Supervised Learning from the Inside

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

October 2020

Overview

- The supervised learning problem
- Perceptron: classification with a linear model
- Neural networks: classification with a non-linear model
- Conclusion

Examples

- A bank has the following table.

Loan amount	Duration	Annual income	Age	Employment sector	Paid back?

  For an ongoing loan, the bank wants to predict whether it will be paid back.

- A trading firm has recorded daily opening and closing prices of stocks, along with other relevant information/news, from the last 5 years.

  The firm would like to get a good estimate of . . . tomorrow's opening stock price!

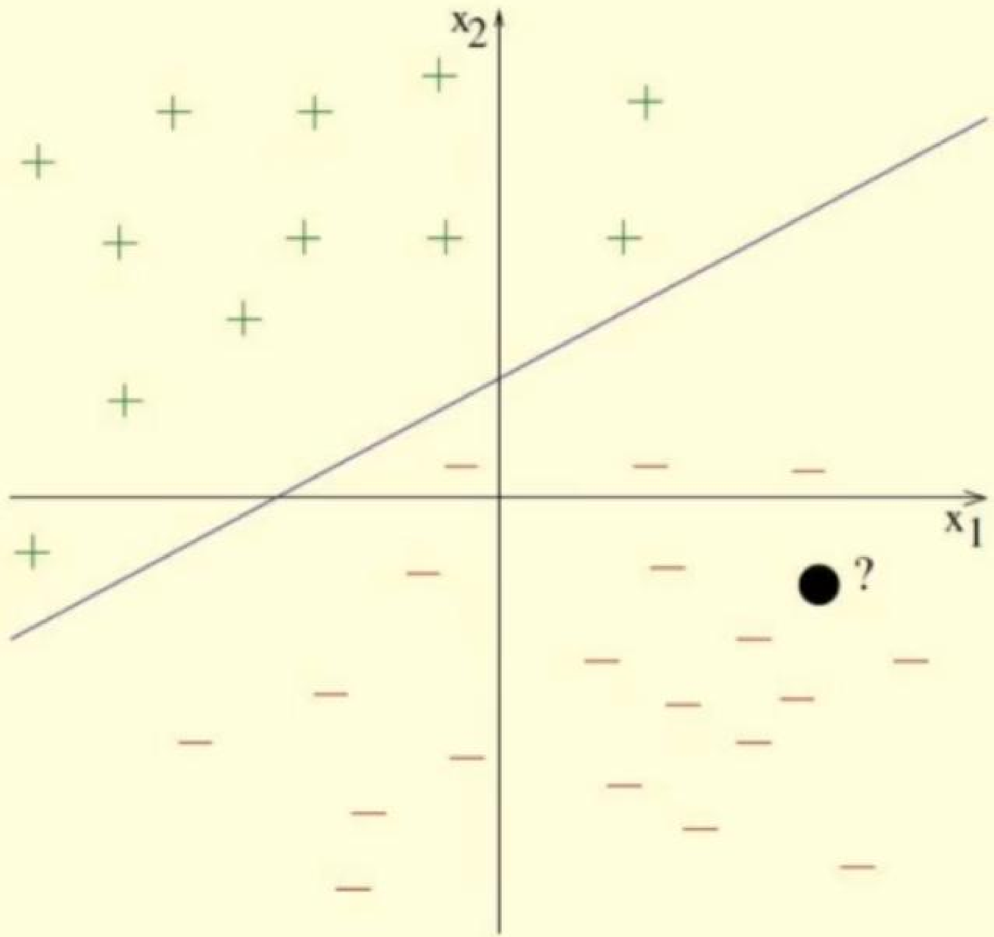
- An industrial unit periodically logs readings from sensors on its machines.

  The unit would like to anticipate imminent failure of machines to take corrective action beforehand.

  What should the bank, trading firm, and industrial unit do?

  Learn a predictive model.

Pictorial View



Supervised Learning Problem

- **Input**: Data set $D = \{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$.
  Each x^i is an array (or vector) of features $x^i_1, x^i_2, \ldots, x^i_d$.
  y^i is the label (or supervision) corresponding to x_i.
  Output: Model.

  $$D \longrightarrow \boxed{\text{Supervised Learning Algorithm}} \longrightarrow \text{Model}.$$

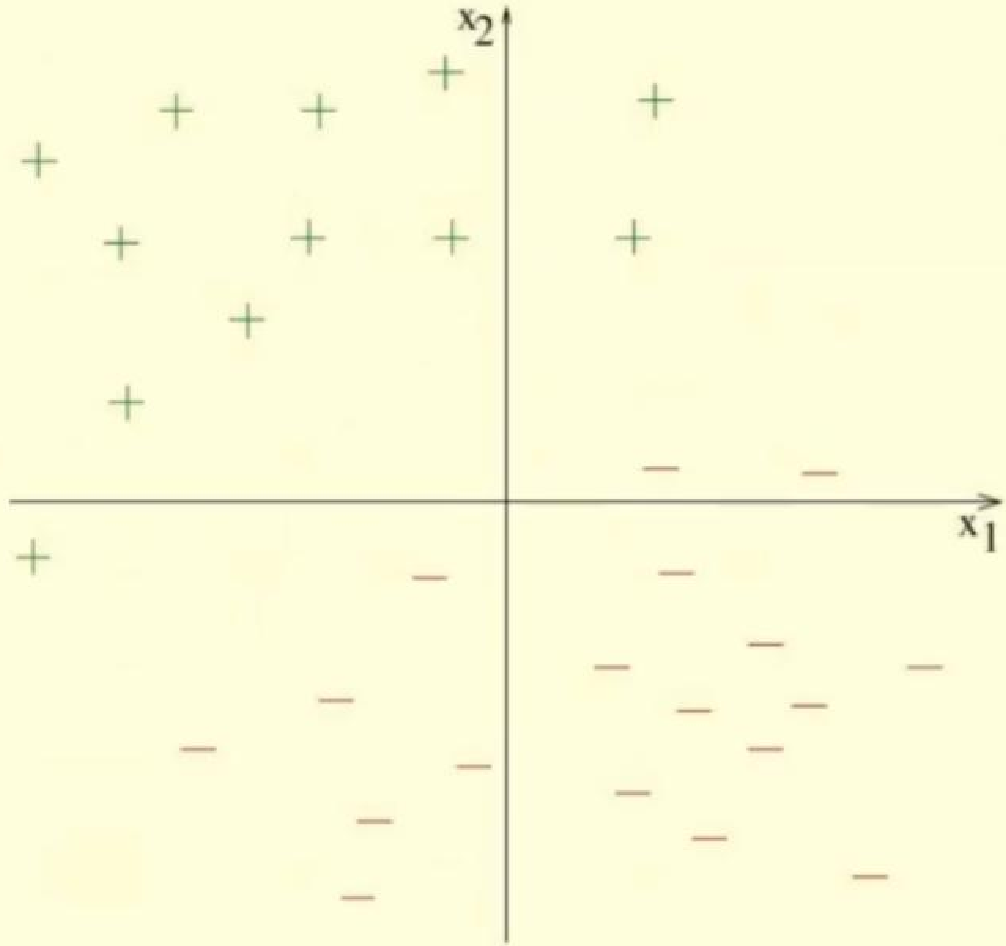
- Using the model on new x (not necessarily present in D) to predict y.

  $$x \longrightarrow \boxed{\text{Model}} \longrightarrow y.$$

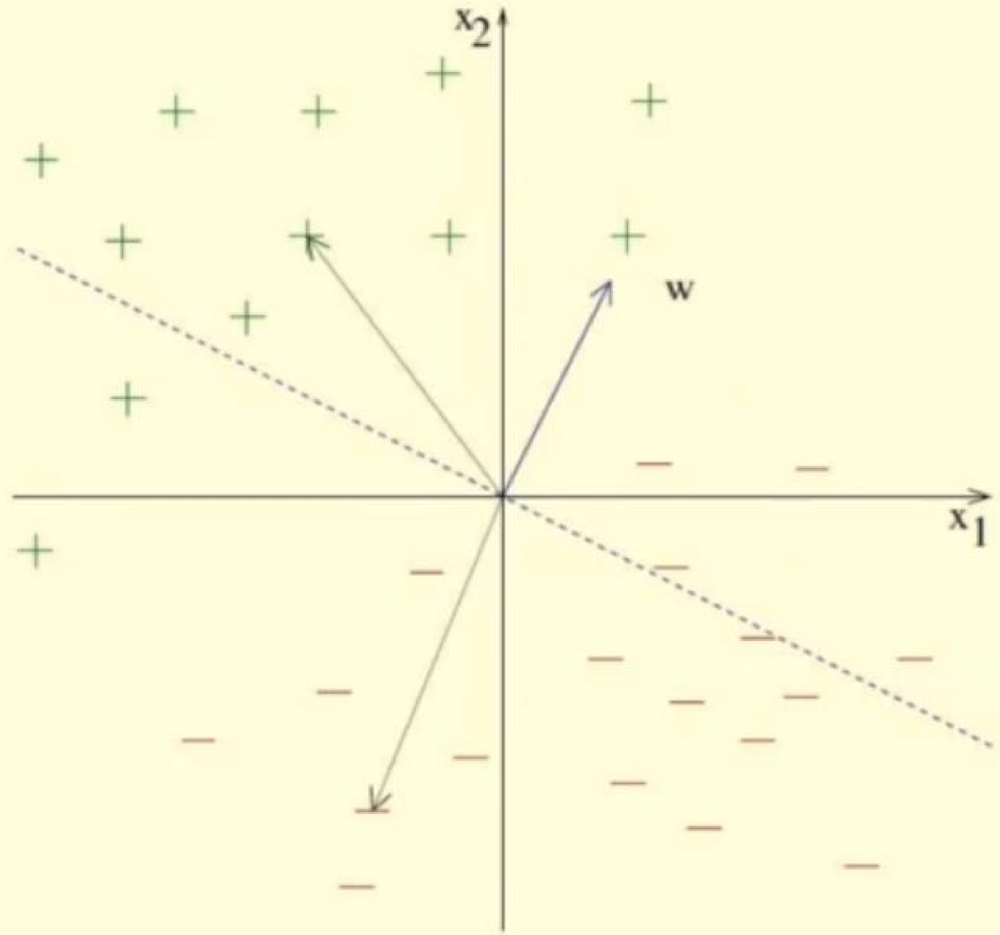
Overview

- The supervised learning problem
- Perceptron: classification with a linear model
- Neural networks: classification with a non-linear model
- Conclusion

Perceptron Learning Algorithm



First, How to *Represent* an Origin-passing Hyperplane?



By a vector $w = (w_1, w_2)$.

For point x,

- assign label "+" if $w \cdot x \geq 0$;
- assign label "−" otherwise.

Now how to find a satisfying w?

Perceptron Learning Algorithm

Initialise w arbitrarily. (Recall that it is a d-dimensional vector.)

While there is some misclassified point:

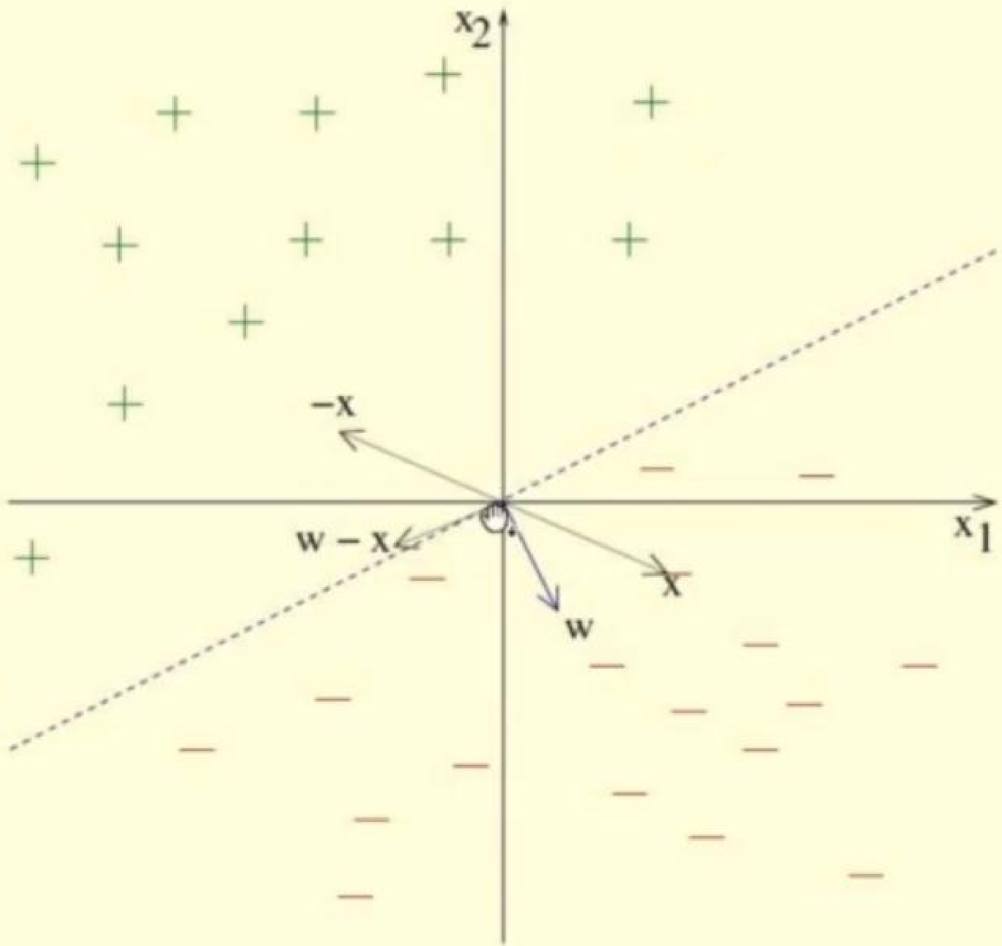
Select an arbitrary misclassified point (x, y).
//That means $y = 1$ but $w \cdot x < 0$, or $y = -1$ but $w \cdot x \geq 0$.

Set $w \leftarrow w + yx$.

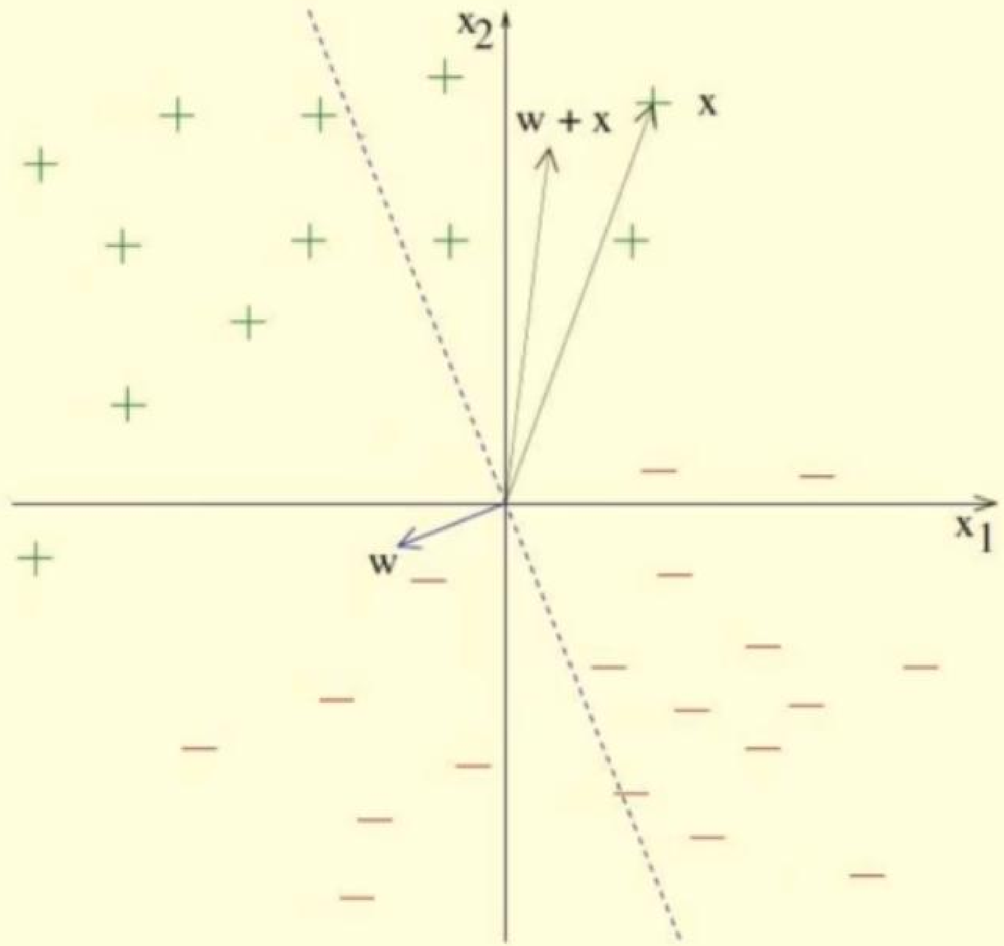
Return w.

That's it!

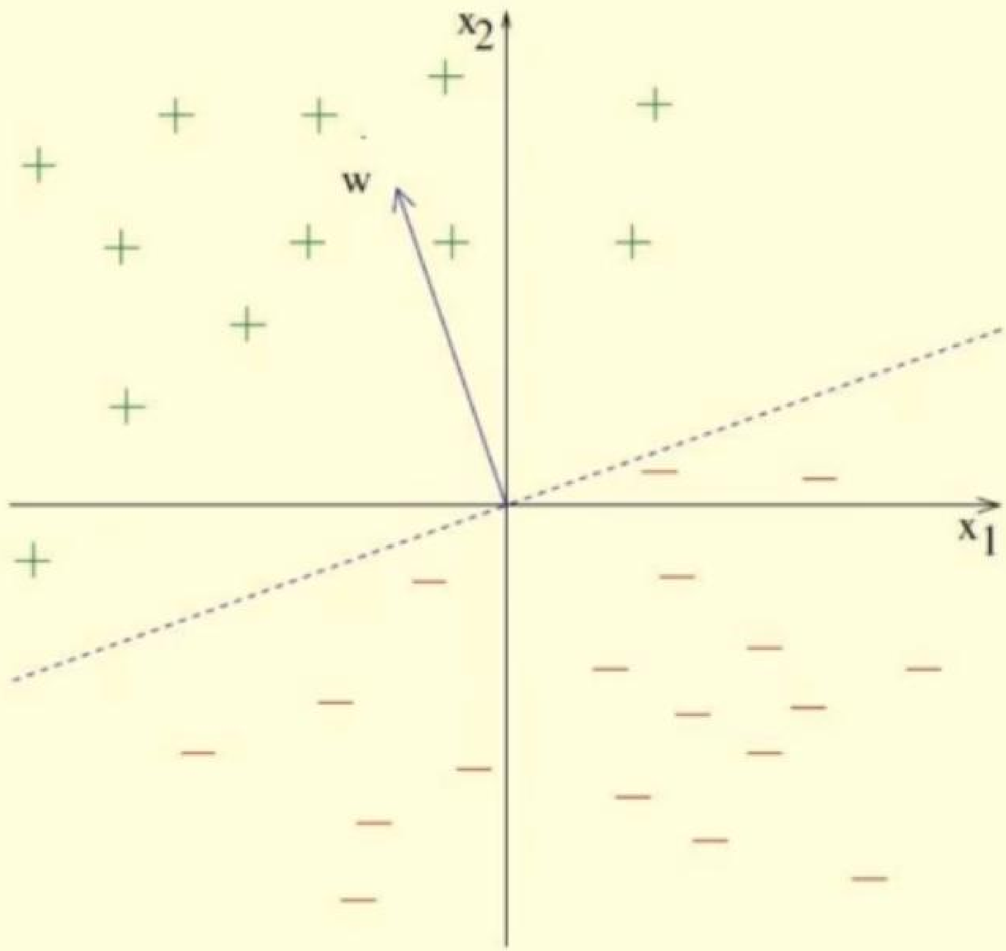
Step-through



Step-through



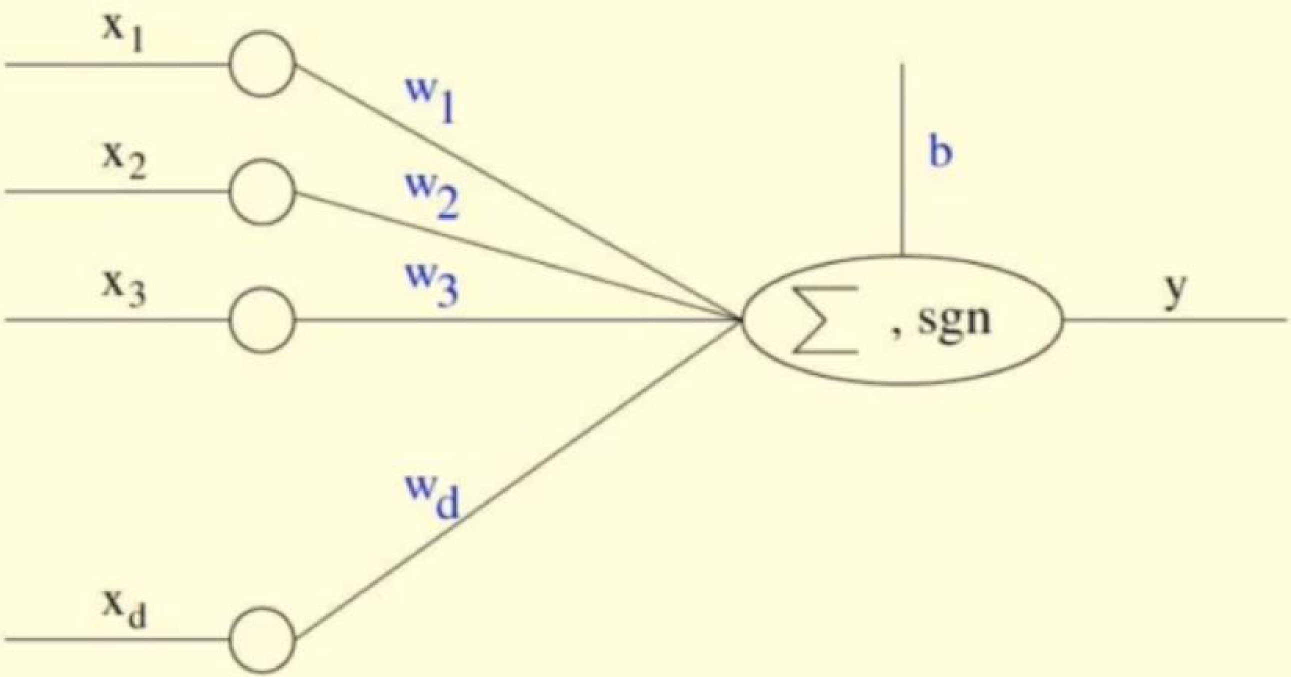
Step-through



- Will this algorithm *always* find a satisfying solution?

  Yes—if indeed there is linear separation of the classes.

Why Called "Perceptron"?



$$sgn(\alpha) \stackrel{\text{def}}{=} \begin{cases} 0 & \alpha < 0, \\ 1 & \alpha \geq 0. \end{cases}$$

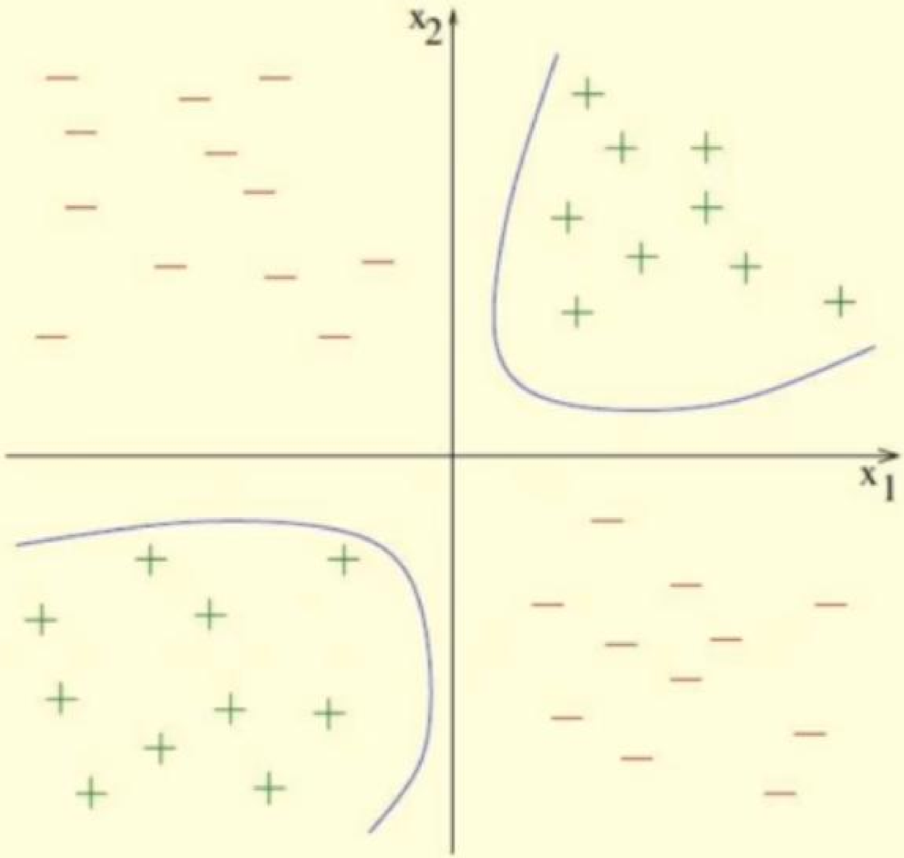
$$y = \text{sgn}(w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b).$$

$w = (w_1, w_2, \ldots, w_d)$ and b are parameters learned from data.

Overview

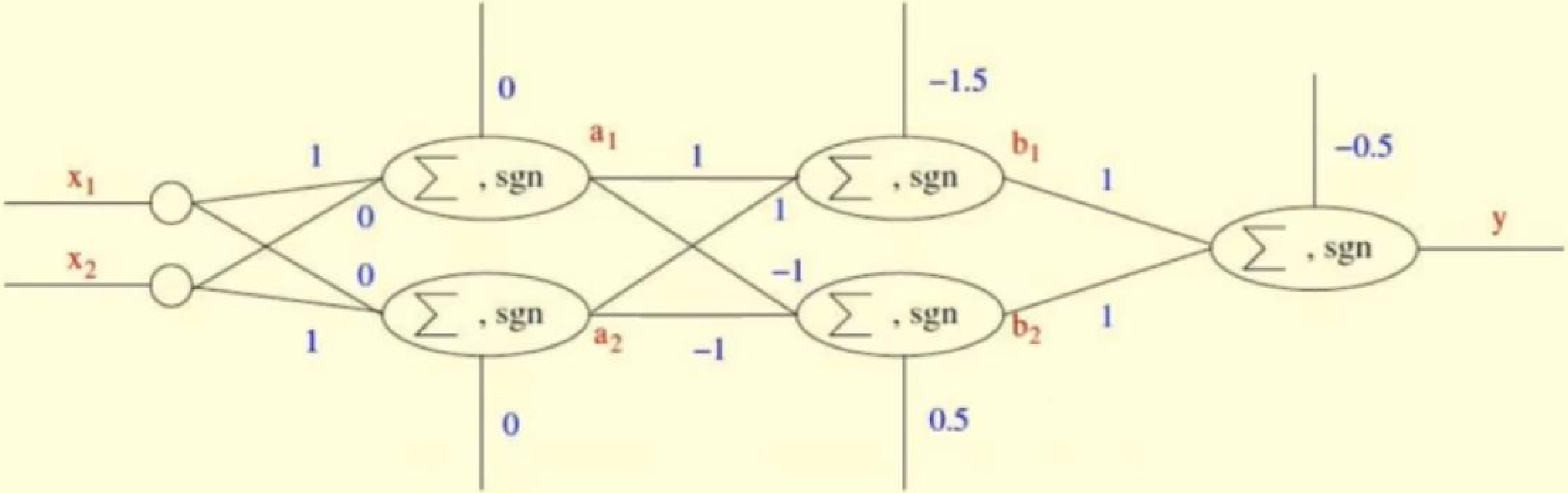
- The supervised learning problem
- Perceptron: classification with a linear model
- Neural networks: classification with a non-linear model
- Conclusion

The XOR Problem



Can we learn accurate predictors from data that is not linearly separable?

Idea: Combine Multiple Perceptrons!



$$a_1 = \text{sign}(x_1).$$
$$a_2 = \text{sign}(x_2).$$
$$b_1 = a_1 \, \text{AND} \, a_2.$$
$$b_2 = (\text{NOT} \, a_1) \text{AND} (\text{NOT} \, a_2).$$
$$y = b_1 \, \text{OR} \, b_2.$$

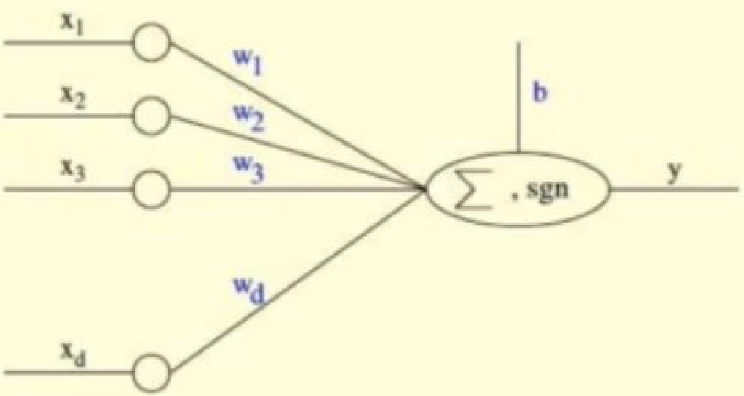
We solved the XOR problem! What's the catch?

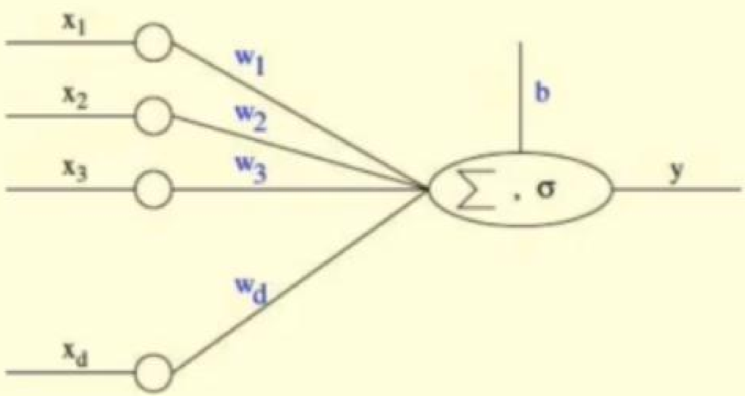
We don't know how to learn these weights for this network of perceptrons.

But we'll make an effort . . . .

Artificial Neuron

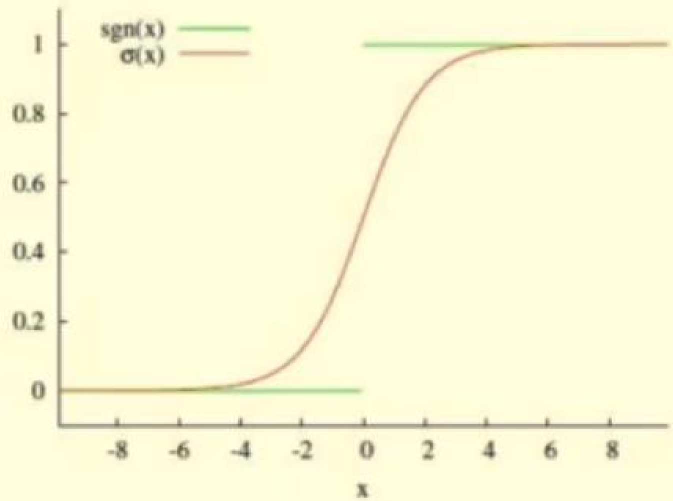
Perceptron | **Artificial neuron**





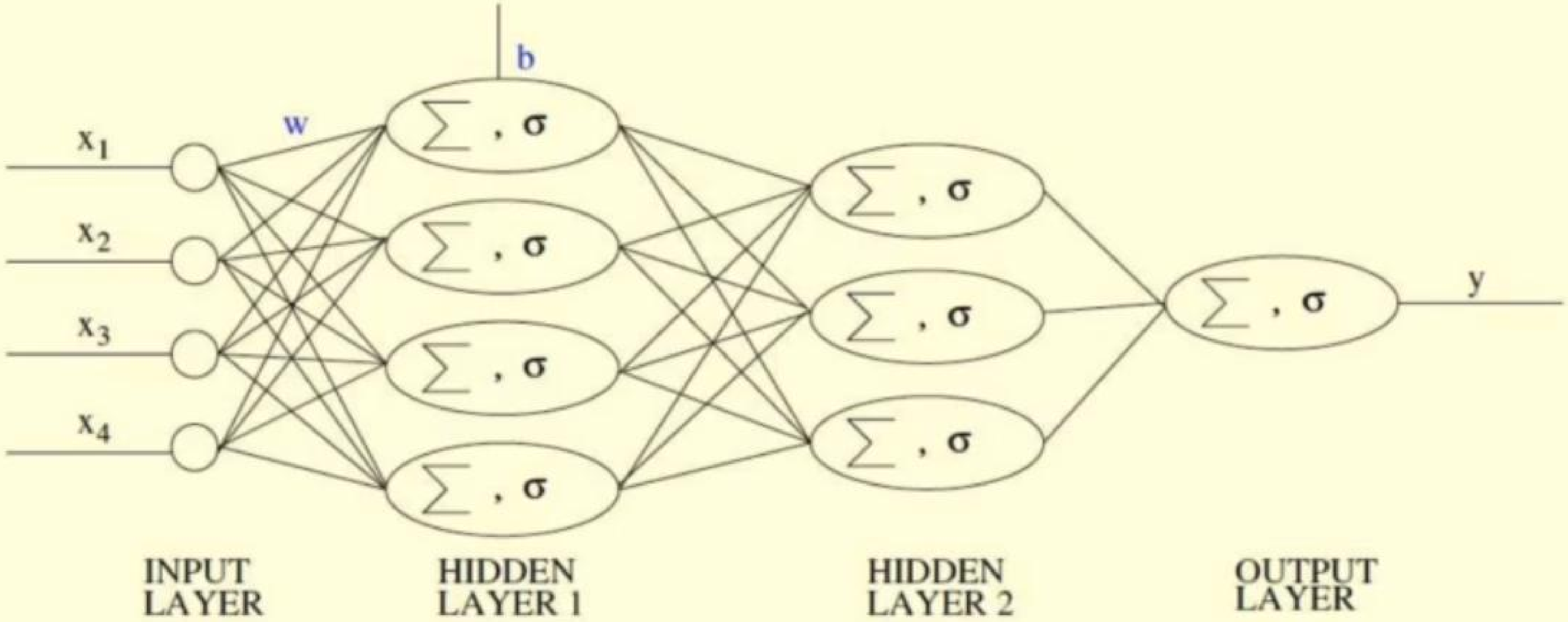
$$sgn(x) \stackrel{\text{def}}{=} \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$$

$$\sigma(x) \stackrel{\text{def}}{=} \frac{1}{1+\exp(-x)}.$$



$\sigma()$ is called an activation function. It is differentiable, while sgn() is not.

Artificial Neural Networks



Artificial neural networks are Universal Approximators.

For any function on a finite data set, there exists a single-hidden-layer neural network that fits it exactly.

Backpropagation Algorithm

We are given a training data set $\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$.
Let us start with some initial weights $\mathbf{w}$.

For each data point r,

the true label is y^r,

the prediction is $p^r(\mathbf{w})$; and

thus, the error is $(y^r - p^r(\mathbf{w}))^2$.

The aggregate error $E(\mathbf{w})$ is $\sum_{r=1}^{n}(y^r - p^r(\mathbf{w}))^2$.
We move a step in the direction minimising error (gradient descent):

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} E(\mathbf{w}),$$

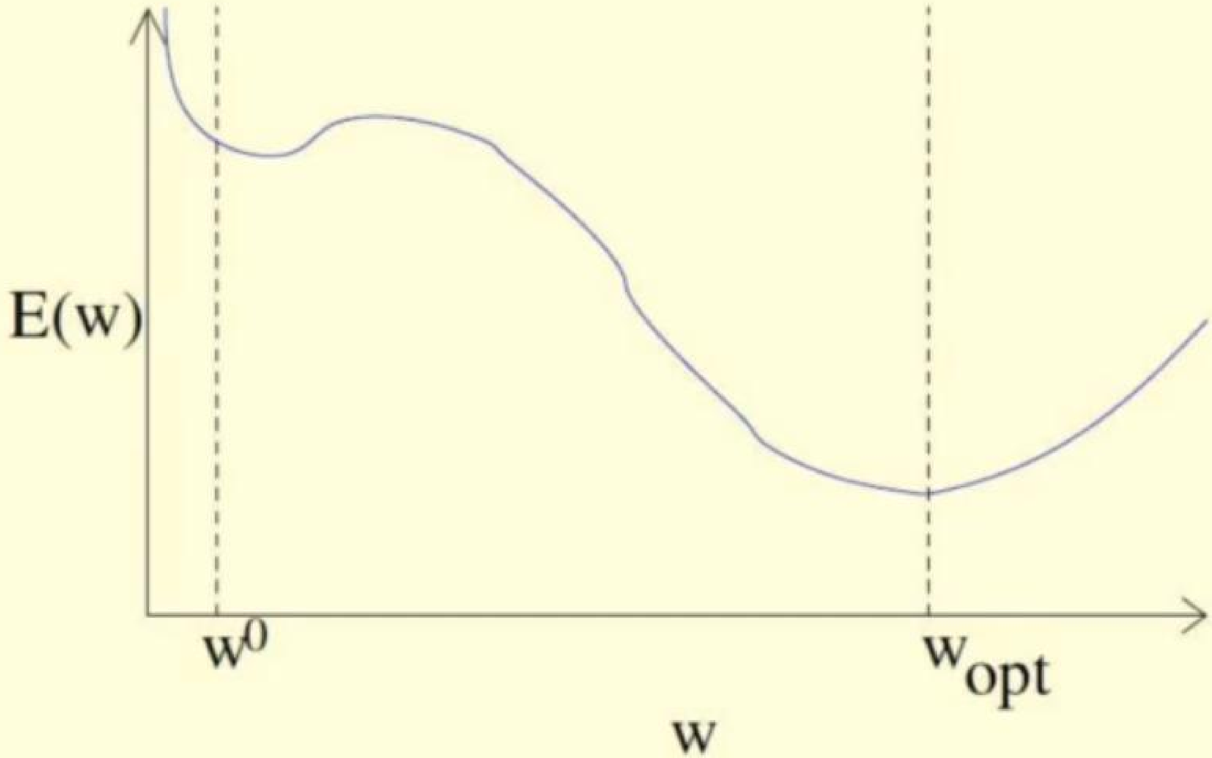
and iterate until convergence.

For a given neural network, $\nabla_{\mathbf{w}} E(\mathbf{w})$ can be easily computed.
This is why we needed to replace sgn() with $\sigma()$.

Convergence of Backprop

Backprop will converge to a local minimum.

Typically works well for small networks.



Overview

- The supervised learning problem
- Perceptron: classification with a linear model
- Neural networks: classification with a non-linear model
- Conclusion

In Everyday Life

[1]

[2]

What are potential use cases for the navy?

1. https://www.publicdomainpictures.net/pictures/360000/velka/atm-cash-machine.jpg.
2. https://www.publicdomainpictures.net/pictures/270000/velka/shopping-online-ecommerce-cons.jpg.

Summary and Outlook

- We covered supervised learning.
- Workhorse of modern AI; well-understood topic.
- Perceptron: limited (linear) representation, optimal solution.
- Neural networks: expressive (non-linear) representation, local optimum.

- We did not cover
- Other topics in AI (unsupervised learning, reinforcement learning, multi-agent systems, knowledge representation and reasoning, robotics, natural language processing, etc.).
- Other algorithms for supervised learning (decision trees, support vector machines, nearest neighbour methods, etc.)
- Engineering aspects (overfitting, training time, feature design, interpretability, etc.).
- Tools/libraries.

- Solution design needs mathematics, programming, domain knowledge.

Thank you!