematik

Your Learning Outcomes

- Supervised learning
- Unsupervised learning
- Geometry of classification: hyperplanes, scalar products, orthogonality
- Perceptron learning, training set
- ANN= artificial neural networks
- Perceptron Convergence (the proof of the convergence theorem is for specially interested, and is not required reading).

KTH Matematik

E-learning



Statistical Learning = Machine Learning + Statistics

Machine Learning means development of algorithms and techniques that allow computers to learn to

1. record observations about a phenomenon
2. build a model of this phenomenon
3. predict the future of this phenomenon

Statistics gives

- a formal definition of machine learning
- some guarantees of expected results
- suggestions for new or improved modelling tools

KTH VETENSKAP OCH KONST

KTH Matematik

Statistical Learning = Machine Learning + Statistics

Machine Learning: a phenomenon is recorded via observations $\{\mathbf{z}_i\}_{i=1}^n$ with $\mathbf{z}_i \in \mathcal{Z}$. There are two generic situations

1. *Unsupervised learning*
   - no predefined structure in $\mathcal{Z}$.
   - the goal is to find some structures: clusters, association rules, estimating probability distributions/densities
2. *Supervised learning*
   - $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, the components are not exchangeable
   - $\mathbf{z} = \mathbf{x} \times \mathbf{y} \in \mathcal{X} \times \mathcal{Y}$.
   - modelling: finding how $\mathbf{x}$ and $\mathbf{y}$ are related.
   - the goal is to make prediction: given $\mathbf{x}$, find a reasonable value for $\mathbf{y}$ such that $\mathbf{z} = \mathbf{x} \times \mathbf{y}$ is compatible with the phenomenon.

KTH VETENSKAP OCH KONST
KTH Matematik

Unsupervised learning

- Learning what normally happens (no teacher)
- No output
- Clustering: Grouping similar instances

KTH VETENSKAP OCH KONST
KTH Matematik

Other types

SemiSupervised Learning, a class of supervised learning techniques that also make use of large amount of non-structured data $\{\mathbf{z}_i\}_{i=1}^{n_1}$ for training and typically a small amount of data $\{\mathbf{x}_i \times \mathbf{y}_i\}_{i=1}^{n_2}$. Semi-supervised learning falls between unsupervised learning and supervised learning.

-

Active learning is said to be a special case of semi-supervised machine learning in which a learning algorithm is able to interactively query some information source to obtain the desired outputs at new data points. In statistics it is sometimes also called optimal experimental design.

-

KTH VETENSKAP OCH KONST

KTH Matematik

Other types

Reinforcement Learning is (Wikipedia) an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

-

Learning by Doing

-

Supervised learning

Several types of $\mathcal{Y}$ can be considered:

1. $\mathcal{Y} = \{1, 2, \ldots, q\}$, classification *i q classes, e.g., to tell whether a patient has a certain kind of disease.*
2. $\mathcal{Y} = R^q$, regression, *e.g., to calculate the price* $\mathbf{y}$ *of a house based on some characteristics* $\mathbf{x}$ *(like the neighbourhood, year of construction, architectural style).*
3. $\mathcal{Y}$ *is something complex but structured.*

The modelling difficulty increases with the complexity of $\mathcal{Y}$.

KTH VETENSKAP OCH KONST

KTH Matematik

Modeling: f from $\mathcal{X}$ to $\mathcal{Y}$

- Given a data set or *training set* $\{\mathbf{x}_i \times \mathbf{y}_i\}_{i=1}^n$ a machine learning method builds up a model f from $\mathcal{X}$ to $\mathcal{Y}$.

Example

Conditional expectation or regression function:

$$f(\mathbf{x}) = E(\mathbf{Y} \mid \mathbf{X} = \mathbf{x})$$

where $\mathbf{Y}$ *and* $\mathbf{X}$ *are random variables with values in* $\mathcal{Y}$ *and* $\mathcal{X}$*, respectively.*

- A Good Model should be such that for all $i = 1, \ldots, n$

$$f(\mathbf{x}_i) \approx \mathbf{y}_i$$

In statistics, one chooses often to measure the quality of modelling by Mean Square Error (assuming $\mathcal{Y} \subseteq R$))

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{y}_i - f(\mathbf{x}_i))^2 .$$

KTH VETENSKAP OCH KONST
KTH Matematik

The MSE in

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{y}_i - f\left(\mathbf{x}_i\right)\right)^2.$$

is computed using the training data that was used to fit the model, and so should more accurately be referred to as the *training MSE*. In general, we do not really care how well the method works on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data. This is called generalization error or test error.

OUTLINE OF TOPICS:

- perceptron
- perceptron training
- artificial neural network
- perceptron convergence

KTH Matematik

Frank Rosenblatt, 1928−1969, suggested the first model of a learning machine called **perceptron**. From Wikipedia:

In a 1958 press conference organized by the US Navy, Rosenblatt made statements about the perceptron that caused a heated controversy; based on Rosenblatt's statements, The New York Times reported the perceptron to be **"the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."**

KTH Matematik

LEARNING MACHINE: NOTATIONS

Next we start studying linear learning machines, also known as *perceptrons*.

$$\mathcal{X} = \quad \text{INPUT SPACE},$$

which is also a vector space of finite dimension and an inner product space. Thus for $\mathbf{w}$ and $\mathbf{x}$ in $\mathcal{X}$ we have an inner product

$$\ll \mathbf{w}, \mathbf{x} \gg (= \sum_{i=1}^{d} w_i x_i),$$

$$\ll \mathbf{w}, \mathbf{w} \gg \geq 0,$$

and we have the norm

$$\|\mathbf{w}\| = \sqrt{\ll \mathbf{w}, \mathbf{w} \gg}$$

KTH VETENSKAP OCH KONST
KTH Matematik

LEARNING MACHINE: NOTATIONS

$\mathcal{Y} =$ OUTPUT DOMAIN OR HYPOTHESIS SPACE

Examples of output domains are $\{+1, -1\}$ (output domain for binary classification), or $\{1, 2, \ldots, q\}$ (multiple classification).

We consider functions $X \overset{f}{\mapsto} R$ such that

$$f(\mathbf{x}) = \ll \mathbf{w}, \mathbf{x} \gg + b$$

where $\mathbf{w}$ is a vector called *weight vector*, and b is a real number called *bias*.

KTH Matematik

Output domains are $\{+1, -1\}$

Most of the theory to be presented deals with the binary output domain represented as $\{+1, -1\}$. This is, of course, restrictive, but this has a lot of practical relevance.

KTH VETENSKAP OCH KONST

KTH Matematik

$$| \ll \mathbf{w}, \mathbf{x} \gg - \ll \mathbf{w}, \mathbf{y} \gg |$$

$$= | \ll \mathbf{w}, \mathbf{x} - \mathbf{y} \gg |$$

by rules of inner product, and then

$$\leq \|\mathbf{w}\| \|\mathbf{x} - \mathbf{y}\|$$

by Cauchy-Schwartz.
Hence whenever the difference $\|\mathbf{x} - \mathbf{y}\|$ is small, the difference in the real-valued outputs is also small.

A *hyperplane* is an affine[1] subspace of dimension $n-1$, if n is the dimension of X, which divides the space into two half spaces. We consider the following *hyperplane* $D \subset X$:

$$D = \{\mathbf{x} \in X \mid f(\mathbf{x}) = 0\}$$

where

$$f(\mathbf{x}) = \ll \mathbf{w}, \mathbf{x} \gg + b$$

Take $\mathbf{x}_1 \in D$ and $\mathbf{x}_2 \in D$. Then

$$\ll \mathbf{w}, \mathbf{x}_1 \gg = \ll \mathbf{w}, \mathbf{x}_2 \gg \Leftrightarrow$$

$$\ll \mathbf{w}, \mathbf{x}_1 - \mathbf{x}_2 \gg = 0.$$

In other words, $\mathbf{w}$ is orthogonal to any vector in D.

[1] = parallel to a linear subspace, a linear subspace contains $\mathbf{0}$

KTH Matematik

$$f(\mathbf{x}) = \ll \mathbf{w}, \mathbf{x} \gg + b$$

$$D = \{\mathbf{x} \in X \mid f(\mathbf{x}) = 0\}$$

A hyperplane is an affine subspace of dimension $n-1$, which divides the space into two half spaces, here $\mathcal{R}_1$ and $\mathcal{R}_2$, as

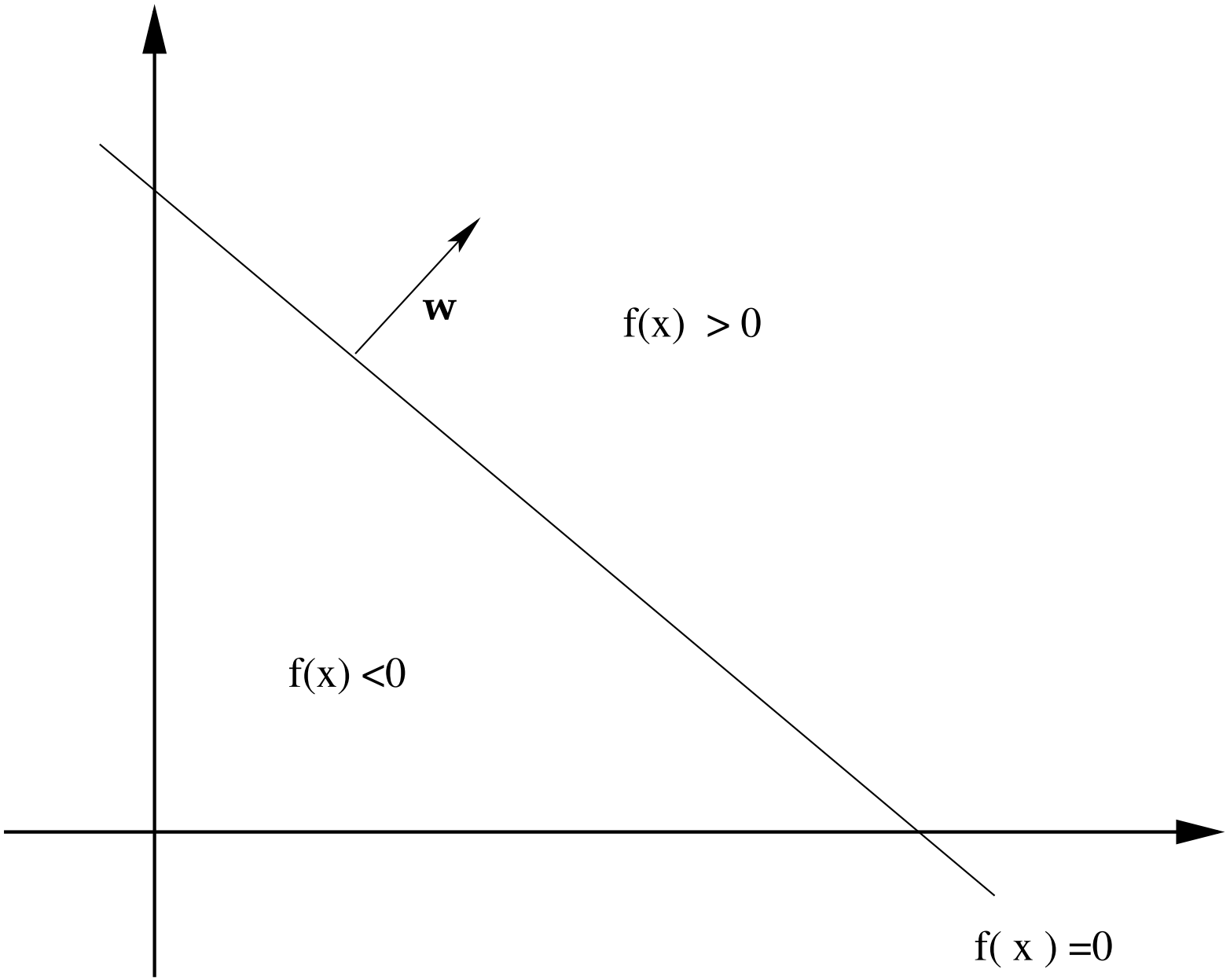
$$\mathcal{R}_1 = \{\mathbf{x} \in X \mid f(\mathbf{x}) > 0\}$$

$$\mathcal{R}_2 = \{\mathbf{x} \in X \mid f(\mathbf{x}) < 0\}$$

By the preceding $\mathbf{w}$ is orthogonal to any vector in D and points to $\mathcal{R}_1$.

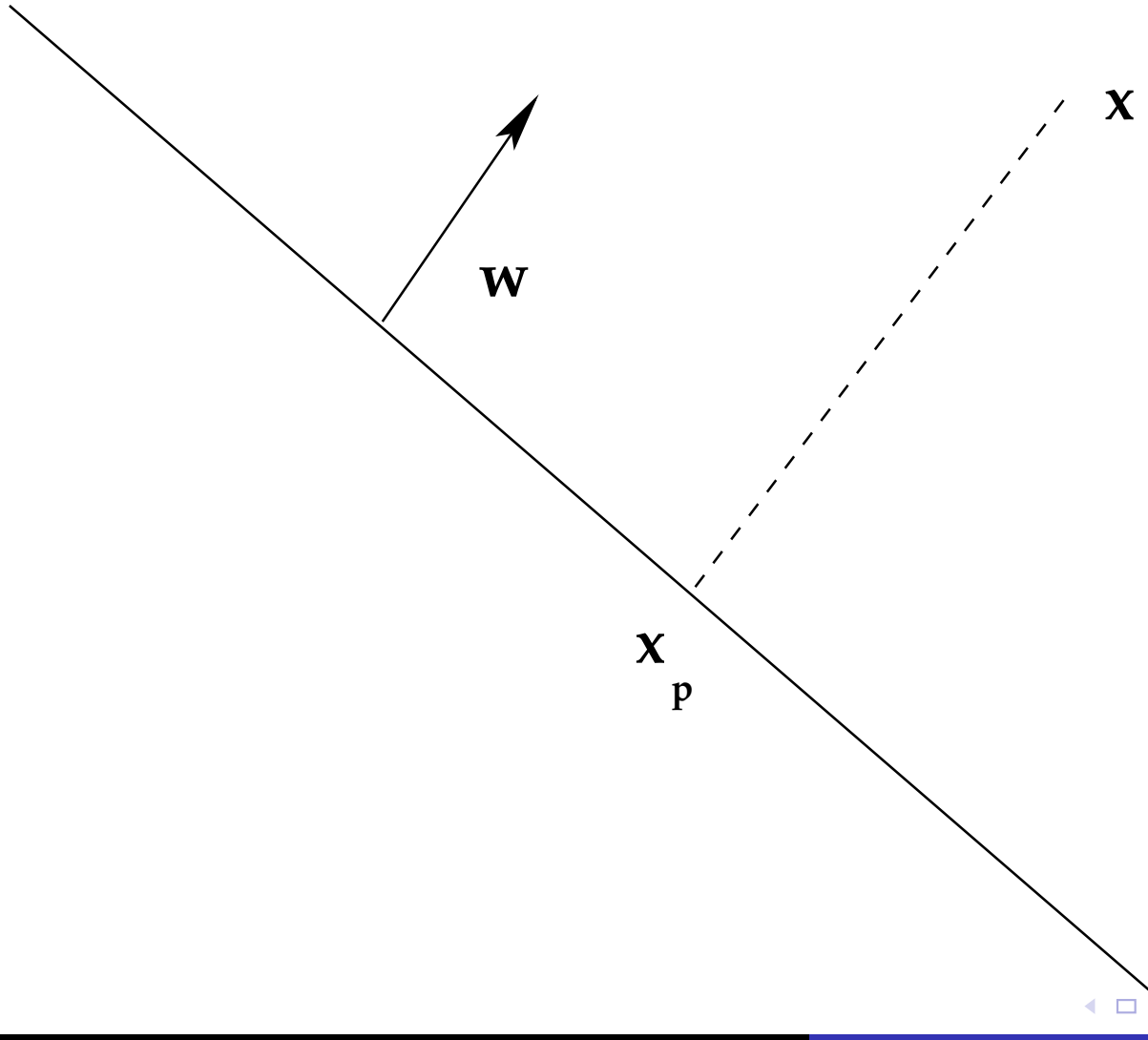
KTH VETENSKAP OCH KONST
KTH Matematik

LINEAR LEARNING MACHINES: geometry



Let $\mathbf{x}_p = P_D(\mathbf{x}_p)$ be the orthogonal projection of $\mathbf{x}$ onto D, i.e.,

$$\ll \mathbf{x}_p, \mathbf{x} - \mathbf{x}_p \gg = 0$$



$$\mathbf{x} = \mathbf{x}_p + r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

where r_x is a number (to be determined) and $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ is a unit length vector in the direction of $\mathbf{w}$. We need to establish the following:

$$f(\mathbf{x}) = r_x \|\mathbf{w}\|$$

$$r_0 = \frac{f(\mathbf{0})}{\|\mathbf{w}\|} = \frac{b}{\|\mathbf{w}\|}$$

We compute the squared length

$$\|\mathbf{x} - \mathbf{x}_p\|^2 = \|r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}\|^2 = \left(\frac{r_x}{\|\mathbf{w}\|}\right)^2 \|\mathbf{w}\|^2$$

$$\Leftrightarrow$$

$$\|\mathbf{x} - \mathbf{x}_p\|^2 = r_x^2$$

$$\Leftrightarrow$$

$$\|\mathbf{x} - \mathbf{x}_p\| = |r_x|$$

Hence r_x is the signed distance of $\mathbf{x}$ to D.

In addition, with

$$f(\mathbf{x}) = \ll \mathbf{w}, \mathbf{x} \gg + b =$$

$$= \ll \mathbf{w}, \mathbf{x}_p + r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg + b = \ll \mathbf{w}, \mathbf{x}_p \gg + \ll \mathbf{w}, r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg + b$$

$$= \underbrace{\ll \mathbf{w}, \mathbf{x}_p \gg + b}_{= f(\mathbf{x}_p)} + \ll \mathbf{w}, r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg$$

$$= \underbrace{f(\mathbf{x}_p)}_{=0} + \ll \mathbf{w}, r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg$$

$$= \ll \mathbf{w}, r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg$$

KTH VETENSKAP OCH KONST

KTH Matematik

$$= \ll \mathbf{w}, r_x \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \gg$$

$$= r_x \cdot \frac{1}{\|\mathbf{w}\|} \ll \mathbf{w}, \mathbf{w} \gg = r_x \|\mathbf{w}\|.$$

In other words we have obtained

$$f(\mathbf{x}) = r_x \|\mathbf{w}\|$$

$$\Leftrightarrow$$

$$r_x = \frac{f(\mathbf{x})}{\|\mathbf{w}\|}$$

Furthermore, $f(\mathbf{0}) = \ll \mathbf{w}, \mathbf{0} \gg + b = 0 + b = b$, i.e.,

$$r_0 = \frac{f(\mathbf{0})}{\|\mathbf{w}\|} = \frac{b}{\|\mathbf{w}\|}$$

Since $f(\mathbf{x}) = r_x\|\mathbf{w}\|$, if $r_x > 0$, then $\mathbf{x}$ is on the positive side, and if $r_x < 0$, then $\mathbf{x}$ is on the negative side. Here we take

$$\mathcal{R}_1 = \{\mathbf{x} \in X \mid f(\mathbf{x}) > 0\}$$

as the positive side of D

$$\mathcal{R}_2 = \{\mathbf{x} \in X \mid f(\mathbf{x}) < 0\}$$

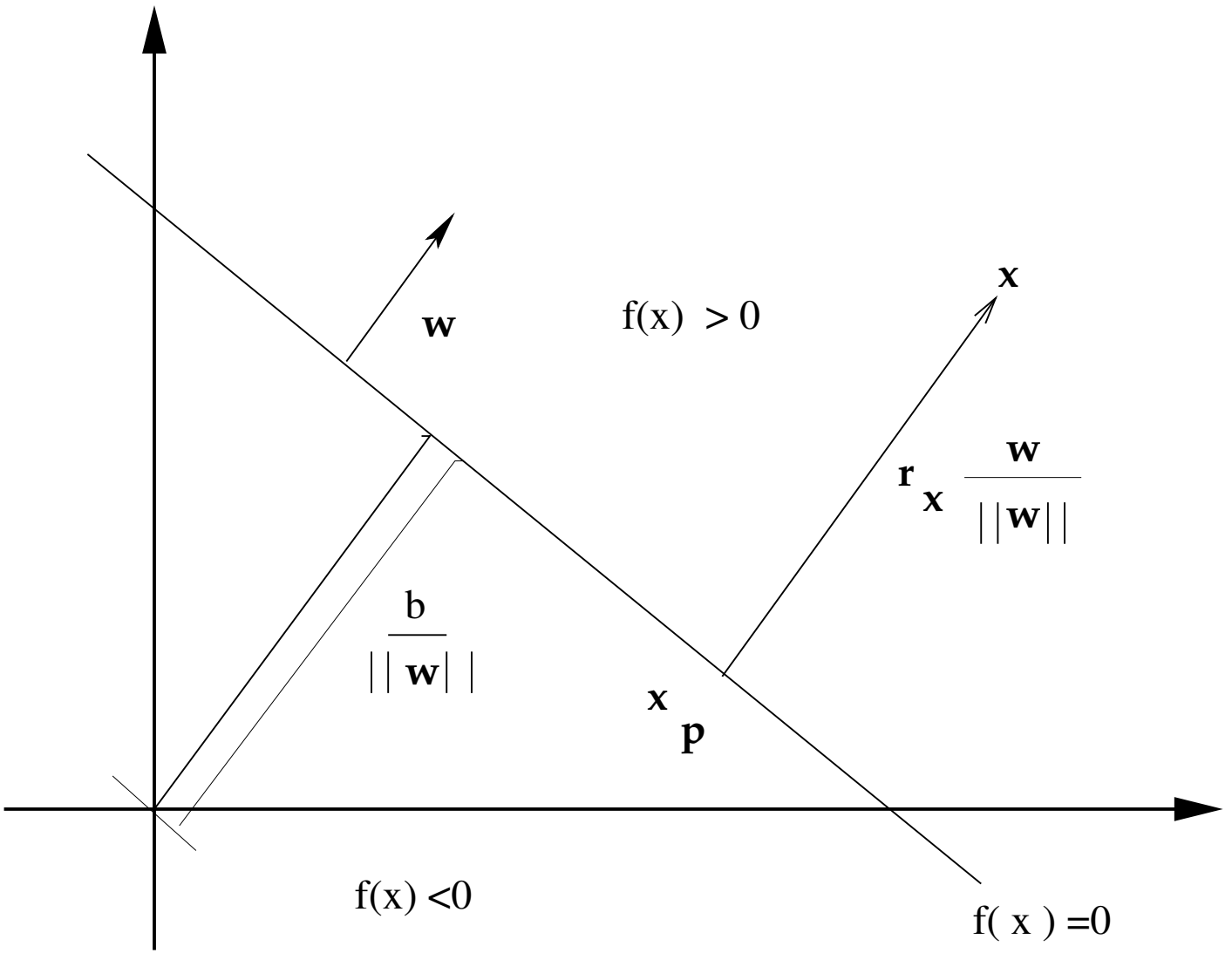
as the negative side of D.

Since r_0 is the signed distance of $\mathbf{0}$ to D, and

$$r_0 = \frac{b}{\|\mathbf{w}\|}$$

we see that varying b moves D parallell to itself. If $b > 0$, then $\mathbf{0}$ is in $\mathcal{R}_1$, and if $b < 0$, then $\mathbf{0}$ is in $\mathcal{R}_2$.

$r_0 = \frac{b}{\|\mathbf{w}\|}$ is the signed distance of $\mathbf{0}$ to D



LINEAR LEARNING MACHINE a.k.a. PERCEPTRON

Let

$$\text{sign}(x) = \begin{cases} +1 & 0 \leq x, \\ -1 & x < 0 \end{cases}$$

(Note that the definition of sign for $x = 0$ is arbitrary.) Then a linear learning machine for a binary classification task (or for a binary *target concept*), a.k.a. **perceptron**, processes an input $\mathbf{x}$ by giving the output

$$y = \text{sign}\left(f\left(\mathbf{x}\right)\right), \quad y \in \{+1, -1\}$$

Note that $\text{sign}\left(f\left(\mathbf{x}\right)\right)$ is a map from $\mathcal{X}$ to $\mathcal{Y}$. This requires, of course, that we have, somehow, set the values of $\mathbf{w}$ and b. The learning of the weights $\mathbf{w}$ and the bias b from examples is our next topic of study.

KTH Matematik

Find $(\mathbf{w}, b)$ using $\mathcal{S}$.

Let $\mathbf{x}_i \in X$, $y_i \in Y = \{+1, -1\}$, $i = 1, \ldots l$ be l pairs of *examples* of a binary target concept. Then

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$$

is called a *training set*. Then we consider the following learning task

KTH Matematik

Supervised learning of perceptrons

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$$

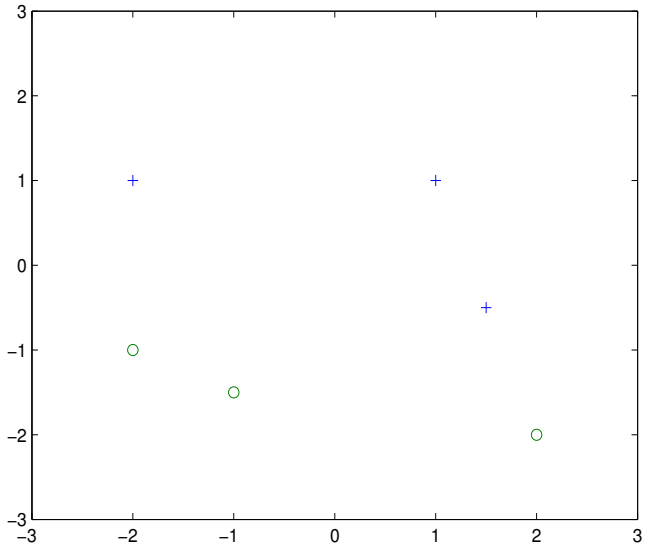
Task: Find $(\mathbf{w}, b)$ using $\mathcal{S}$ so that $f(\mathbf{x}_i) = y_i$ for all i, i.e., all inputs are correctly classified.
Questions:

1. Does a solution to this task exist ?
2. If a solution exists, is there an algorithm that finds it in a finite number of steps ?

KTH VETENSKAP OCH KONST
KTH Matematik

Example in R^2: $+1 \mapsto +$, $-1 \mapsto o$

Point nr l	Input $\mathbf{x}(l) = (x_1(l), x_2(l))$		Labels y
l	$x_1(l)$	$x_2(l)$	$y(l)$
1	1.5	-0.5	$+1$
2	1	1	$+1$
3	-2	1	$+1$
4	-1	-1.5	-1
5	2	-2	-1
6	-2	-1	-1



Find w_1, w_2 and b so that

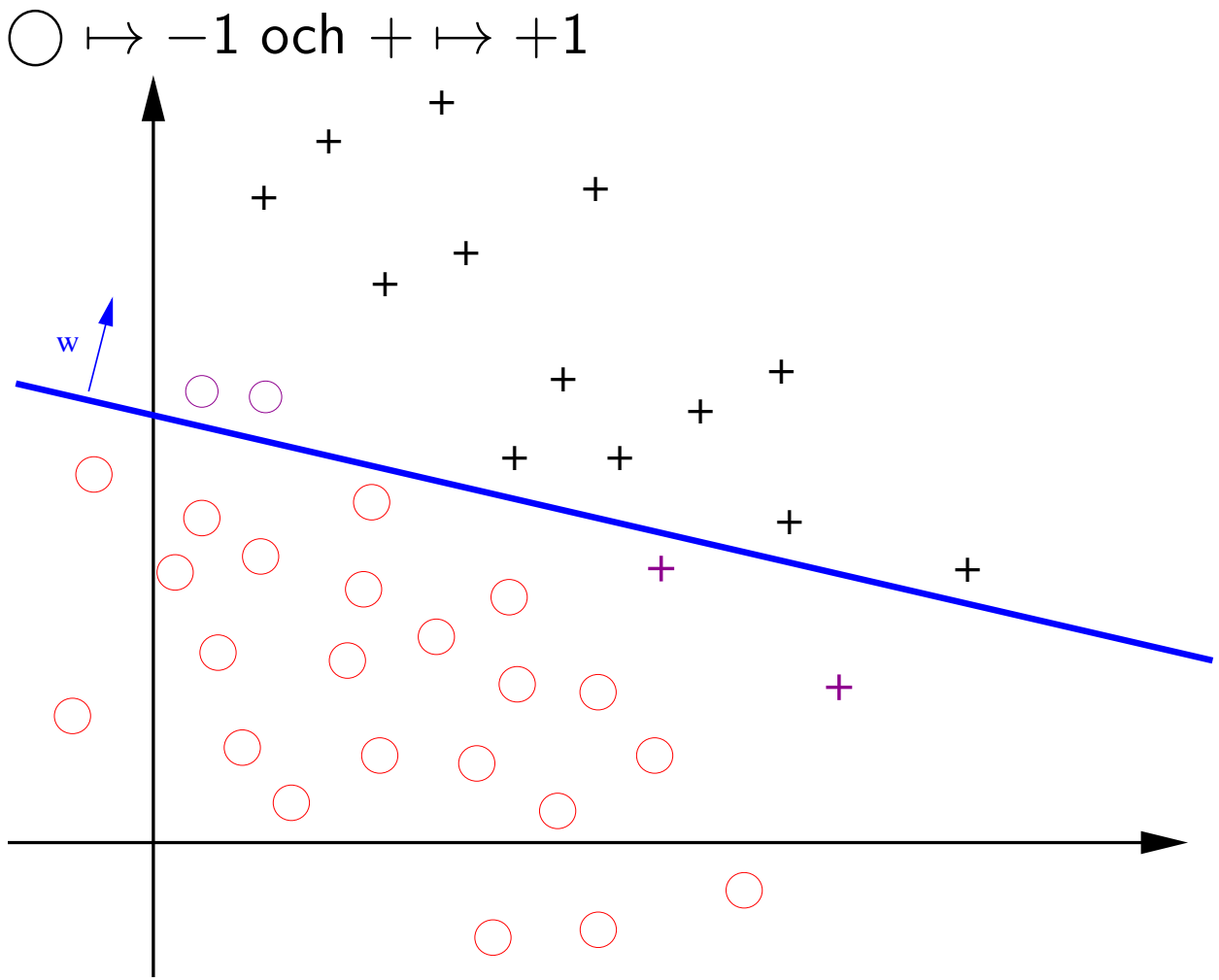
$$f(\mathbf{x}(l)) = w_2 x_2(l) + w_1 x_1(l) + b$$

have the right signs (and therefore the right label) for every $\mathbf{x}(l)$, i.e.,

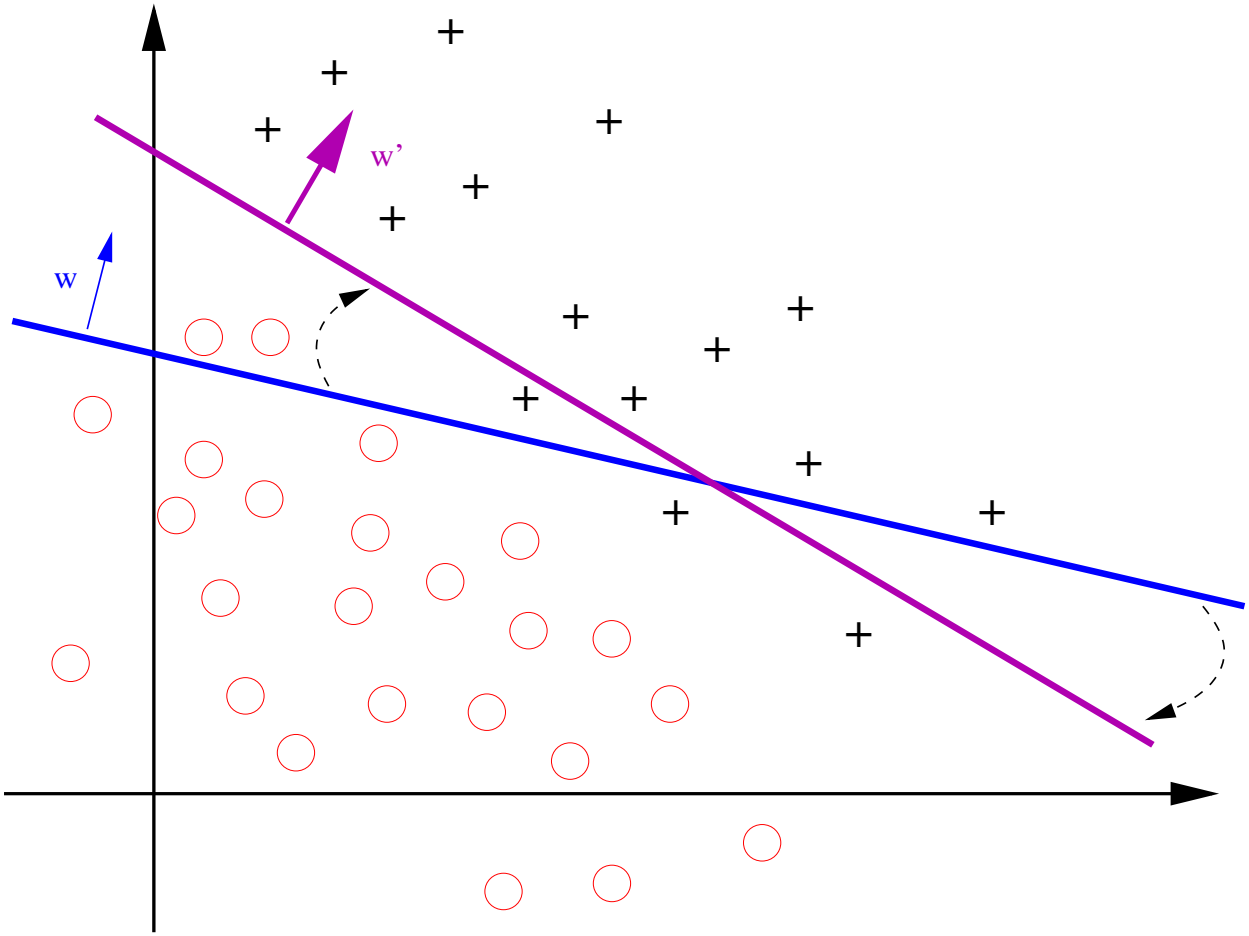
$$\text{sign}(f(\mathbf{x}(l))) = y(l)$$

hold for every l.

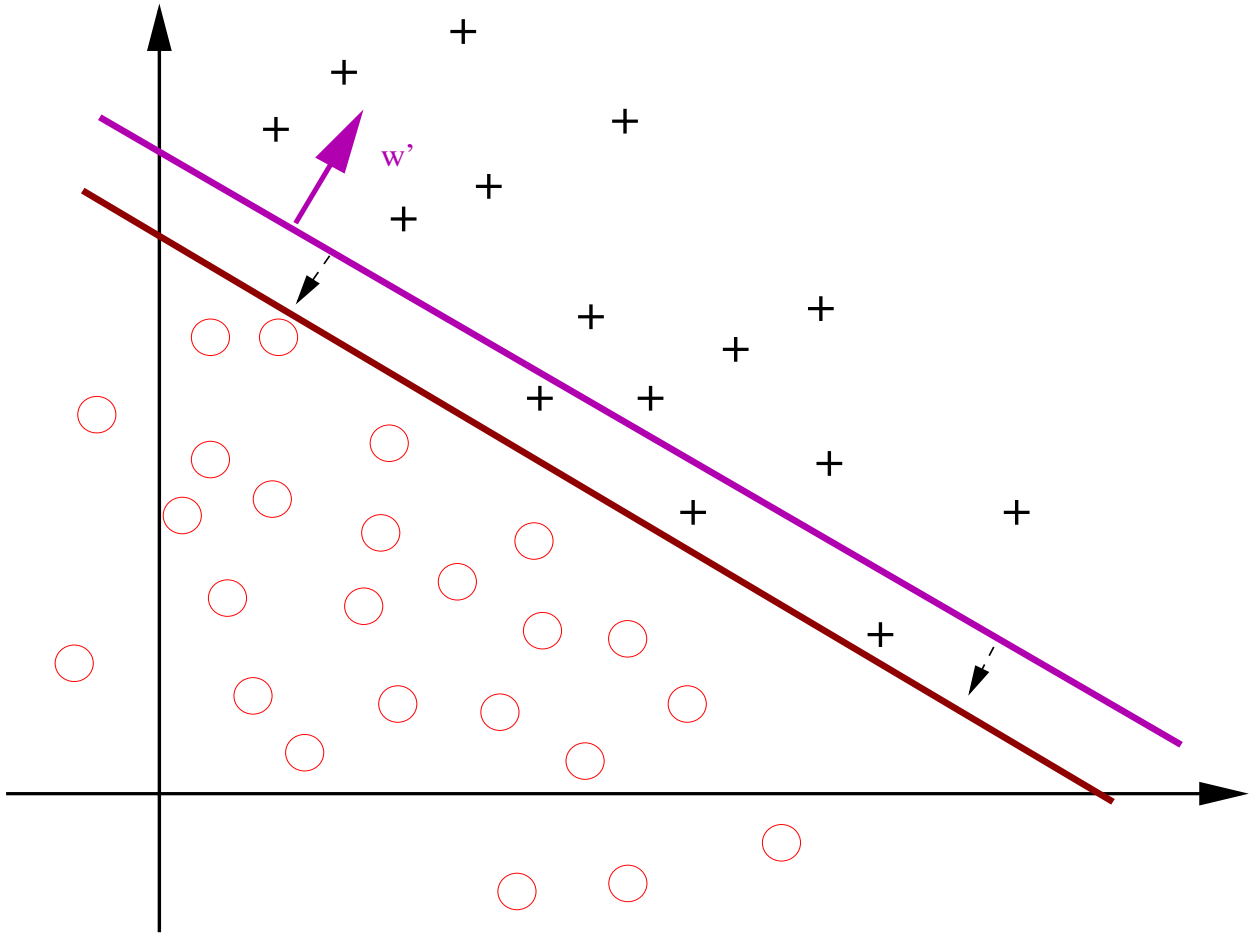
This is an instance of *supervised learning*.


◯ ↦ −1 och + ↦ +1
w

Change of direction



Parallel shift



An algorithm doing these two steps

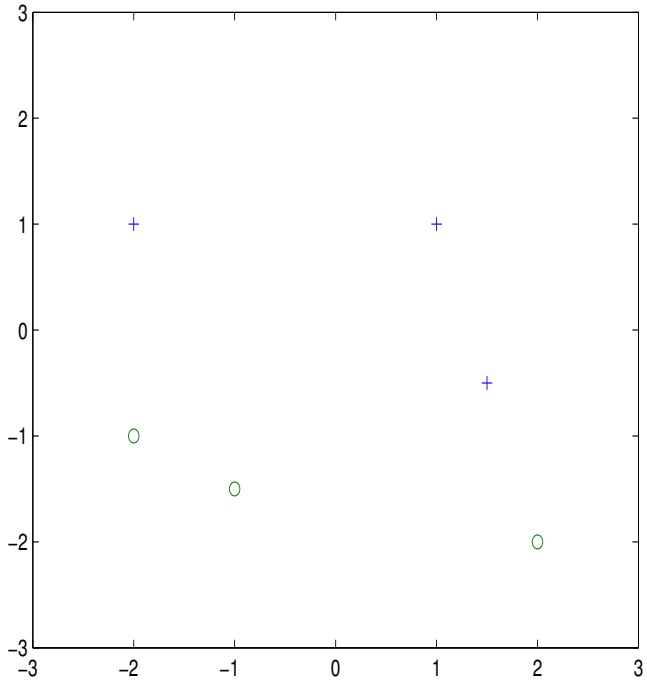
- Choose initial values w_1, w_2 and b at random.
- For every l: if $f(\mathbf{x}(l))$ has right sign for these weights and bias, do nothing.
- If $\text{sign}(f(\mathbf{x}(l))) \neq y(l)$, up-date bias and weights as follows $\eta = 0.2$):

$$\begin{aligned} b &\leftarrow b + \eta y(l) \\ w_1 &\leftarrow w_1 + \eta y(l) x_1(l) \\ w_2 &\leftarrow w_2 + \eta y(l) x_2(l) \end{aligned}$$

- Continue by passing through $\mathbf{x}(l)$ until all have correct label.

KTH VETENSKAP OCH KONST
KTH Matematik

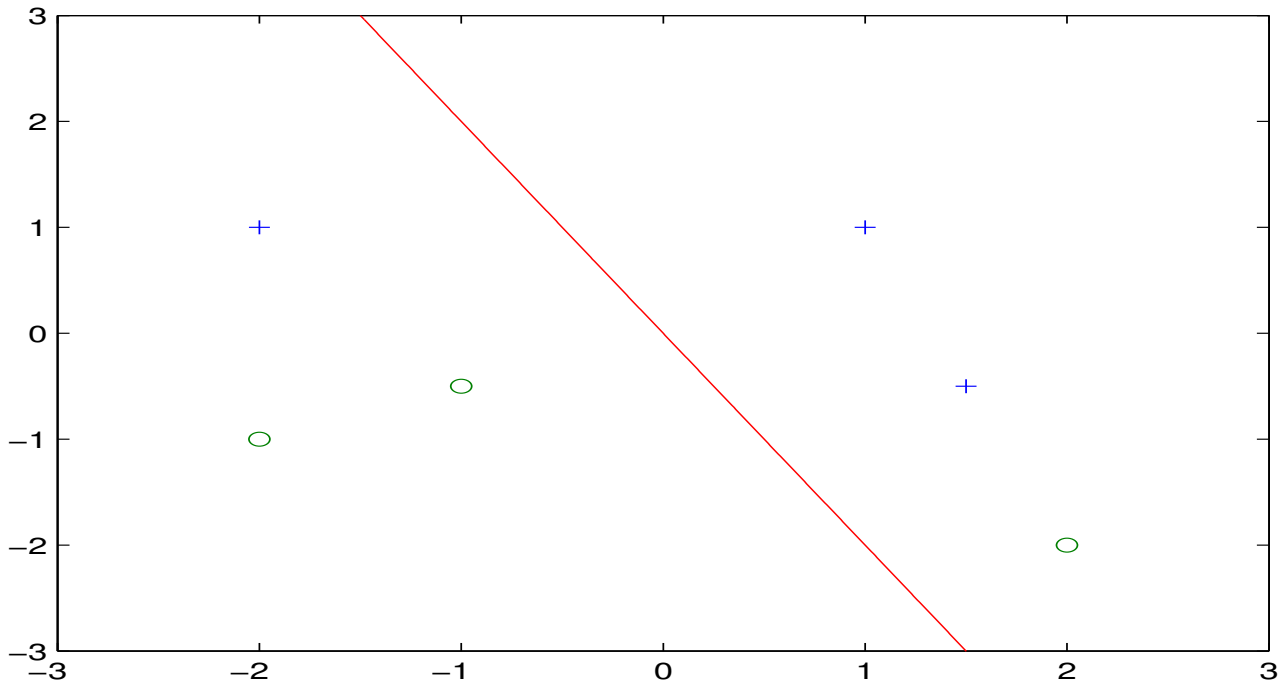
Choose initial values $b = 0, w_1 = 1, w_2 = 0.5$ at random, i.e.,



$$f(\mathbf{x}) = 0.5x_2 + x_1$$

Example

We plot the data points and the straight line $0.5x_2 + x_1 = 0$ d.v.s. $x_2 = -2x_1$



Example: Up dates

There is an error in $l = 5$ with $\mathbf{x}(5) = (2, -2)$, $y(5) = -1$ The rule

$$\begin{aligned} b &\leftarrow b + \eta y(5) \\ w_1 &\leftarrow w_1 + \eta y(5) x_1(5) \\ w_2 &\leftarrow w_2 + \eta y(5) x_2(5) \end{aligned}$$

gives with $b = 0, w_1 = 1, w_2 = 0.5, \eta = 0.2$

$$\begin{aligned} -0.2 &= 0 + 0.2 \cdot (-1) \\ 0.6 &= 1 + 0.2 \cdot (-1) \cdot 2 \\ 0.9 &= 0.5 + 0.2 \cdot (-1) \cdot (-2) \end{aligned}$$

i.e., we have

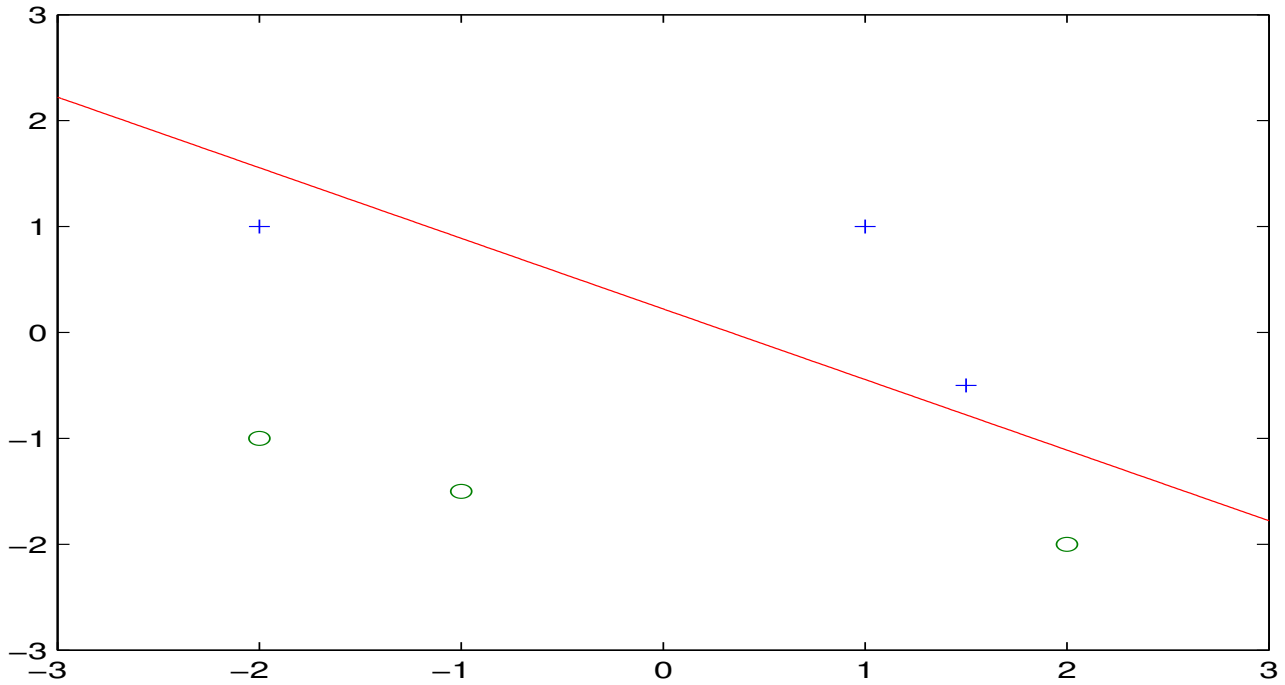
$$f(\mathbf{x}) = 0.9 x_2 + 0.6 x_1 - 0.2$$

i.e., $x_2 = -\frac{0.6}{0.9} x_1 + \frac{0.2}{0.9}$.

KTH VETENSKAP OCH KONST
KTH Matematik

After the first up date

Data points and $x_2 = -\frac{0.6}{0.9}x_1 + \frac{0.2}{0.9}$



Exeample : up date

Error in $l = 3$ with $\mathbf{x}(3) = (-2, 1)$, $y(3) = 1$ and $b = -0.2, w_1 = 0.6, w_2 = 0.9, \eta = 0.2$

$$\begin{aligned} 0 &= 0.2 + (-0.2) \cdot 1 \\ 0.2 &= 0.6 + 0.2 \cdot 1 \cdot (-2) \\ 1.1 &= 0.9 + 0.2 \cdot 1 \cdot 1 \end{aligned}$$

which gives

$$f(\mathbf{x}) = 1.1x_2 + 0.2x_1$$

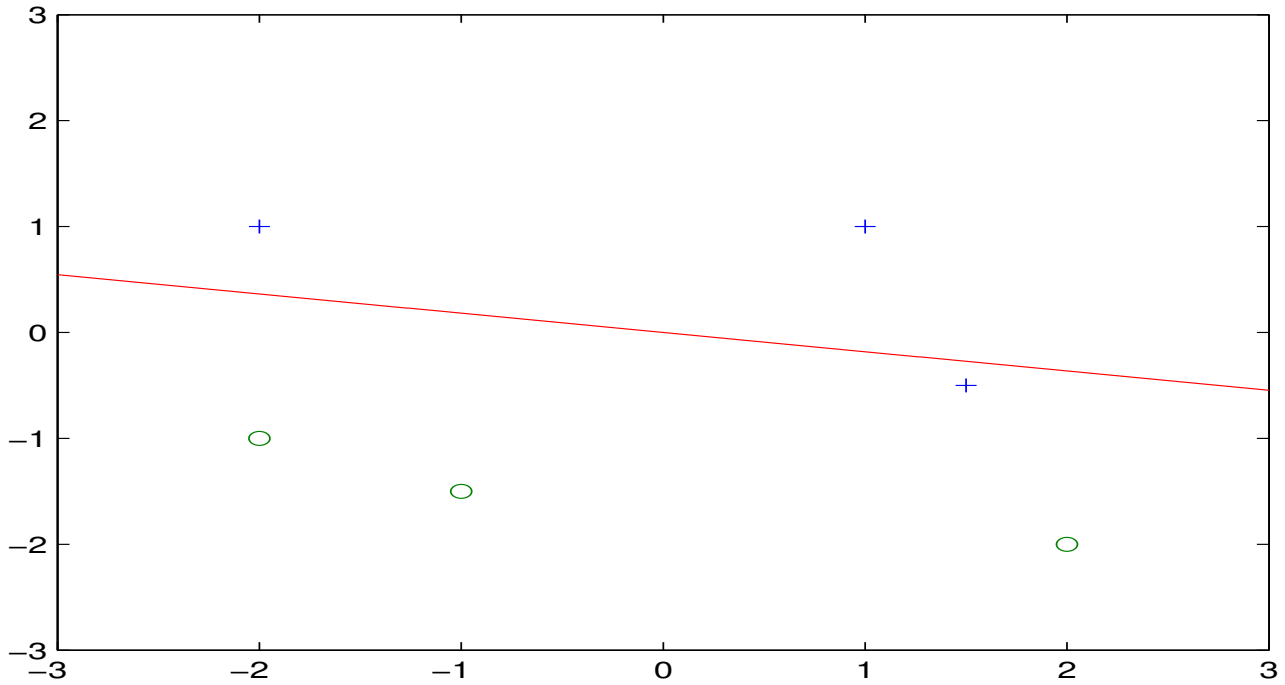
i.e., $x_2 = -\frac{0.2}{1.1}x_1$.

KTH VETENSKAP OCH KONST

KTH Matematik

Example in R^2: andra uppdateringen

Data points and $x_2 = -\frac{0.2}{1.1}x_1$.



Example in R^2: uppdatering

Now it went wrong in $l = 1$ with $\mathbf{x}(1) = (1.5, -0.5)$, $y(1) = 1$.
With $b = 0, w_1 = 0.2, w_2 = 1.1, \eta = 0.2$

$$\begin{aligned} 0.2 &= 0 + 0.2 \cdot 1 \\ 0.5 &= 0.2 + 0.2 \cdot 1 \cdot 1.5 \\ 1.0 &= 1.1 + 0.2 \cdot 1 \cdot (-0.5) \end{aligned}$$

we get

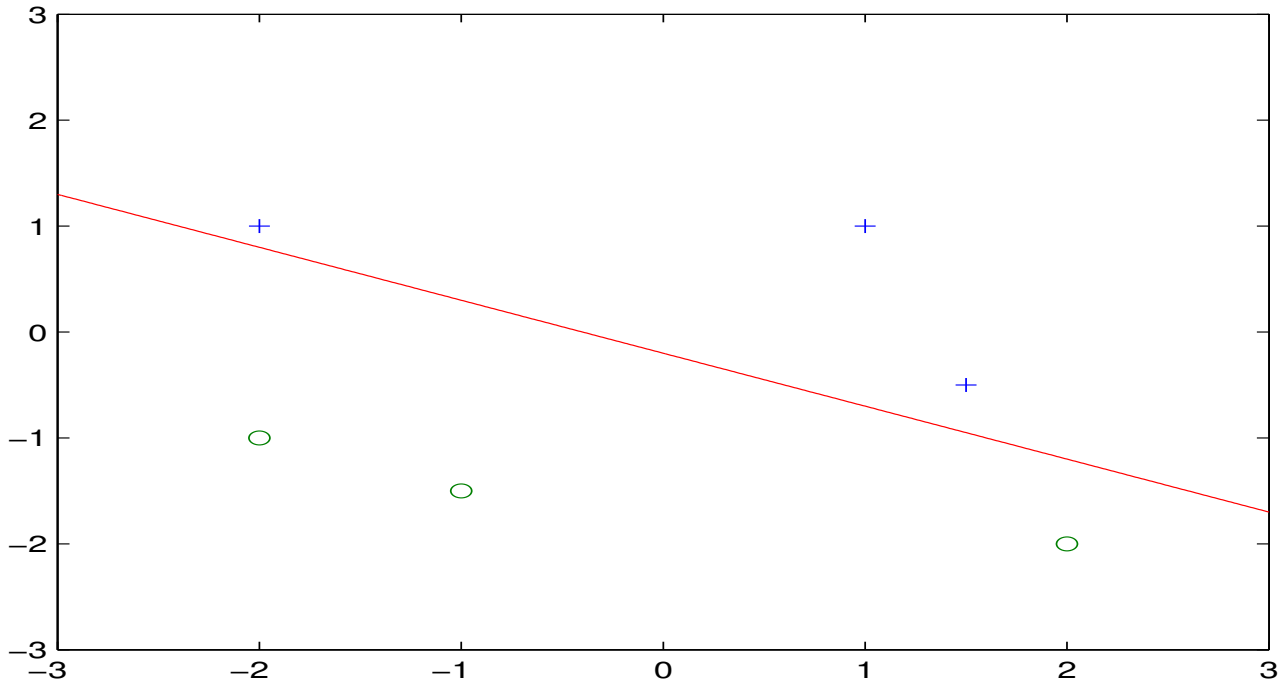
$$f(\mathbf{x}) = 1.0x_2 + 0.5x_1 + 0.2$$

i.e., $x_2 = -0.5x_1 - 0.2$.

KTH VETENSKAP OCH KONST
KTH Matematik

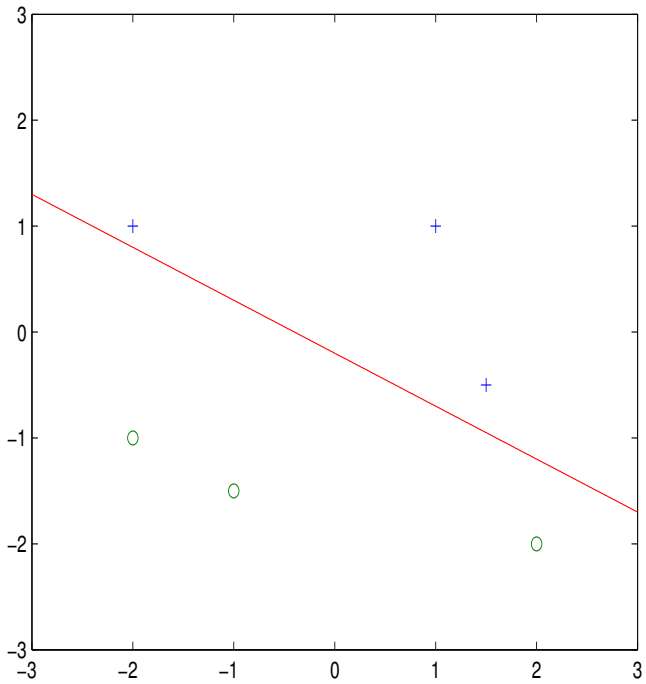
Example: the third up date

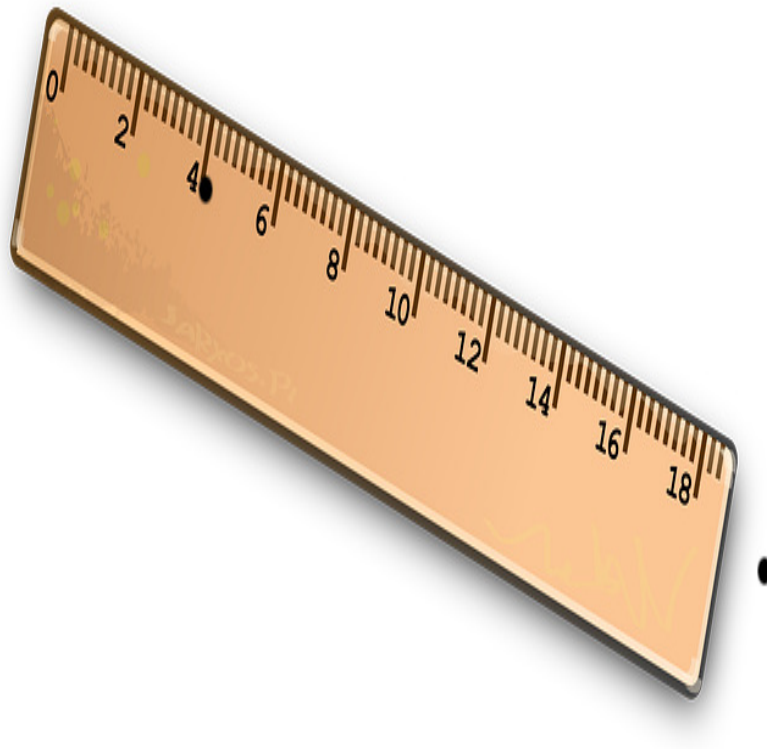
Data points and $x_2 = -0.5x_1 - 0.2$.



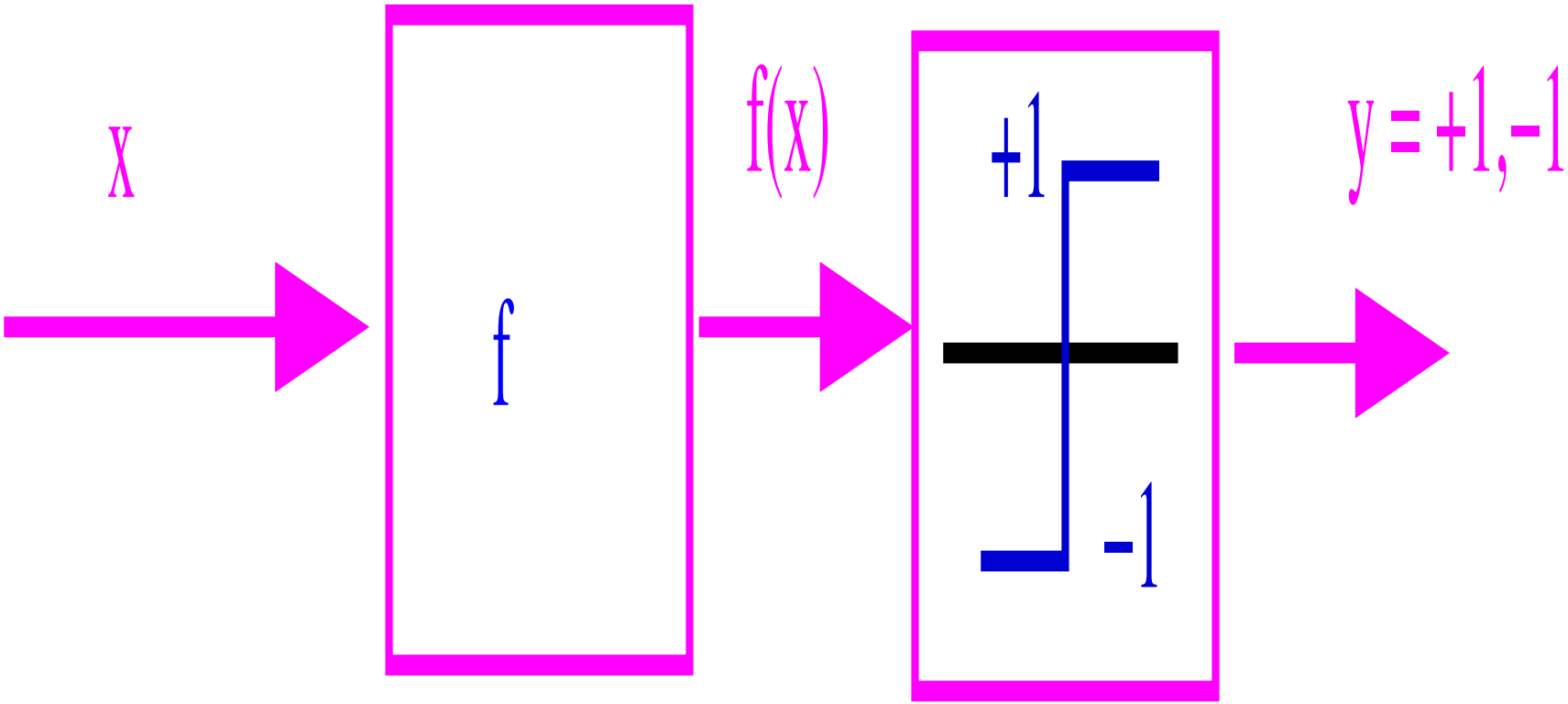
Example in R^2: Correct ! The algorithm stops !

Data points and $x_2 = -0.5x_1 - 0.2$.

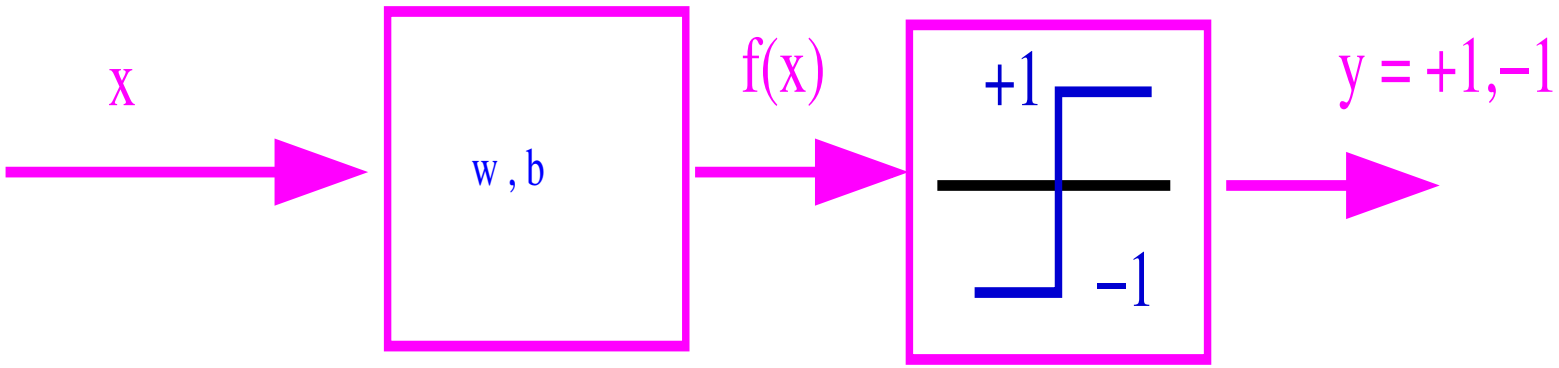


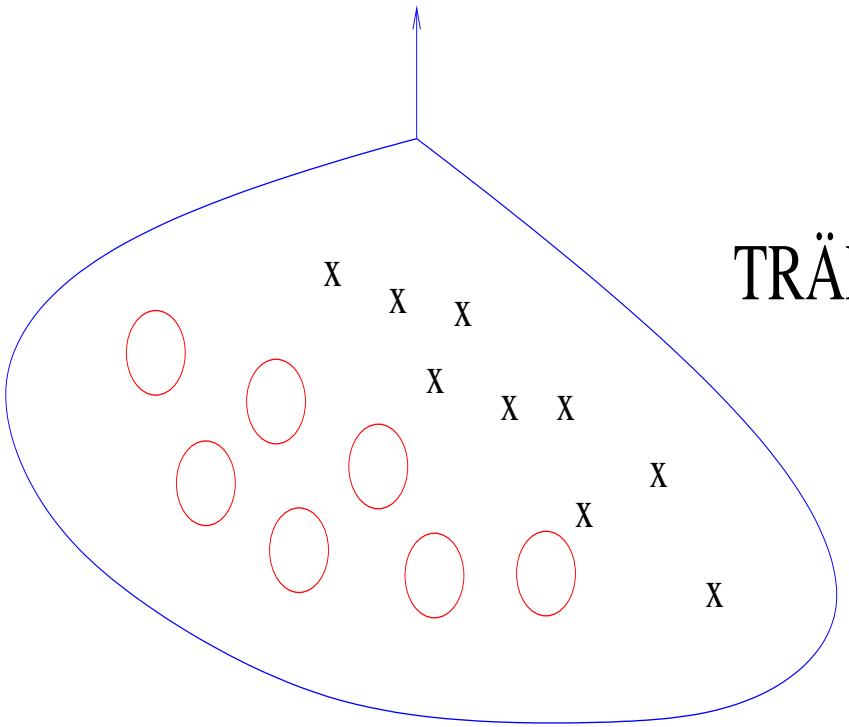


System diagram



Learning





TRÄNINGSMÄNGD

KTH VETENSKAP OCH KONST

KTH Matematik

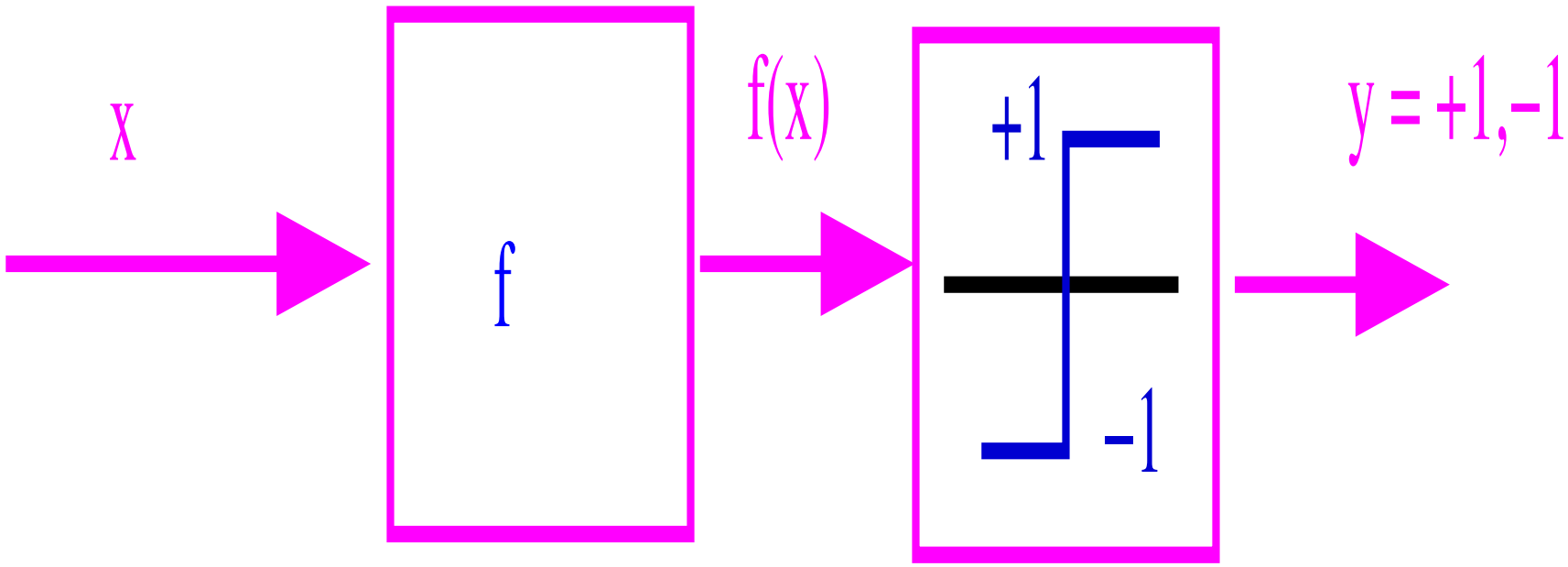
Learning: training

The examples

Number l	Data $\mathbf{x}(l) = (x_1(l), x_2(l))$		labels y
l	$x_1(l)$	$x_2(l)$	$y(l)$
1	1.5	-0.5	+1
2	1	1	+1
3	-2	1	+1
4	-1	-1.5	-1
5	2	-2	-1
6	-2	-1	-1

have been **stored** med in $b = 0.2, w_1 = 0.5, w_2 = 1.0$.

Running the perceptron after training



We feed in a **new** **x** out comes a prediction of the label: $+1$,or -1

LINEAR LEARNING MACHINE a.k.a. PERCEPTRON

The perceptron, i.e.,

$$y = \text{sign}\left(f\left(\mathbf{x}\right)\right), \quad y \in \{+1, -1\}$$

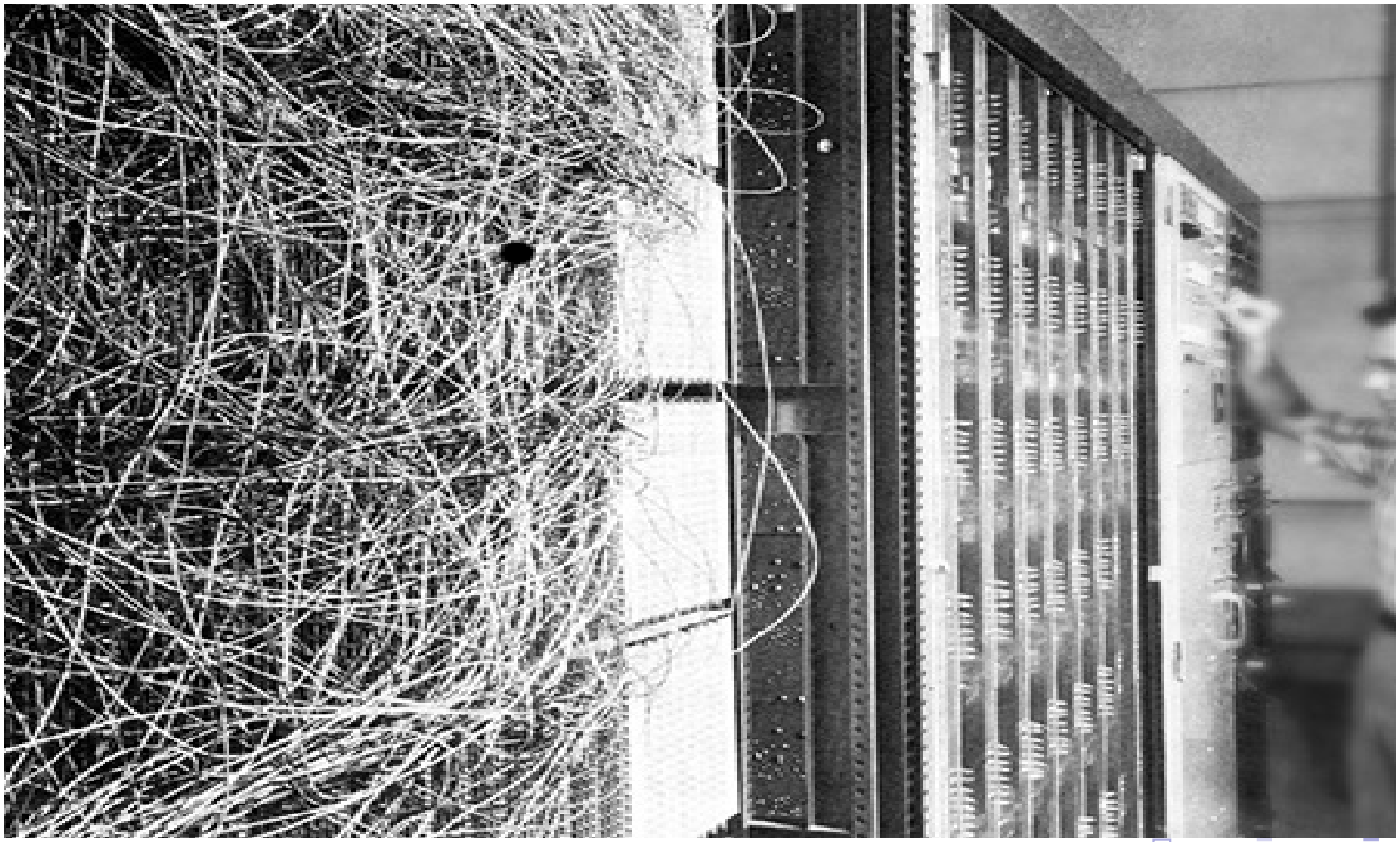
is a model to understand human memory, learning, and cognition. The perceptron was based on biological ideas about networks of neurons in the brain. In 1960 Rosenblatt demonstrated a piece of electromechanical hardware, named **Mark I Perceptron**, the first machine that could learn to recognize and identify optical patterns.

- F. Rosenblatt (1958): The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, vol. 65, n:o 6, pp. 386−408.

KTH VETENSKAP OCH KONST
KTH Matematik

The Mark 1 Perceptron

This machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors.



KTH Matematik

Linear Separability

We assume that there exists a hyperplane that correctly classifies $\mathcal{S}$. More formally this can be stated as

Linear Separability of $\mathcal{S}$

There exist $(\mathbf{w}_*, b_*)$ such that if

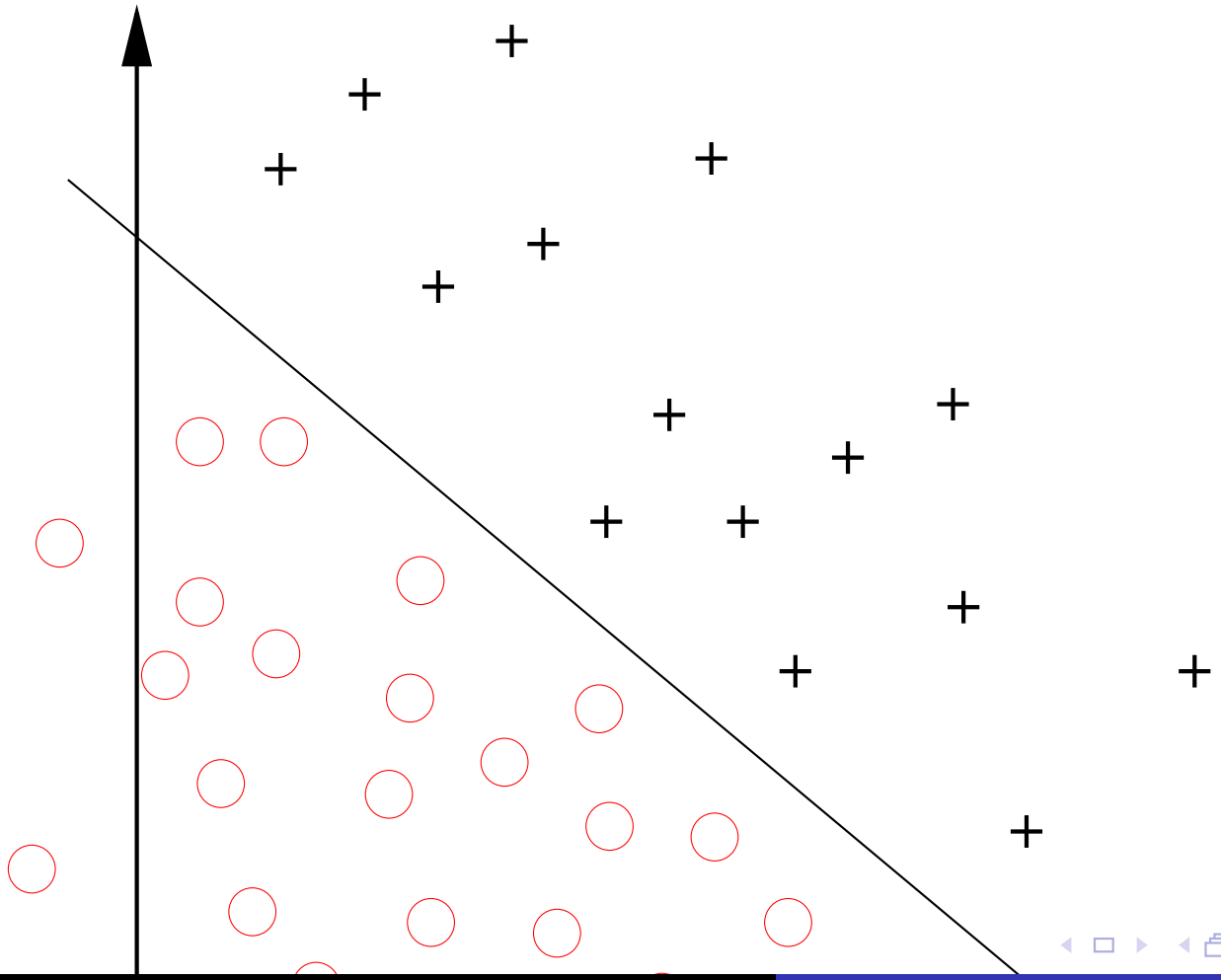
$$f_*(\mathbf{x}) = \ll \mathbf{w}_*, \mathbf{x} \gg + b_*$$

then for all $(\mathbf{x}_i, y_i) \in \mathcal{S}$ we have $f_*(\mathbf{x}) > 0$ if and only if $y_i = +1$. When this assumption is true, we call $\mathcal{S}$ *linearly separable*.

KTH VETENSKAP OCH KONST
KTH Matematik

Linear Separability

Linear separability of $\mathcal{S}$ is easy to illustrate with hyperplane geometry: The examples $(\mathbf{x}_i, y_i)$ are represented by points with two labels $\circ$ and $+$. All points with labels $+$ lie in $\mathcal{R}_1$ and all points with labels $\circ$ lie in $\mathcal{R}_2$ with regard to the hyperplane $f_*(\mathbf{x}) = 0$.

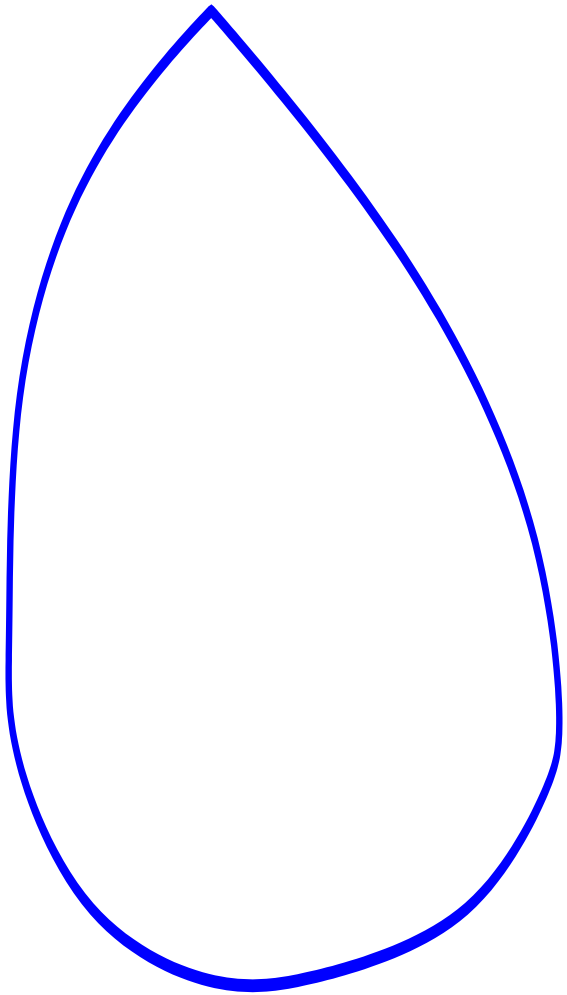


Linear Separability

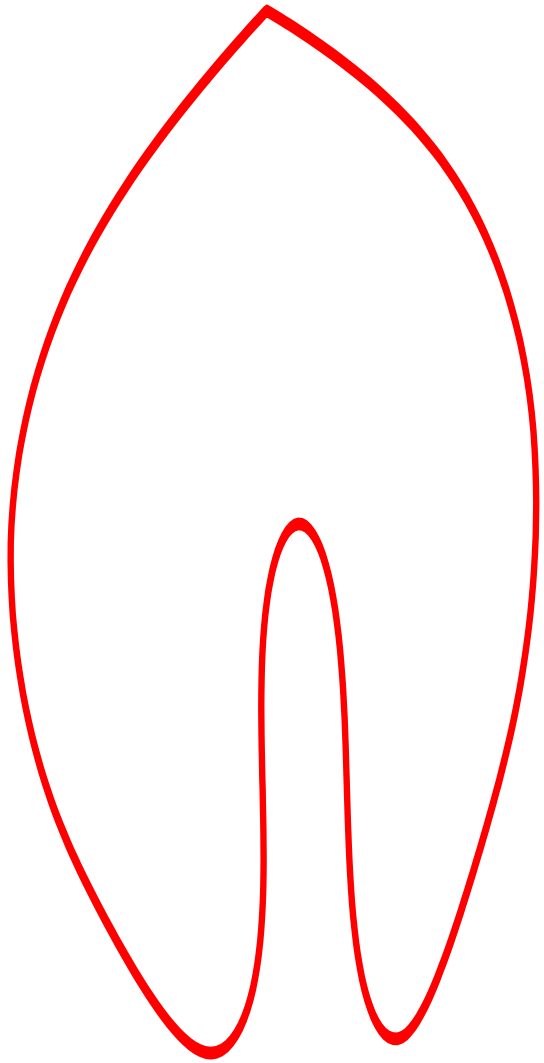
There are theorems giving sufficient conditions for separability.
E.g., two disjoint convex sets in R^k are linearly separable.
Examples of convex sets are halfplanes, and open balls.

KTH VETENSKAP OCH KONST
KTH Matematik

CONVEXITY



CONVEX



NONCONVEX

Let A and B be two disjoint nonempty convex subsets of R^n. Then there exist a nonzero vector $\mathbf{v}$ and a real number c such that

$$\langle \mathbf{x}, \mathbf{v} \rangle \geq c \text{ and } \langle \mathbf{y}, \mathbf{v} \rangle \leq c \langle \mathbf{x}, \mathbf{v} \rangle \geq c \text{ and } \langle \mathbf{y}, \mathbf{v} \rangle \leq c$$

for all x in A and y in B; i.e., the hyperplane

$$\langle \cdot, \mathbf{v} \rangle = c \langle \cdot, \mathbf{v} \rangle = c,$$

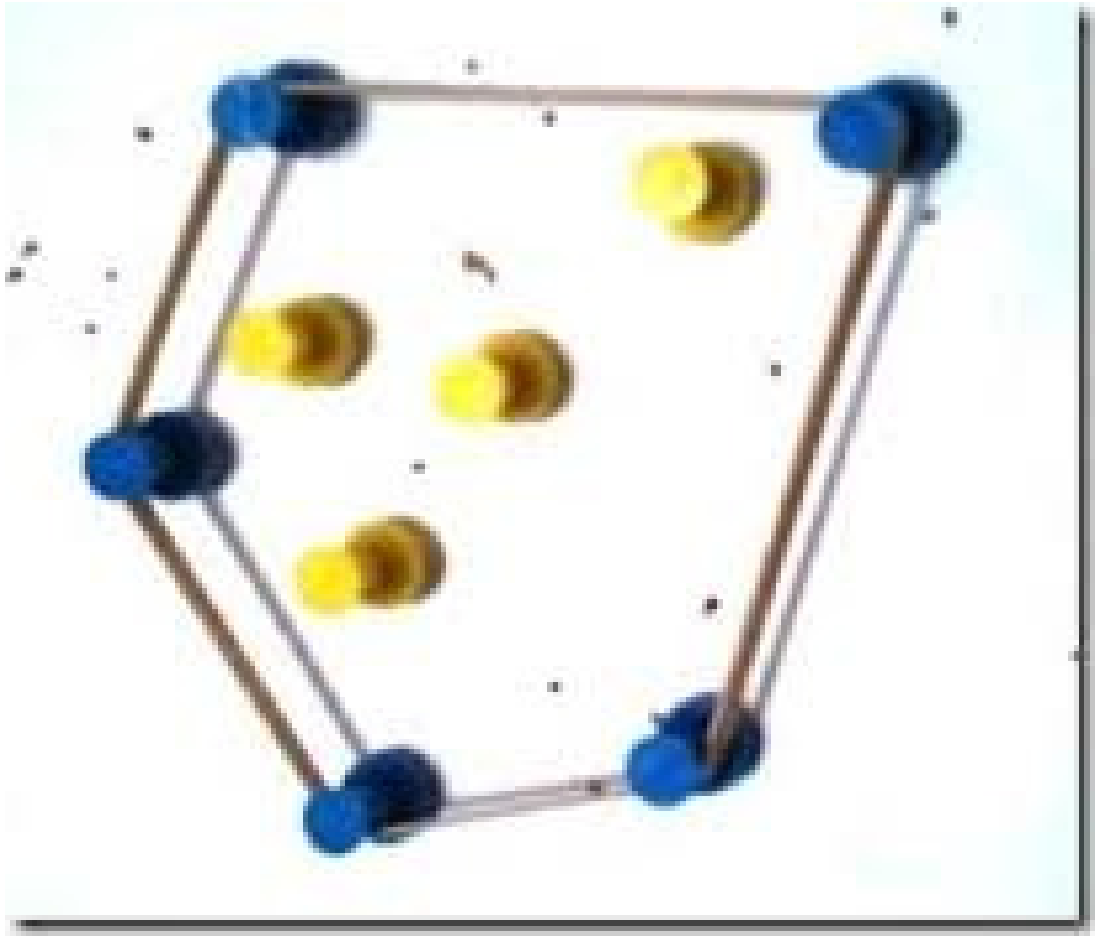
$\mathbf{v}$ the normal vector, separates A and B.

KTH VETENSKAP OCH KONST

KTH Matematik

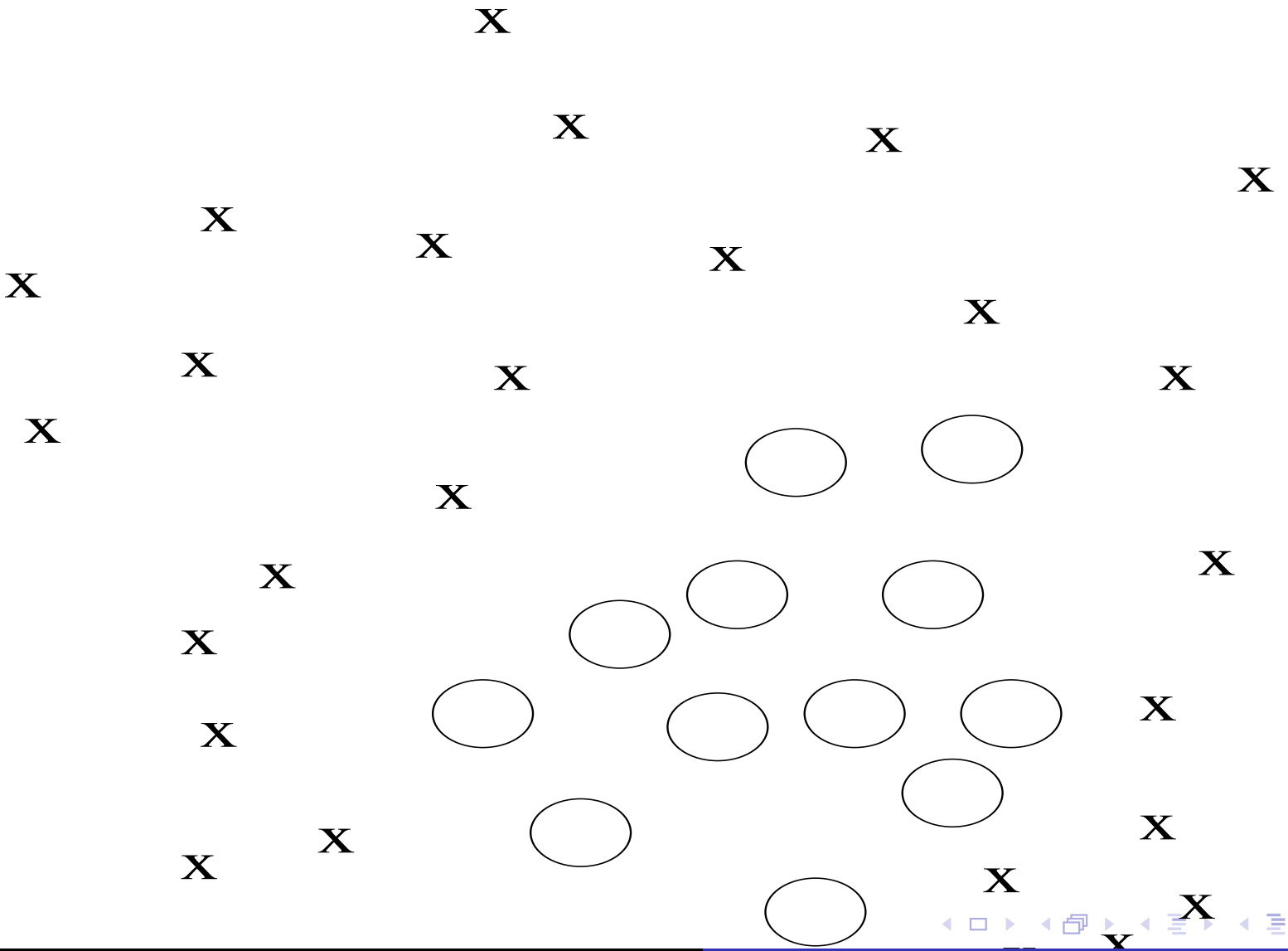
Convex Hull

The convex hull for a finite number of points: span a rubber band around it.



A LINEARLY NONSEPARABLE SITUATION

The examples correspond to points with the labels x and ○. The training set cannot be separated by a hyperplane



Rosenblatt's perceptron algorithm

An algorithm that finds a perceptron that classifies correctly a linearly separable training set can be constructed as follows. One assumes a weight vector and a bias, and then sweeps through the training set with the corresponding perceptron. For every incorrectly classified example the hyperplane direction is changed and the hyperplane is shifted parallell to itself.
It turns out that this algorithm halts after a finite number of runs, i.e., it finds a correctly classifying perceptron in a finite time, and that there is an explicit upper bound on the number of sweeps through $\mathcal{S}$.

Rosenblatt's perceptron algorithm

Every time an example is classified incorrectly, the weight vector is updated by adding a term that is proportional to the error margin (with sign). We note that

- If

$$y_i \left(\ll \mathbf{w}_k, \mathbf{x}_i \gg + b_k \right) > 0$$

  then $\mathbf{x}_i$ is correctly classified.

Hence it would seem to be a good idea to update, i.e. rotate the hyperplane, by adding to the weight vector a vector proportional to $y_i \left(\ll \mathbf{w}_k, \mathbf{x}_i \gg + b_k \right)$, as soon as this is negative.

Rosenblatt's perceptron algorithm

The bias is updated (the hyperplane is shifted parallell to itself) by adding a term that is proportional to the square of the maximum length of the input vectors in the training set. The algorithm is run until no examples in the training set are longer misclassified.

- $$R = \max_{1 \leq i \leq l} \|\mathbf{x}_i\|$$

  Clearly R gives radius for the ball $\mathcal{B}_R(\mathbf{0})$, where all training inputs lie. In addition, we must have $b < R$, since otherwise all of $\mathcal{S}$ will lie on one side of the hyperplane, which is meaningful only for *trivial training sets*, i.e. those that contain only one label.

Functional margin

The properties of the perceptron algorithm[2] will be clearly understood, once we introduce the concept of a *functional margin*. Here we set for some $(\mathbf{w}, b)$

$$\gamma_i := y_i \left(\ll \mathbf{w}, \mathbf{x}_i \gg + b_k \right) = y_i f\left(\mathbf{x}_i\right)$$

We call γ_i the functional margin of $(\mathbf{x}_i, y_i)$ with regard to $(\mathbf{w}, b)$. We set

$$\gamma := \min\left(\gamma_1, \ldots, \gamma_d\right)$$

and call γ the *functional margin* of $\mathcal{S}$ with regard to $(\mathbf{w}, b)$.

[2] as well as of support vector machines

KTH Matematik

Functional margin & the geometric margin

Recall from the above, $r_x = \frac{f(\mathbf{x})}{\|\mathbf{w}\|}$, then

$$\gamma_i = y_i f(\mathbf{x}_i) = y_i \cdot r_{x_i} \|\mathbf{w}\|$$

We obtain in view of the preceding geometric analysis, that the distance from $\mathbf{x}_i$ to D is

$$\|\mathbf{x}_i - P_D(\mathbf{x}_i)\| = |r_{x_i}|$$

If the input $\mathbf{x}_i$ is correctly classified, then

$$|r_{x_i}| = y_i \cdot r_{x_i}$$

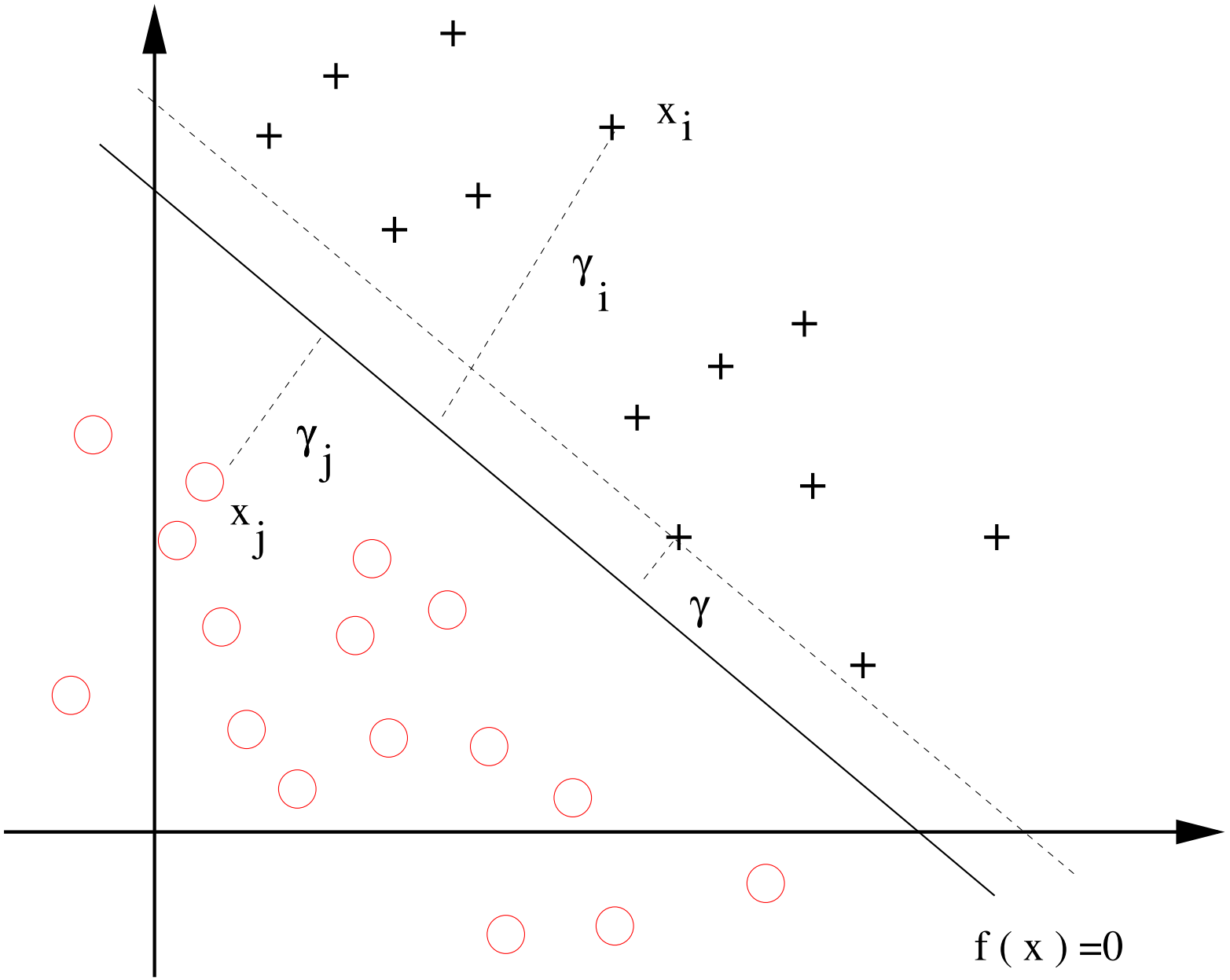
Hence, in case $\|\mathbf{w}\| = 1$, and **all $\mathbf{x}_i$ are correctly classified**,

$$\gamma = \min(\gamma_1, \ldots, \gamma_d) = \min y_i \cdot r_{x_i}$$

Hence, here γ is the *geometric margin* .
The geometric margin of $\mathcal{S}$ is the maximum of γ over all hyperplanes.

KTH VETENSKAP OCH KONST
KTH Matematik

Functional margin & the geometric margin



Rosenblatt's perceptron algorithm

1: $\mathbf{w}_0 \leftarrow \mathbf{0},\ b_0 \leftarrow 0,\ k \leftarrow 0\ R \leftarrow \max_{1 \le i \le I} \|\mathbf{x}_i\|$
2: **repeat**
3: **for** $i = 1$ to I **do**
4: **if** $y_i \left(\ll \mathbf{w}, \mathbf{x}_i \gg + b_k \right) \le 0$ **then**
5: $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + y_i \mathbf{x}_i,$
$b_{k+1} \leftarrow b_k + y_i R^2,$
$k \leftarrow k + 1$
6: **end if**
7: **end for**
8: **until** no mistakes made in the loop
9: **return** $\mathbf{w}_k, b_k$, where k is the number of mistakes.

Perceptron Convergence Theorem (Novikoff 1962)

Let $\mathcal{S}$ be a non-trivial linearly separable training set and let for all i

$$\gamma_i = y_i \left(\ll \mathbf{w}_*, \mathbf{x}_i \gg + b_* \right) \geq \gamma$$

where $\|\mathbf{w}_*\| = 1$.
Then number of repeats made by Rosenblatt's perceptron algorithm is at most

$$\left(\frac{R^2}{\gamma} \right)$$

KTH Matematik

Perceptron Convergence Theorem (Novikoff 1962)

The theorem above started the Learning Theory.

KTH Matematik

Comments on Perceptron Convergence

The number of errors t was found to bounded by

$$t \leq \left(\frac{2R}{\gamma}\right)^2$$

- The bound does not depend on the learning rate (counterintuitive)
- A large geometric margin promises a fast convergence
- A small radius on inputs promises a fast convergence

KTH VETENSKAP OCH KONST
KTH Matematik

The Scaling Freedom

If $\alpha \neq 0$, then

$$D = \{\mathbf{x} \in X \mid f(\mathbf{x}) = 0\}$$
$$= \{\mathbf{x} \in X \mid \ll \mathbf{w}, \mathbf{x} \gg + b = 0\}$$
$$= \{\mathbf{x} \in X \mid \ll \alpha\mathbf{w}, \mathbf{x} \gg + \alpha b = 0\}$$

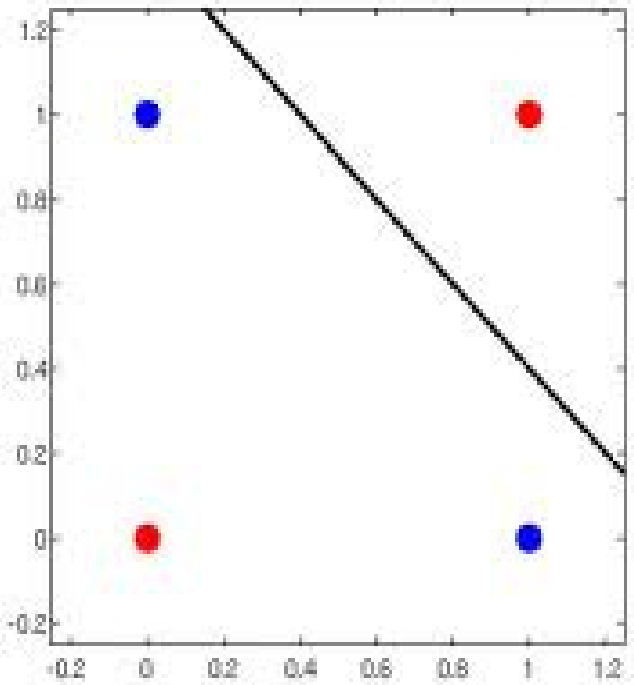
Hence $\alpha\mathbf{w}, \alpha b$ corresponds to the same hyperplane as $\mathbf{w}, b$. This is used in the proof of convergence of perceptron algorithm found below by taking $\alpha = \frac{1}{\|\mathbf{w}_*\|}$, since then

$$\|\alpha\mathbf{w}\| = \|\frac{1}{\|\mathbf{w}_*\|}\mathbf{w}\| = 1.$$

KTH VETENSKAP OCH KONST
KTH Matematik

A simple but important case where perceptron cannot work: xor

$$f(0,0) = f(1,1) = 0, f(0,1) = f(1,0) = 1$$



ARTIFICIAL NEURAL NETWORKS

The perceptron theory, as it stood in the 1960's, is very lucidly presented in

- N.J. Nilsson: *The Mathematical Foundations of Learning Machines*, originally published in 1965, reprinted in 1990, Morgan Kaufman Publishers, San Mateo, California.

A thorough presentation and at the same time a devastating criticism of the performance of perceptrons, and then a reverse reappraisal, is in

- M.L. Minsky & S.A. Papert: *Perceptrons (Expanded Edition)*, first in 1969, third (extended) printing in 1988, MIT Press, Cambridge, Massachusetts.

When $\operatorname{sign}(f(\mathbf{x}))$ was replaced by smooth functions, e.g. $\tanh(f(\mathbf{x}))$, and several such perceptrons were layered after each other, and backpropagation algorithm was invented, the theory of artificial neural networks or feedforward multilayer perceptrons, was launched.

KTH Matematik

- M.L. Minsky & S.A. Papert: *Perceptrons, Reissue Of The 1988 Expanded Edition With A New Foreword By Léon Bottou. An Introduction to Computational Geometry*. 2017 MIT Press, Cambridge, Massachusetts.

From the Publisher:
In 1969, ten years after the discovery of the perceptronwhich showed that a machine could be taught to perform certain tasks using examplesMarvin Minsky and Seymour Papert published Perceptrons, their analysis of the computational capabilities of perceptrons for specific tasks. As Lon Bottou writes in his foreword to this edition, *Their rigorous work and brilliant technique does not make the perceptron look very good.* Perhaps as a result, research turned away from the perceptron. Then the pendulum swung back, and machine learning became the fastest-growing field in computer science. Minsky and Papert's insistence on its theoretical foundations is newly relevant.

We replace $\operatorname{sign}(f(\mathbf{x}))$ by a smooth function

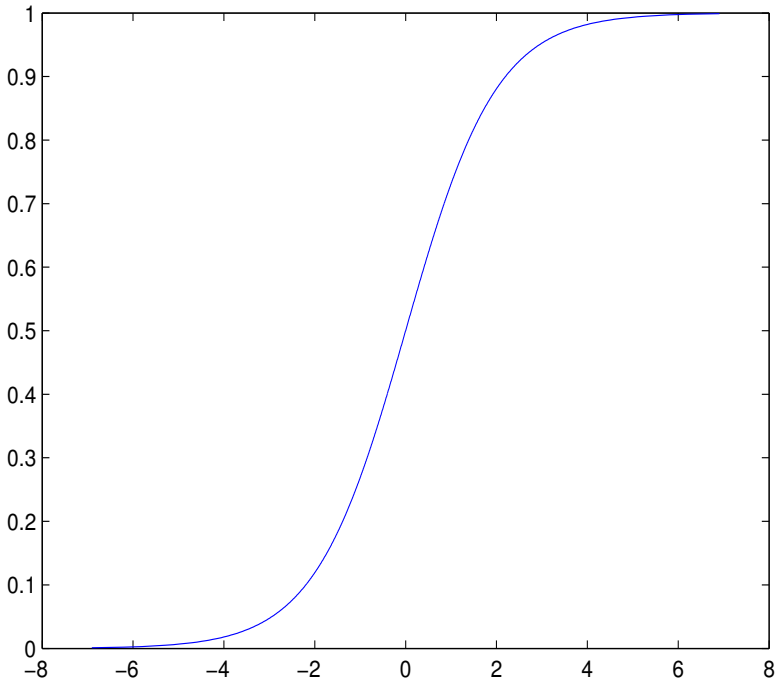
$$\operatorname{sign}(f(\mathbf{x})) \mapsto F(\ll \mathbf{w}, \mathbf{x} \gg + b) = \frac{1}{1 + e^{-(\ll \mathbf{w}, \mathbf{x} \gg + b)}}$$

or

$$F(\ll \mathbf{w}, \mathbf{x} \gg + a) = \tanh(\ll \mathbf{w}, \mathbf{x} \gg + b)$$

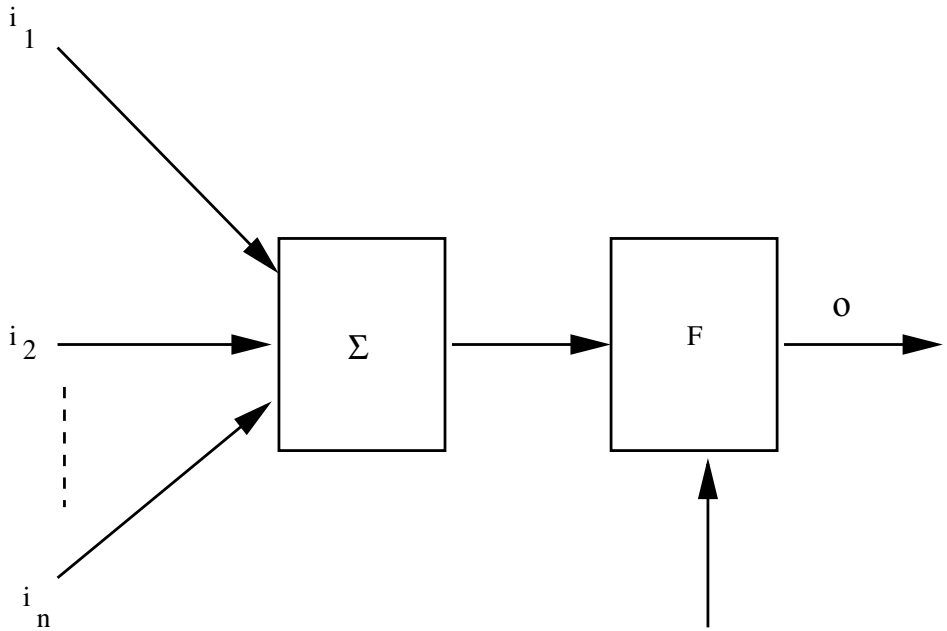
and several such smoothed perceptrons were layered after each other, and **backpropagation algorithm** was invented, the theory of artificial neural networks or feedforward multilayer perceptrons, was launched.

$F(x) = \frac{1}{1+e^{-x}}$, the logistic function



KTH VETENSKAP OCH KONST
KTH Matematik

A Neuron



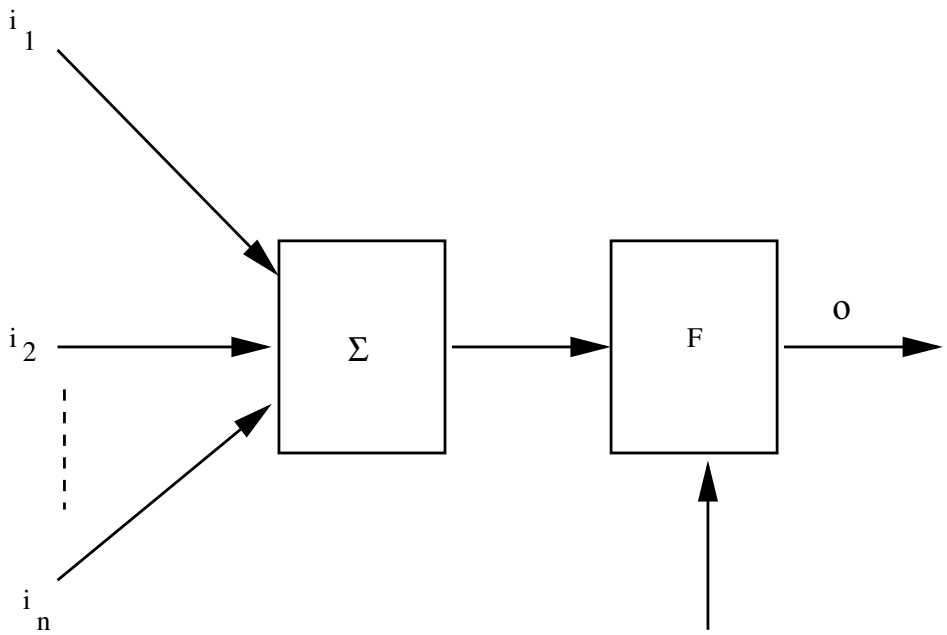
$$\mathbf{o} = F\left(\ll \mathbf{w}, \mathbf{i} \gg + b\right)$$

$$\ll \mathbf{w}, \mathbf{i} \gg = \sum_{i=1}^{n} \mathbf{w}_i \mathbf{i}_i$$

$$F(x) = \frac{1}{1 + e^{-x}}$$

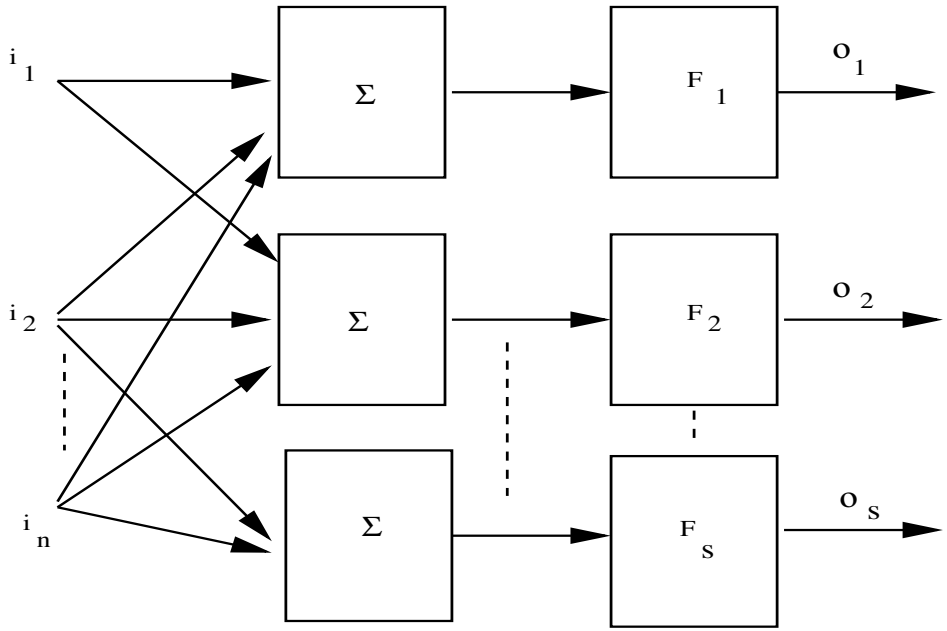
KTH Matematik

A Neuron



$$F(x) = \frac{1}{1 + e^{-x}}$$

A Neuron Layer

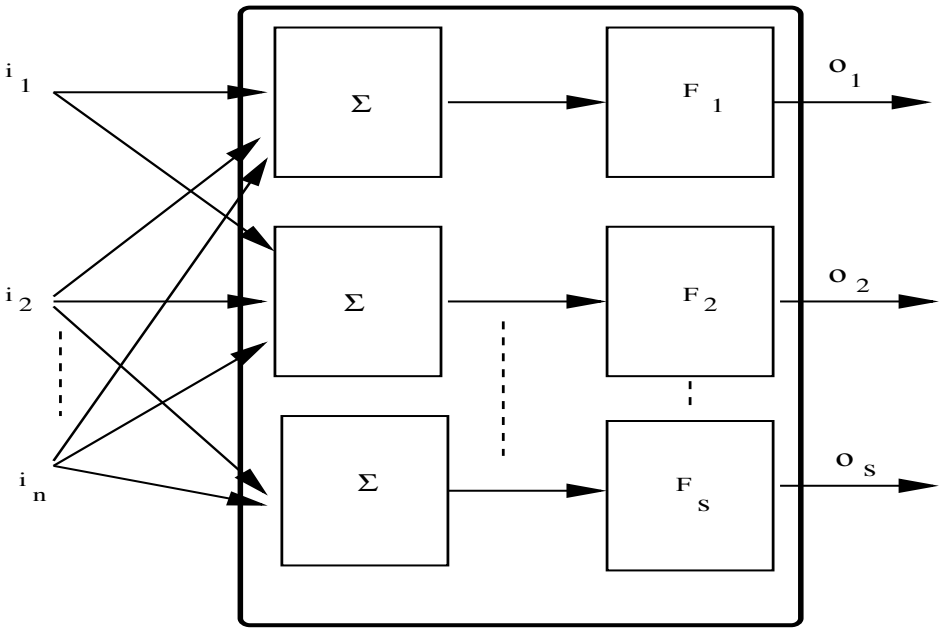


$$\mathbf{o}_j = F\left(\ll \mathbf{w}, \mathbf{i} \gg + b_j\right)$$

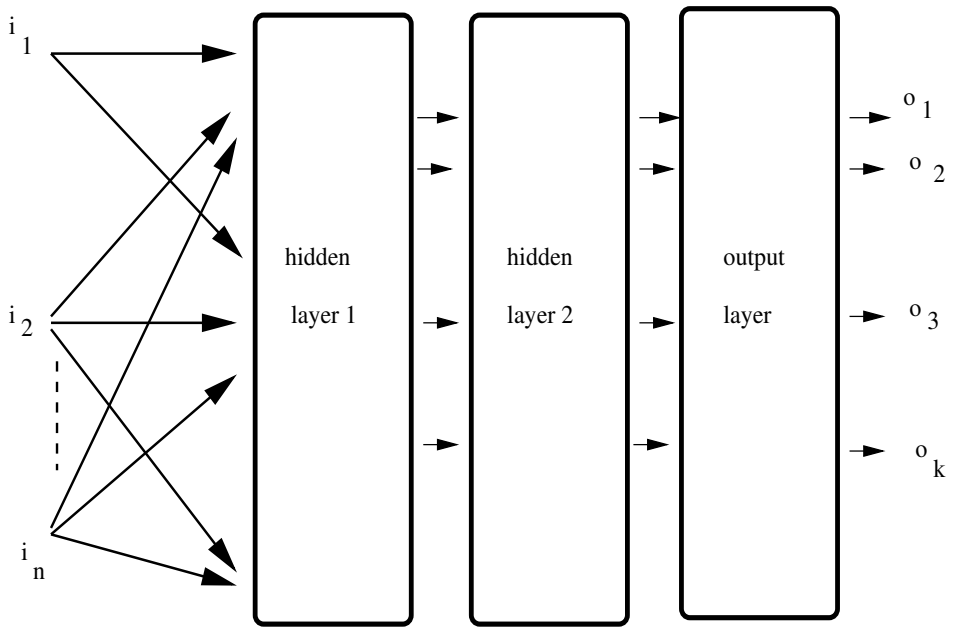
$$\ll \mathbf{w}, \mathbf{i} \gg = \sum_{i=1}^{n} \mathbf{w}_{ij}\mathbf{i}_i$$

$$F(x) = \frac{1}{1 + e^{-x}}$$

A Neuron Layer



A Neural Network Architecture



Training a Neural Network in classification

We use the t training items $\mathbf{x}^{(l)}$ as input to the net and will tune the all parameters (weights and biases) $\mathbf{w}$ and $\mathbf{b}$ so that

$$\sum_{l=1}^{t}\sum_{j=1}^{k}\left(o_j^{(l)} - u_j^{(l)}\right)^2$$

is minimized. Here $u_j^{(k)}$ labels of the items $\mathbf{x}^{(l)}$, i.e. $u_j^{(k)}$ contains $+1$ and the rest are 0s.

KTH Matematik

Training a Multilayer Neural Network

There exists a theoretically effective algorithm for tuning known as **backpropagation**, which propagates the gradients of the function

$$\sum_{l=1}^{t}\sum_{j=1}^{k}\left(o_j^{(l)} - u_j^{(k)}\right)^2$$

with respect to the the weights and biases **w** and **b** to the hidden layers.

Deep Learning Network: an Example

$$y_i^{(1)} = \max\left(\ll \mathbf{w}_i^{(1)}, \mathbf{x} \gg, 0 \right), i = 1, \ldots, 600$$

$$y_i^{(2)} = \log\left(\ll \mathbf{w}_i^{(2)}, \mathbf{y}^{(1)} \gg)^2 + 1 \right), i = 1, \ldots, 100$$

Restriction: $\mathbf{w}_i^{(2)} \geq 0$

$$\mathbf{z}^{(2)} = \text{gain control}\left(\mathbf{y}^{(2)} \right)$$

$$y_i^{(3)} = \max\left(\ll \mathbf{w}_i^{(3)}, \mathbf{z}^{(2)} \gg, 0 \right), i = 1, \ldots, 50$$

In the visual system, cells in V1 are thought to mutually inhibit, causing normalization of responses to contrast, a form of automatic gain control.

KTH VETENSKAP OCH KONST
KTH Matematik

An up-to-date survey of the these topics and more

Huang, Guang-Bin: What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation* 7, pp. 263−278, 2015.

KTH VETENSKAP OCH KONST

KTH Matematik

Novikoff's proof of perceptron convergence

Perceptron Convergence Theorem (Novikoff 1962). A simpler proof in section 6.5.3 of A.Blum, J. Hopcroft and R. Kannan: Foundations of Data Science, 2016

Let $\mathcal{S}$ be a non-trivial linearly separable training set and let for all i

$$\gamma_i = y_i \left(\ll \mathbf{w}_*, \mathbf{x}_i \gg + b_* \right) \geq \gamma$$

where $\|\mathbf{w}_*\| = 1$.
Then number of repeats made by Rosenblatt's perceptron algorithm is at most

$$\left(\frac{R^2}{\gamma} \right)$$

KTH VETENSKAP OCH KONST
KTH Matematik

Proof: Perceptron Convergence (1)

We extend the inner product space to $X \times R$; for $\widehat{\mathbf{x}} = (\mathbf{x}, x)$ and $\widehat{\mathbf{y}} = (\mathbf{y}, y)$. We take $\widehat{\mathbf{x}} + \widehat{\mathbf{y}} = (\mathbf{x} + \mathbf{y}, x + y)$, $\alpha\widehat{\mathbf{x}} = (\alpha\mathbf{x}, \alpha x)$ and we set

$$\ll \widehat{\mathbf{x}}, \widehat{\mathbf{y}} \gg_{ext} = \ll \mathbf{x}, \mathbf{y} \gg + xy$$

This is an inner product on $X \times R$.

We set

$$\widehat{\mathbf{x}}_i = (\mathbf{x}_i, R)$$

$$\widehat{\mathbf{w}} = \left(\mathbf{w}, \frac{b}{R}\right)$$

By our extended inner product above we get

$$y_i \ll \widehat{\mathbf{w}}, \widehat{\mathbf{x}}_i \gg_{ext} = y_i \left(\ll \mathbf{w}, \mathbf{x}_i \gg + b\right)$$

Let now $\widehat{\mathbf{w}}_{t-1}$ be the weight vector prior to the tth mistake. The tth mistake/update happens when $(\mathbf{x}_i, y_i)$ is classified incorrectly by $\widehat{\mathbf{w}}_{t-1}$, i.e.,

$$y_i \ll \widehat{\mathbf{w}}_{t-1}, \widehat{\mathbf{x}}_i \gg_{ext} \leq 0$$

Proof: Perceptron Convergence (2)

In view of the pseudocode above the updates can be written as

$$\left(\mathbf{w}_t, \frac{b_t}{R}\right) \leftarrow \left(\mathbf{w}_{t-1} + \eta y_i \mathbf{x}_i, \frac{b_{t-1}}{R} + \eta y_i R\right)$$

$$\Leftrightarrow$$

$$\widehat{\mathbf{w}}_t = \widehat{\mathbf{w}}_{t-1} + \eta y_i \widehat{\mathbf{x}}_i$$

Let

$$\widehat{\mathbf{w}}_* = \left(\mathbf{w}_*, \frac{b_*}{R}\right)$$

correspond to a separating hyperplane with $\mathbf{w}_*$ and b_*. Then we get

$$\ll \widehat{\mathbf{w}}_t, \widehat{\mathbf{w}}_* \gg_{ext} = \ll \widehat{\mathbf{w}}_{t-1}, \mathbf{w}_* \gg_{ext} + \eta y_i \ll \widehat{\mathbf{x}}_i, \widehat{\mathbf{w}}_* \gg_{ext}$$

and

$$\ll \widehat{\mathbf{x}}_i, \widehat{\mathbf{w}}_* \gg_{ext} = \ll \mathbf{x}_i, \mathbf{w}_* \gg + R\frac{b_*}{R}$$

$$= \ll \mathbf{x}_i, \mathbf{w}_* \gg + b_*$$

Proof: Perceptron Convergence (3)

Hence linear separability of $\mathcal{S}$ w.r.t. $\mathbf{w}_*, b_*$ entails

$$\eta y_i \ll \widehat{\mathbf{x}}_i, \widehat{\mathbf{w}}_* \gg = \eta y_i \left(\ll \mathbf{x}_i, \mathbf{w}_* \gg + b_* \right) \geq \eta\gamma.$$

In other words we have obtained that

$$\ll \widehat{\mathbf{w}}_t, \widehat{\mathbf{w}}_* \gg_{ext} \geq \ll \underline{\mathbf{w}}_{t-1}, \underline{\mathbf{w}}_* \gg + \eta\gamma$$

By successive application of this inequality we get

$$\ll \widehat{\mathbf{w}}_t, \widehat{\mathbf{w}}_* \gg_{ext} \geq t\eta\gamma$$

KTH VETENSKAP OCH KONST
KTH Matematik

Proof: Perceptron Convergence (4)

We use

$$\widehat{\mathbf{w}}_t = \widehat{\mathbf{w}}_{t-1} + \eta y_i \widehat{\mathbf{x}}_i$$

and the rules for norms and inner products and get

$$\|\widehat{\mathbf{w}}_t\|_{ext}^2 = \|\widehat{\mathbf{w}}_{t-1}\|_{ext}^2 + \eta^2 \|\widehat{\mathbf{x}}_{t-1}\|_{ext}^2 + 2\eta \underbrace{y_i \ll \widehat{\mathbf{w}}_{t-1}, \widehat{\mathbf{w}}_* \gg_{ext}}_{<0}$$

$$\leq \|\widehat{\mathbf{w}}_{t-1}\|_{ext}^2 + \eta^2 \|\widehat{\mathbf{x}}_{t-1}\|_{ext}^2$$

by incorrect classification of $(\mathbf{x}_i, y_i)$, and

$$\leq \|\widehat{\mathbf{w}}_{t-1}\|_{ext}^2 + \eta^2 \left(\|\mathbf{x}_{t-1}\|^2 + R^2 \right)$$

$$\leq \|\widehat{\mathbf{w}}_{t-1}\|_{ext}^2 + \eta^2 \left(R^2 + R^2 \right)$$

$$\leq \|\widehat{\mathbf{w}}_{t-1}\|_{ext}^2 + 2\eta^2 R^2$$

I.e., by successive application

$$\|\widehat{\mathbf{w}}_t\|_{ext}^2 \leq 2t\eta^2 R^2$$

Proof: Perceptron Convergence (5)

By the results

$$\ll \widehat{\mathbf{w}}_t, \widehat{\mathbf{w}}_* \gg_{ext} \geq t\eta\gamma$$

and

$$\|\widehat{\mathbf{w}}_t\|_{ext}^2 \leq 2t\eta^2 R^2$$

we get by Cauchy-Schwarz' inequality

$$t\eta\gamma \leq \ll \widehat{\mathbf{w}}_t, \widehat{\mathbf{w}}_* \gg_{ext} \leq \|\widehat{\mathbf{w}}_t\|_{ext} \cdot \|\widehat{\mathbf{w}}_*\|_{ext} \leq \sqrt{2t}\eta R \|\widehat{\mathbf{w}}_*\|_{ext}$$

We need to bound $\|\widehat{\mathbf{w}}_*\|_{ext}$.

Proof: Perceptron Convergence (6)

We need to bound $\|\widehat{\mathbf{w}}_*\|_{ext}$ and therefore we check

$$\|\widehat{\mathbf{w}}_*\|_{ext}^2 = \|\mathbf{w}_*\|^2 + \frac{b_*^2}{R^2}$$

and since $\mathcal{S}$ is non-trivial,

$$\leq \underbrace{\|\mathbf{w}_*\|^2}_{=1} + \frac{R^2}{R^2} = 2$$

In all we have obtained

$$t\eta\gamma \leq \sqrt{2t}\eta R^2 \|\widehat{\mathbf{w}}_*\|_{ext} \leq \sqrt{t}\eta 2R$$

i.e.,

$$\sqrt{t} \leq \frac{2R}{\gamma}$$

and

$$t \leq \left(\frac{2R}{\gamma}\right)^2$$

which verifies the assertion as claimed.

Rosenblatt's perceptron algorithm. Given linearly separable $\mathcal{S}$ and a learning rate $\eta > 0$

1: $\mathbf{w}_0 \leftarrow \mathbf{0}$, $b_0 \leftarrow 0$, $k \leftarrow 0$ $R \leftarrow \max_{1 \leq i \leq l} \|\mathbf{x}_i\|$
2: **repeat**
3: **for** $i = 1$ to l **do**
4: **if** $y_i \left(\ll \mathbf{w}_k, \mathbf{x}_i \gg + b_k \right) \leq 0$ **then**
5: $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \eta y_i \mathbf{x}_i$,
$b_{k+1} \leftarrow b_k + \eta y_i R^2$,
$k \leftarrow k + 1$
6: **end if**
7: **end for**
8: **until** no mistakes made in the loop
9: **return** $\mathbf{w}_k, b_k$, where k is the number of mistakes.

KTH VETENSKAP OCH KONST
KTH Matematik