Classification

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)/x_i \in \mathbb{R}^d$ and y_i is discrete (categorical/qualitative), $y_i \in \mathbb{Y}$.

Example $\mathbb{Y} = \{-1, +1\}, \mathbb{Y} = \{0, 1\}$.

Task: Learn a classification function:

$$f : \mathbb{R}^d \longrightarrow \mathbb{Y}$$

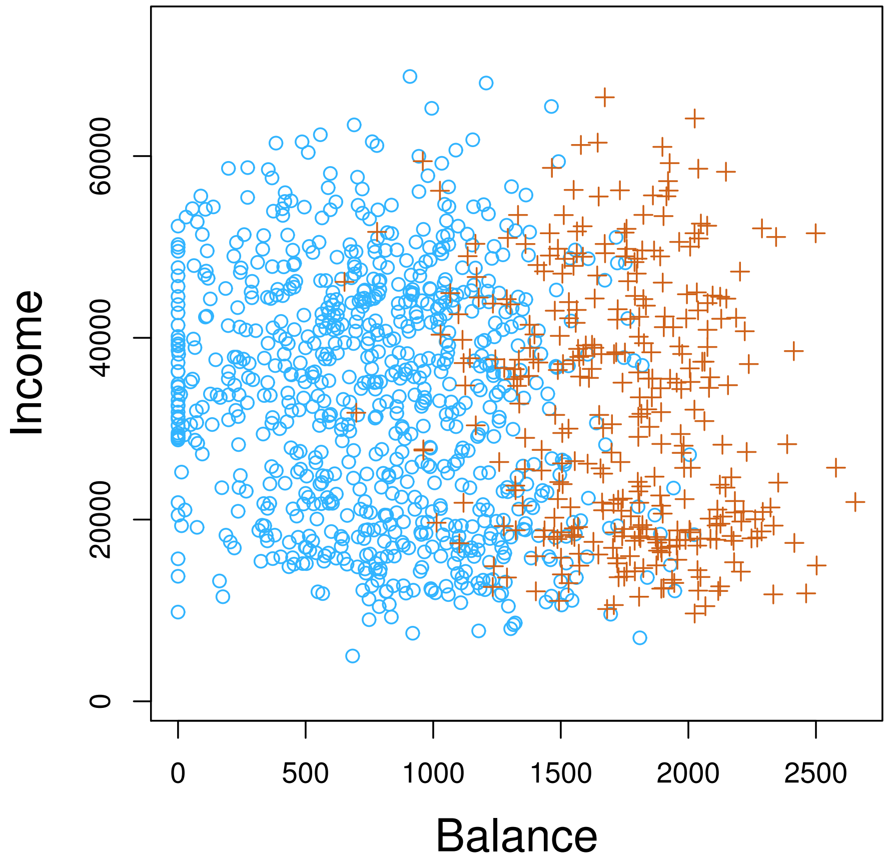
Linear Classification: A classification model is said to be linear if it is represented by a linear function f (linear hyperplane)

Classification: examples

1. Fruit classification → Banana/Orange?
2. Email Spam/Ham → Which email is junk?
3. Tumor benign/malignant → Which patient has cancer?
4. Credit default/not default → Which customers will default on their credit card debt?

Balance	Income	Default
300	$20,000.00	no
2000	$60,000.00	no
5000	$45,000.00	yes
.	.	.
.	.	.
.	.	.

Classification: example

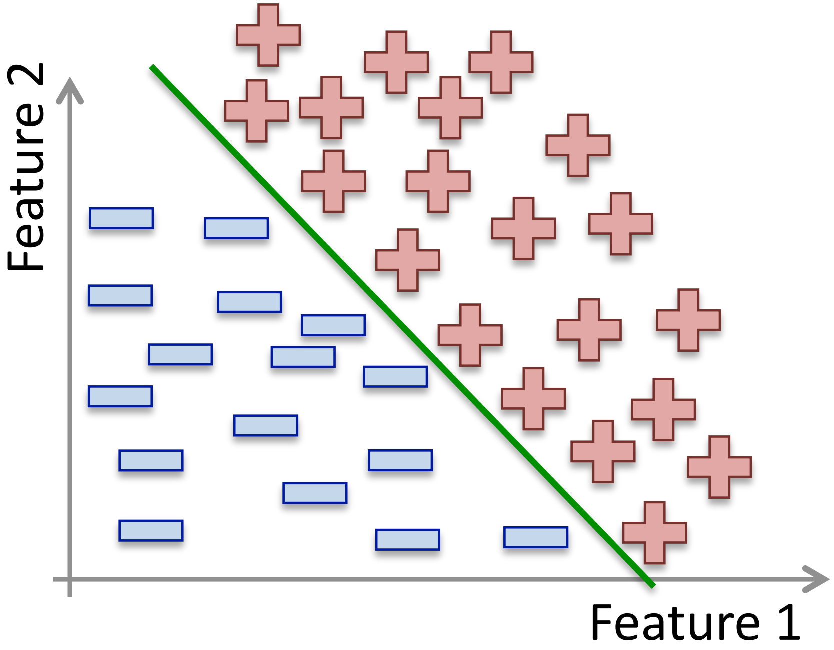


Credit: Introduction to Statistical Learning.

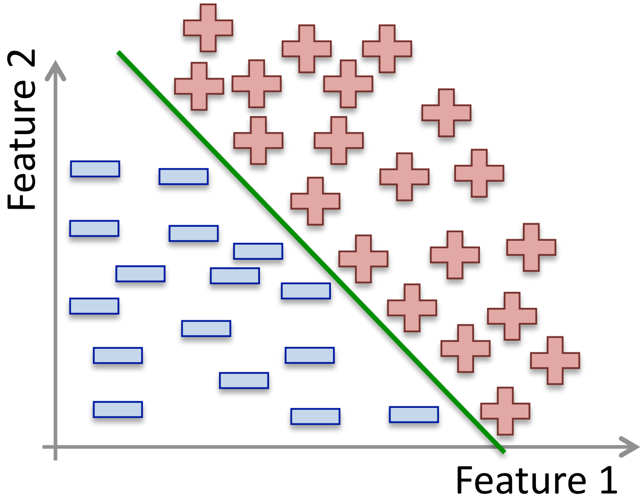
Perceptron

- Belongs to Neural Networks class of algorithms (algorithms that try to mimic how the brain functions).
- The first algorithm used was the Perceptron (Resenblatt 1959).
- Worked extremely well to recognize:
  1. handwritten characters (LeCun et a. 1989),
  2. spoken words (Lang et al. 1990),
  3. faces (Cottrel 1990)
- NN were popular in the 90's but then lost some of its popularity.
- Now NN back with deep learning.

Perfectly separable data

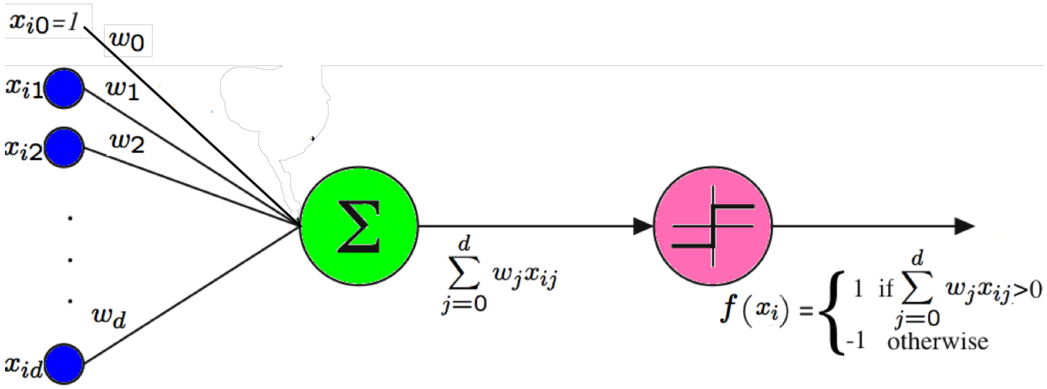


Perceptron



- Linear classification method.
- Simplest classification method.
- Simplest neural network.
- For perfectly separated data.

Perceptron


$x_{i0}=1$
w_0
x_{i1}
w_1
x_{i2}
w_2
w_d
x_{id}
Σ
$\sum_{j=0}^{d} w_j x_{ij}$
$f(x_i) = \begin{cases} 1 & \text{if } \sum_{j=0}^{d} w_j x_{ij} > 0 \\ -1 & \text{otherwise} \end{cases}$

Given n examples and d features.

$$f(x_i) = sign(\sum_{j=0}^{d} w_j x_{ij})$$

Perceptron

- Works perfectly if data is linearly separable. If not, it will not converge.
- Idea: Start with a random hyperplane and adjust it using your training data.
- Iterative method.

Perceptron

Perceptron Algorithm

Input: A set of examples, $(x_1, y_1), \cdots, (x_n, y_n)$

Output: A perceptron defined by $(w_0, w_1, \cdots, w_d)$

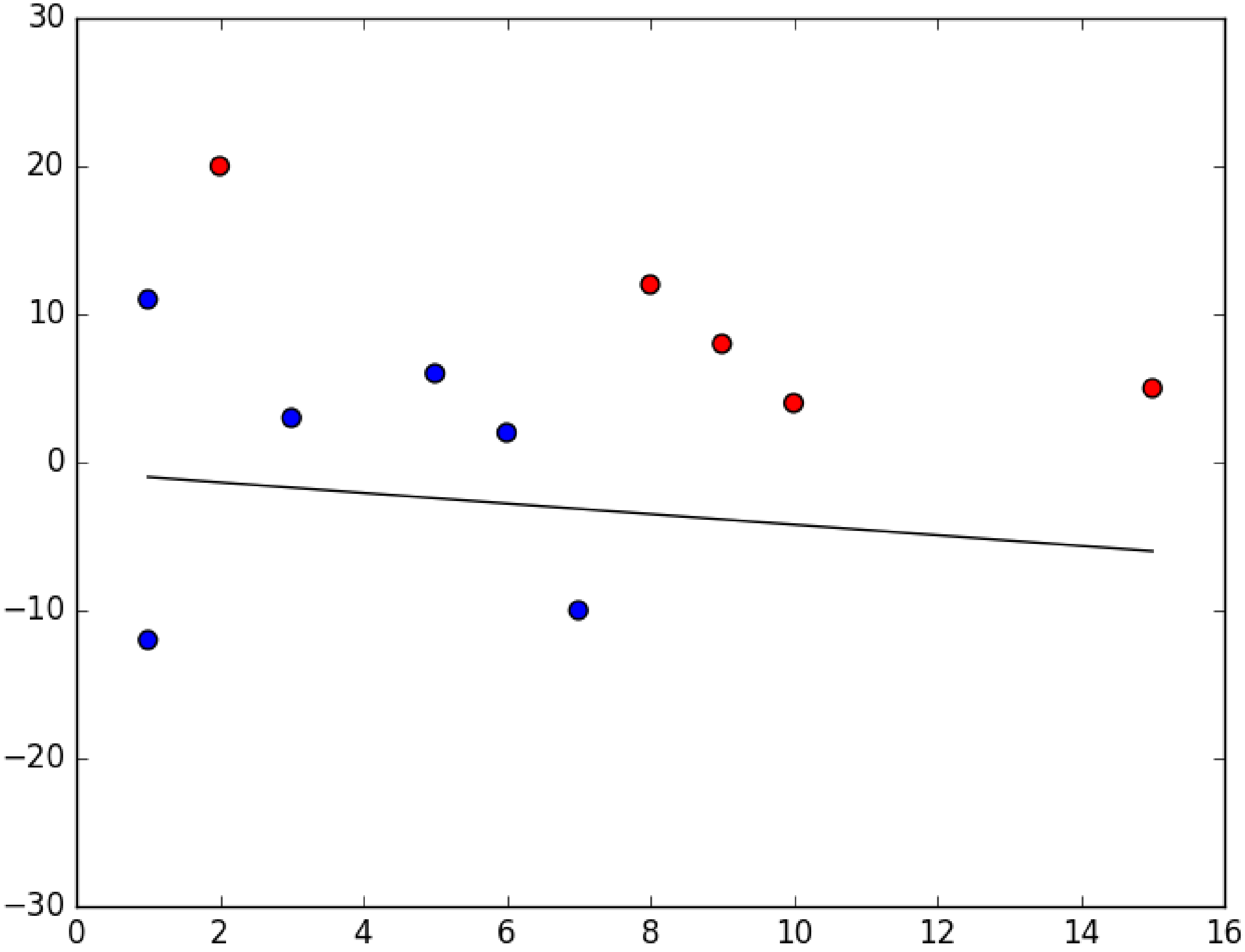
```
Begin
    2. Initialize the weights $w_j$ to 0 $\forall j \in \{0, \cdots, d\}$
    3. Repeat until convergence
        4. For each example $x_i$ $\forall i \in \{1, \cdots, n\}$
        5.     if $y_i f(x_i) \leq 0$          #an error?
        6.         update all $w_j$ with $w_j := w_j + y_i x_i$ #adjust the weights
End
```

Perceptron

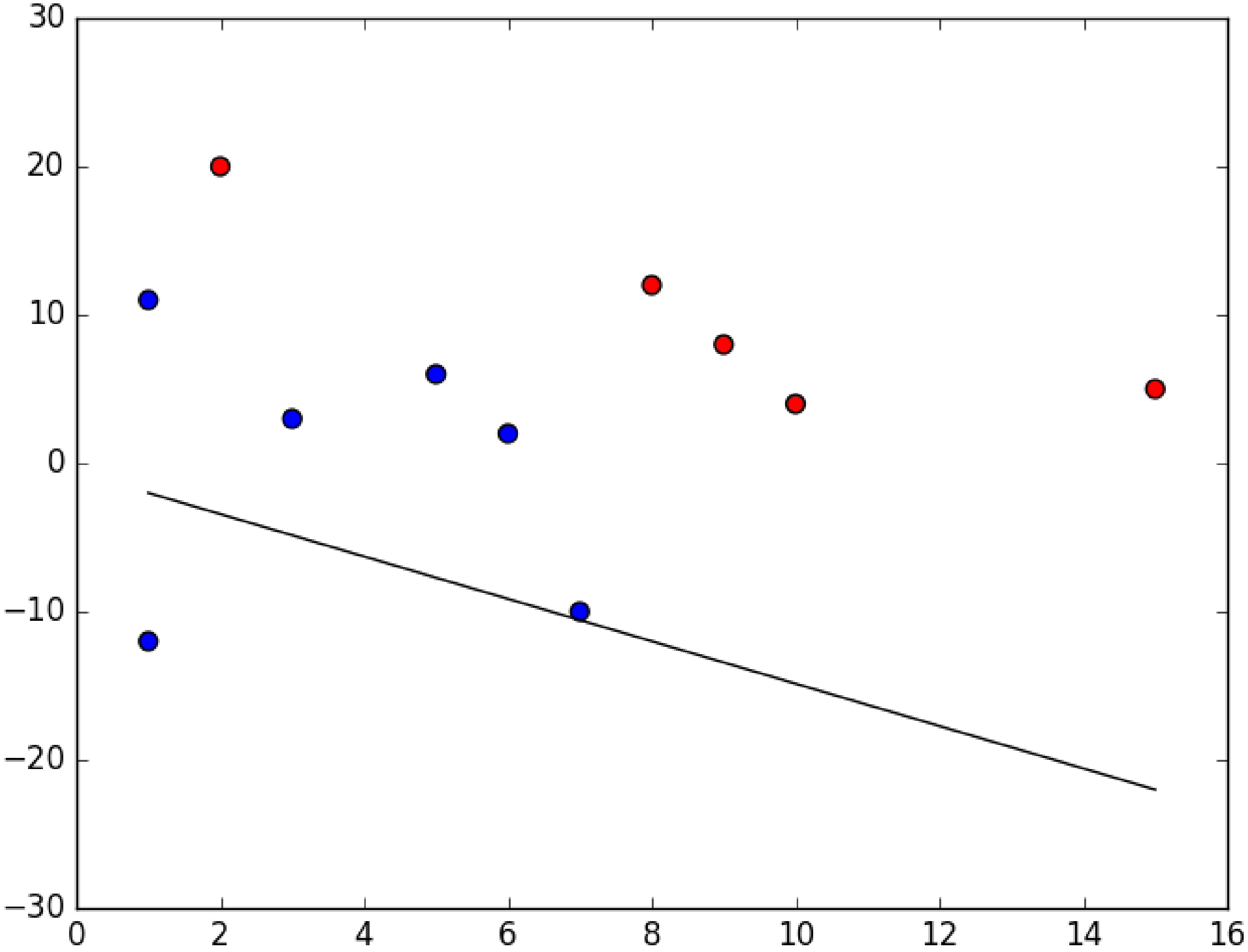
Some observations:

- The weights $w_1, \ldots, w_d$ determine the slope of the decision boundary.
- w_0 determines the offset of the decision boundary (sometimes noted b).
- Line 6 corresponds to:
  Mistake on positive: add x to weight vector.
  Mistake on negative: substract x from weight vector.
  Some other variants of the algorithm add or subtract 1.
- Convergence happen when the weights do not change anymore (difference between the last two weight vectors is 0).

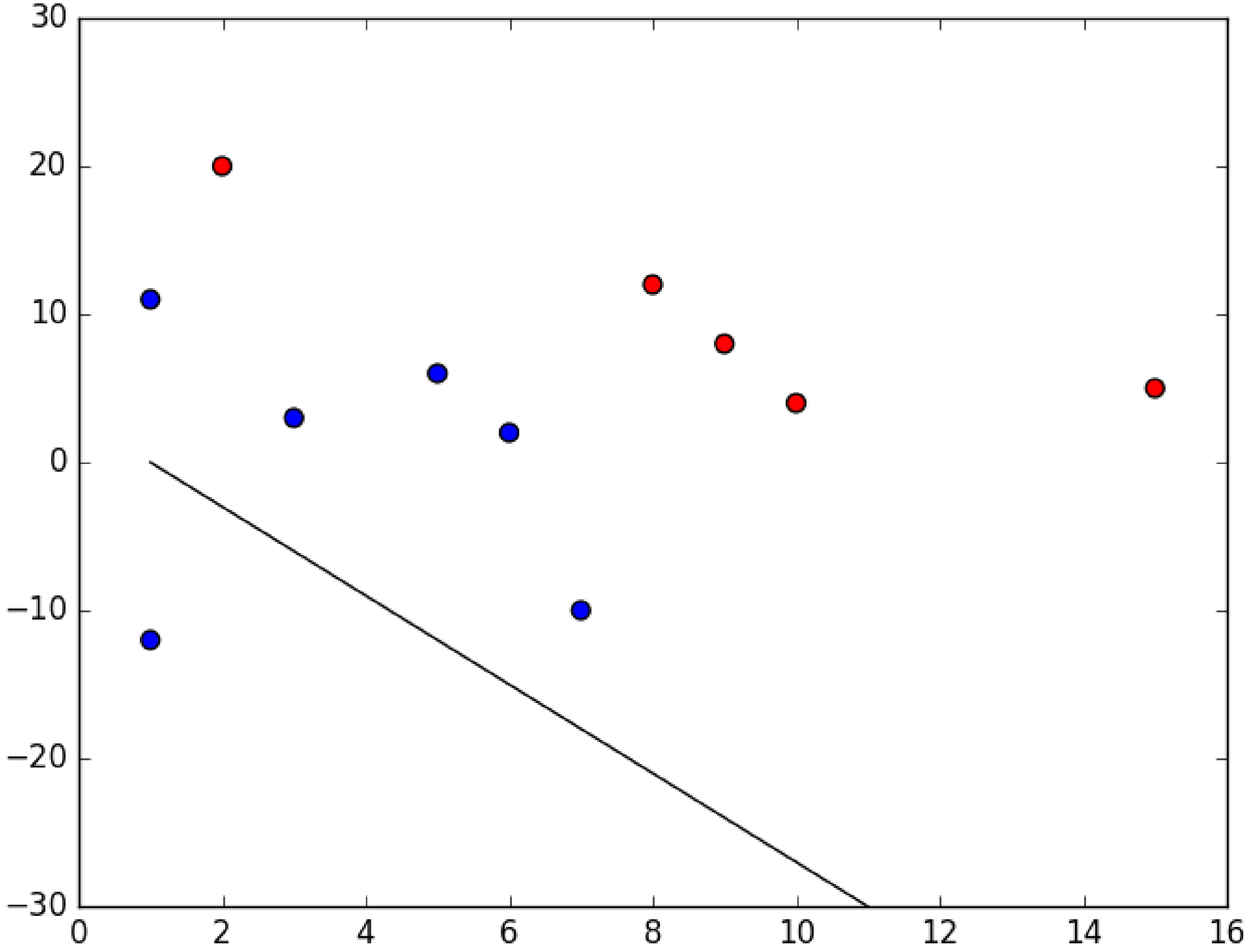
Perceptron: Example



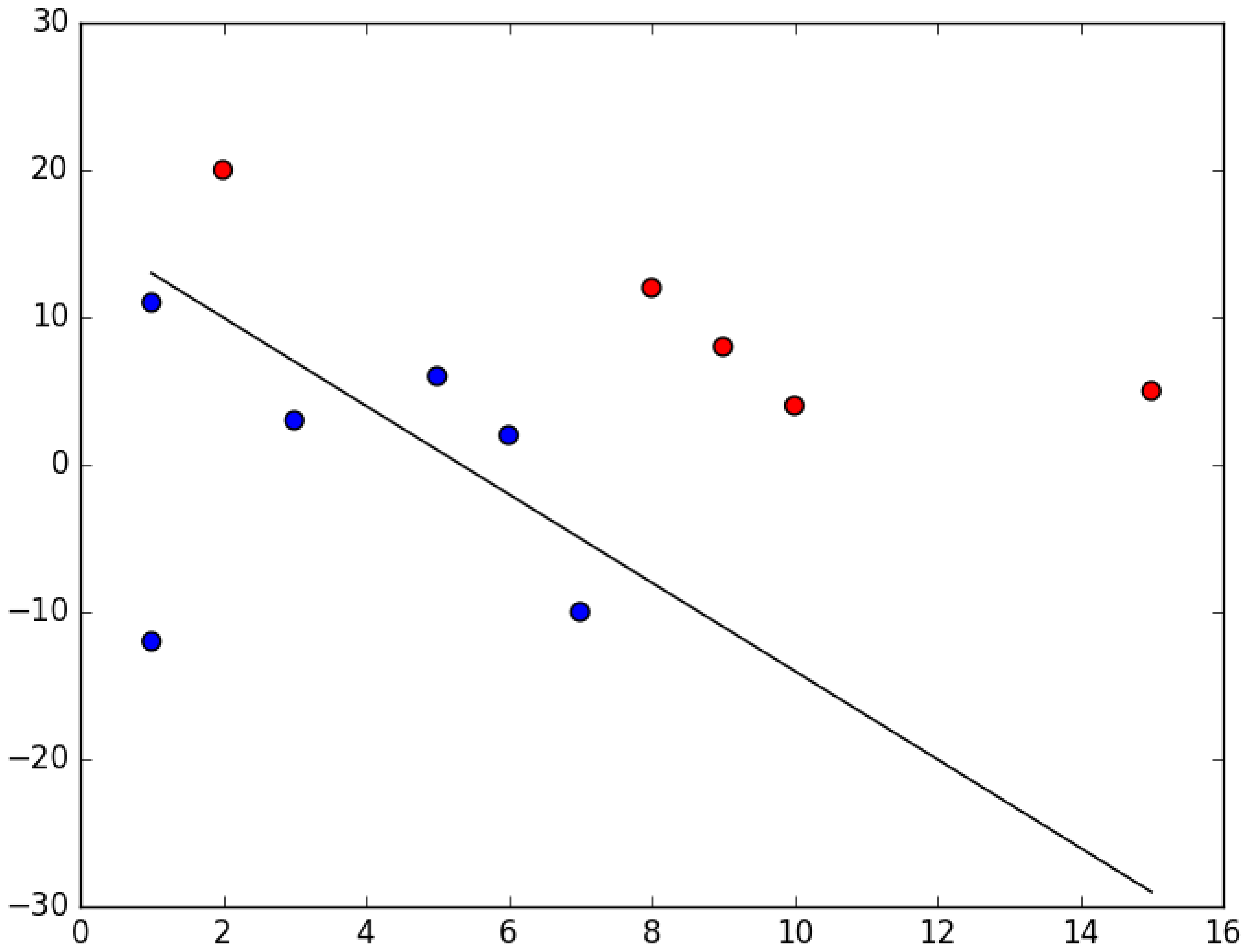
Perceptron: Example



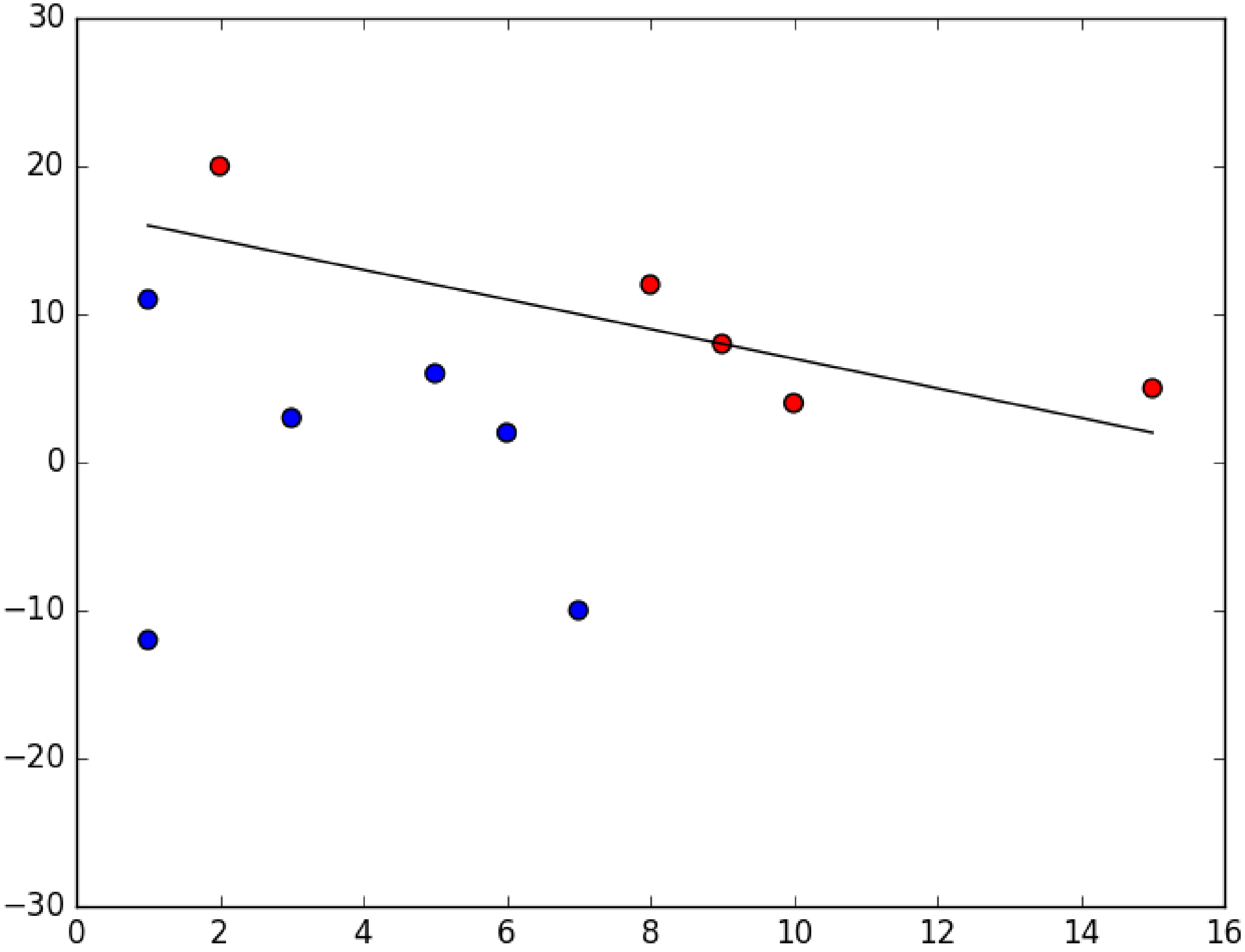
Perceptron: Example



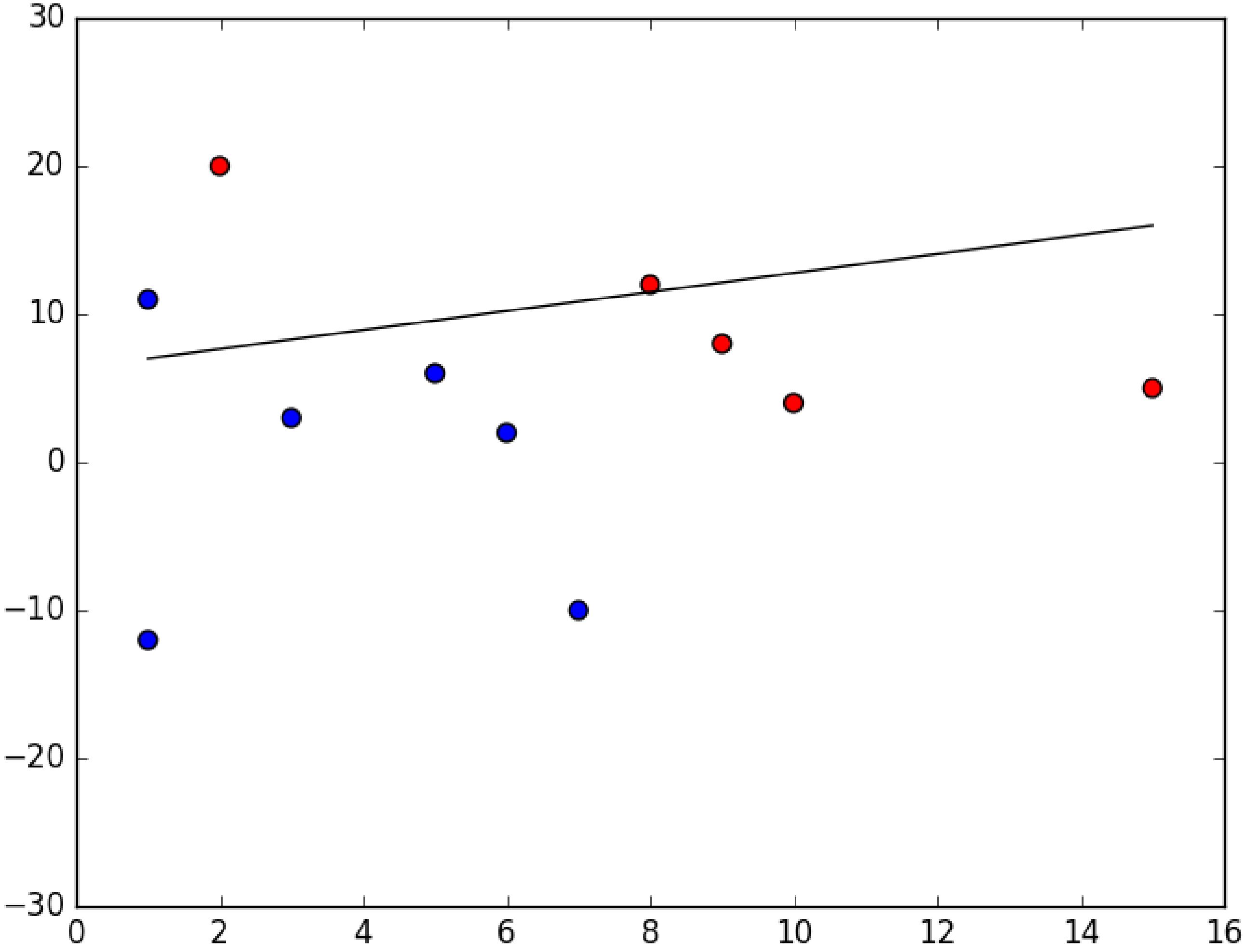
Perceptron: Example



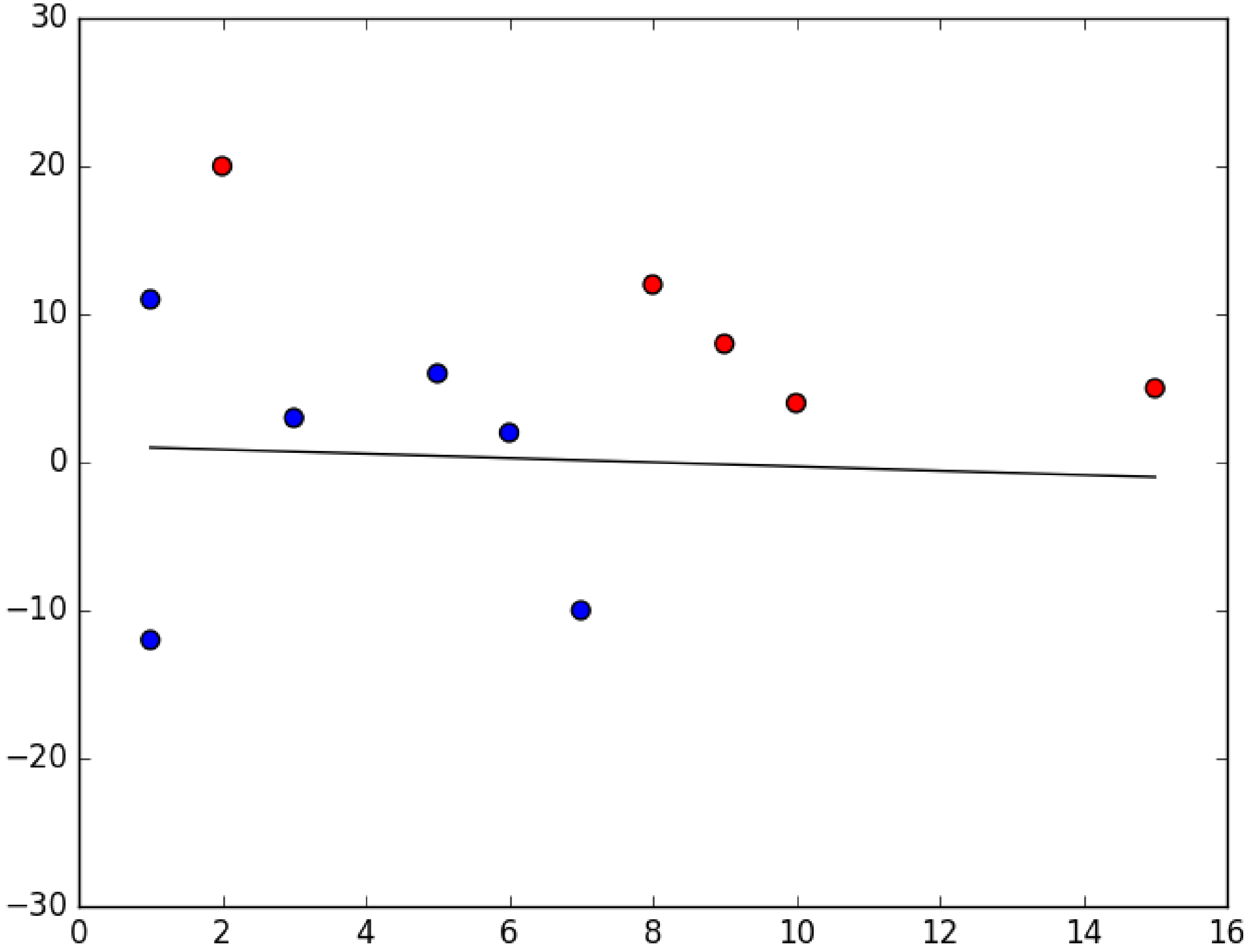
Perceptron: Example



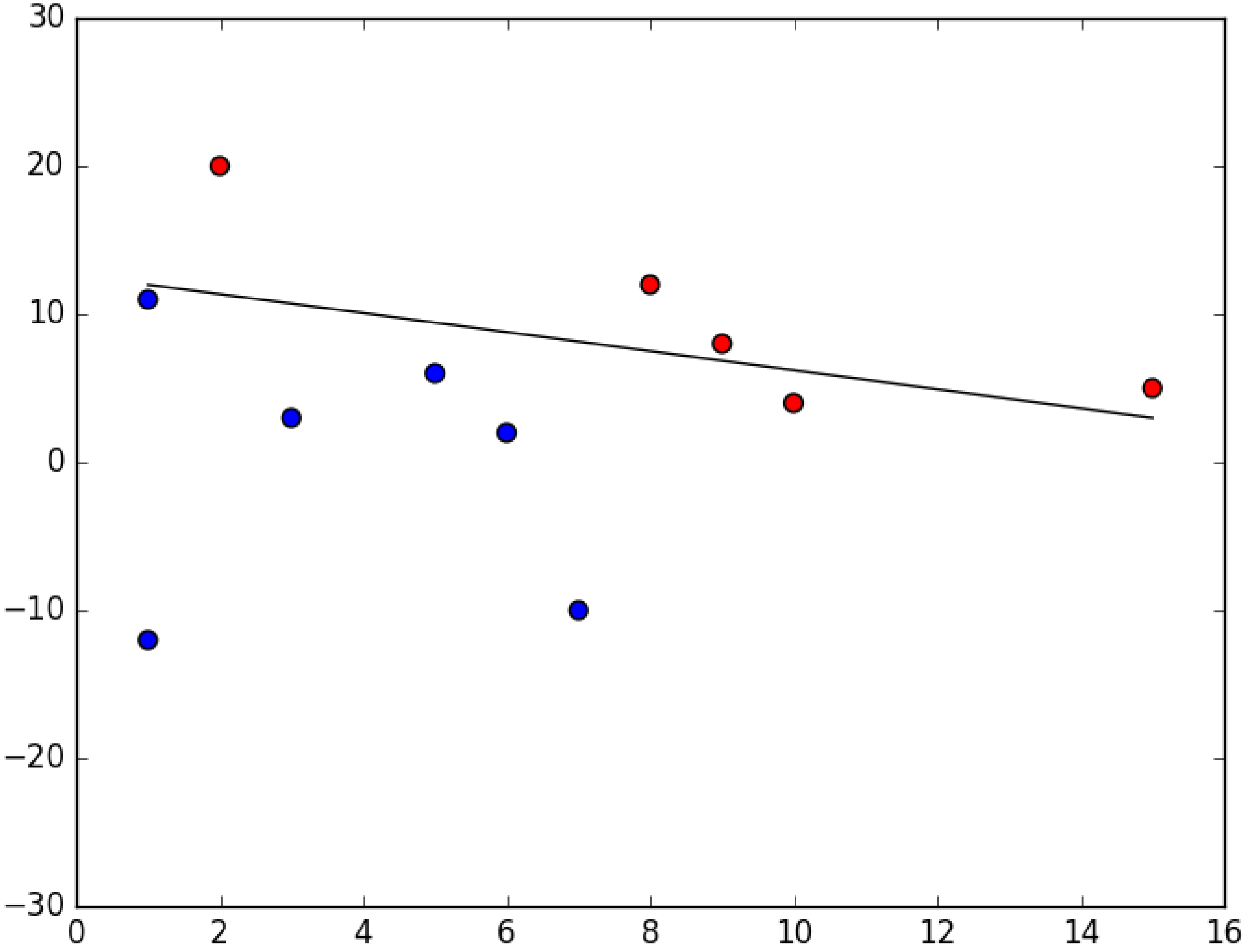
Perceptron: Example



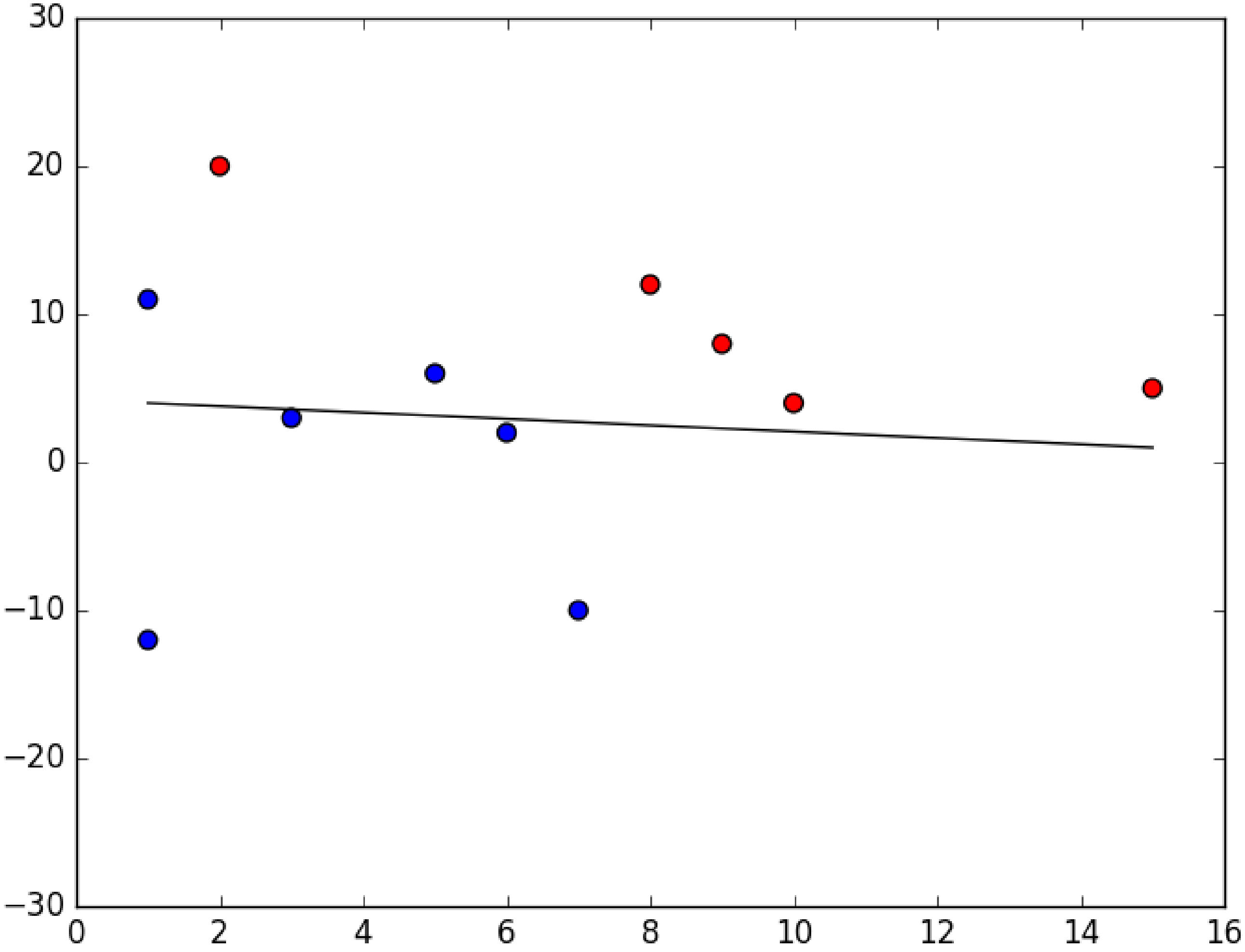
Perceptron: Example



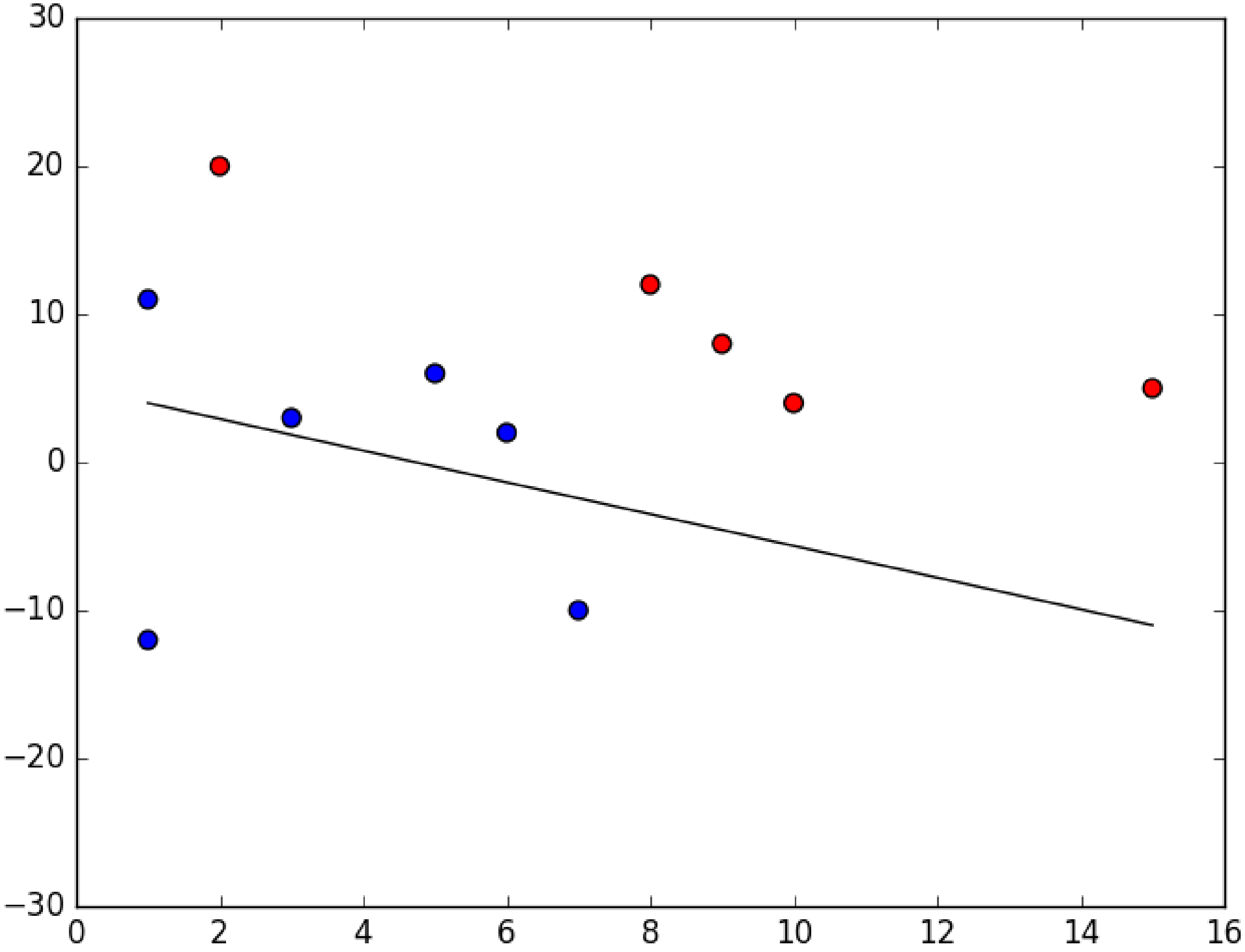
Perceptron: Example



Perceptron: Example

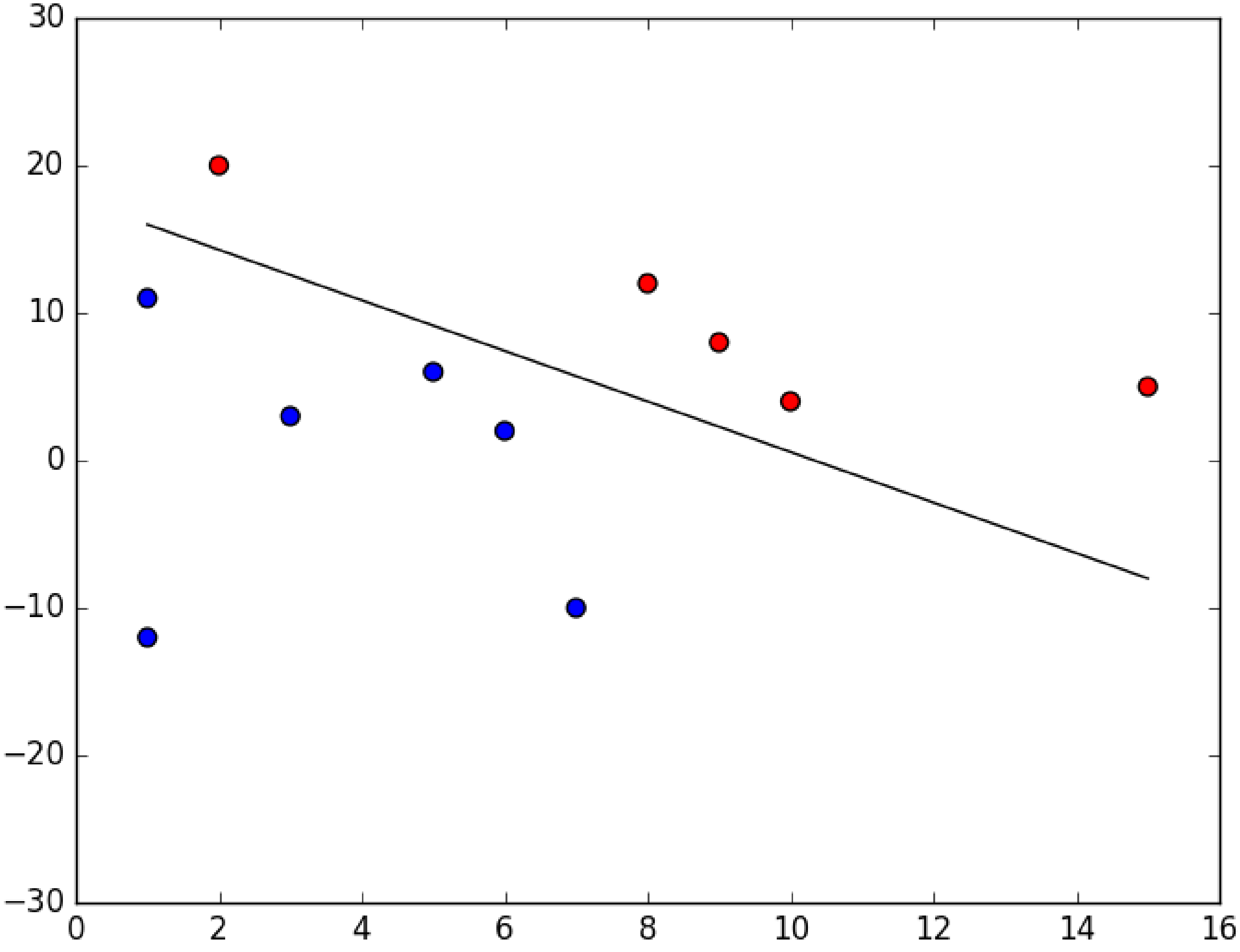


Perceptron: Example



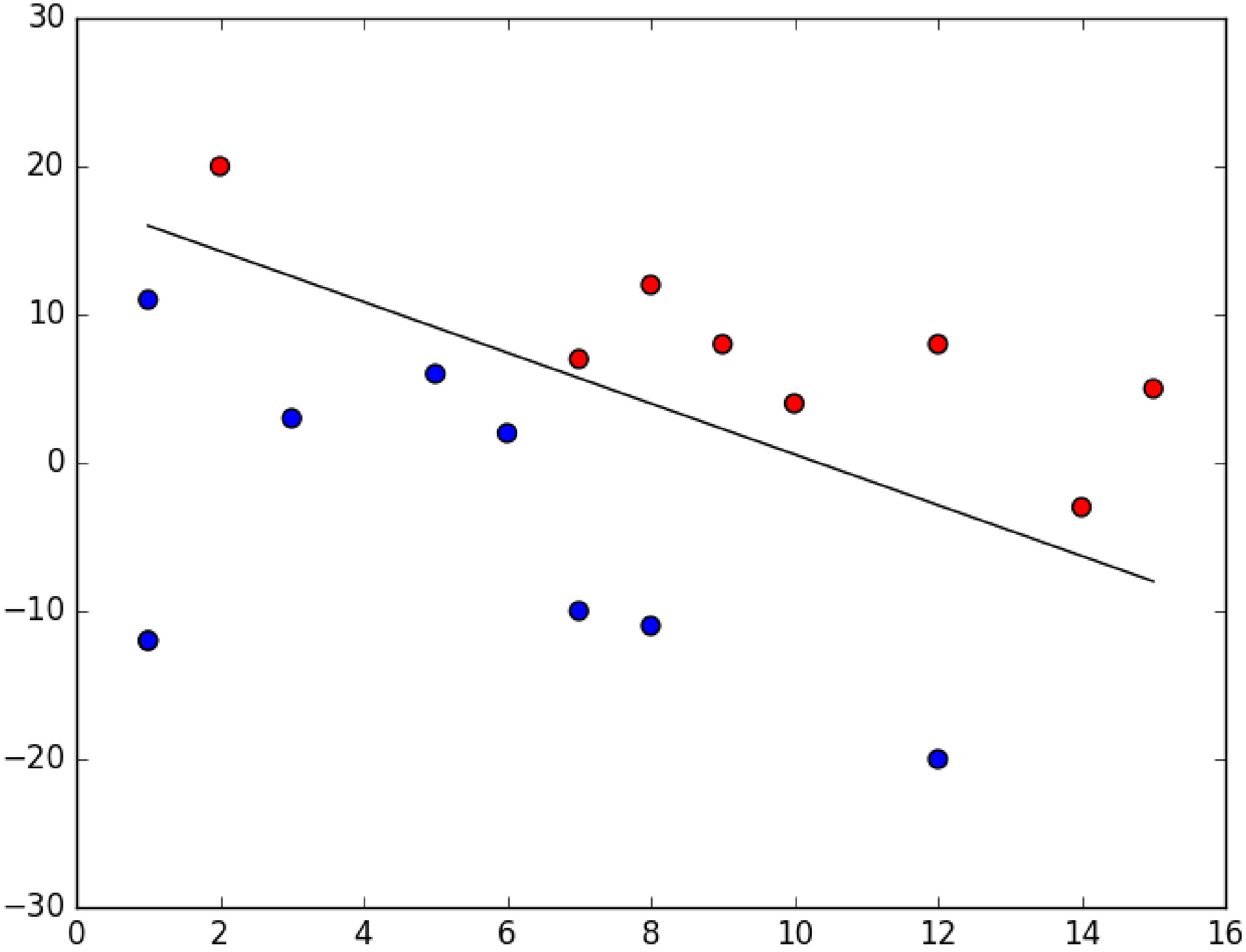
Perceptron: Example

Finally converged!



Perceptron: Example

With some test data:



Perceptron

- The w_i determine the contribution of x_i to the label.

- $-w_0$ is a quantity that $\sum_{i=1}^{n} w_i x_1$ needs to exceed for the perceptron to output 1.

- Can be used to represent many Boolean functions: AND, OR, NAND, NOR, NOT but not all of them (e.g., XOR).

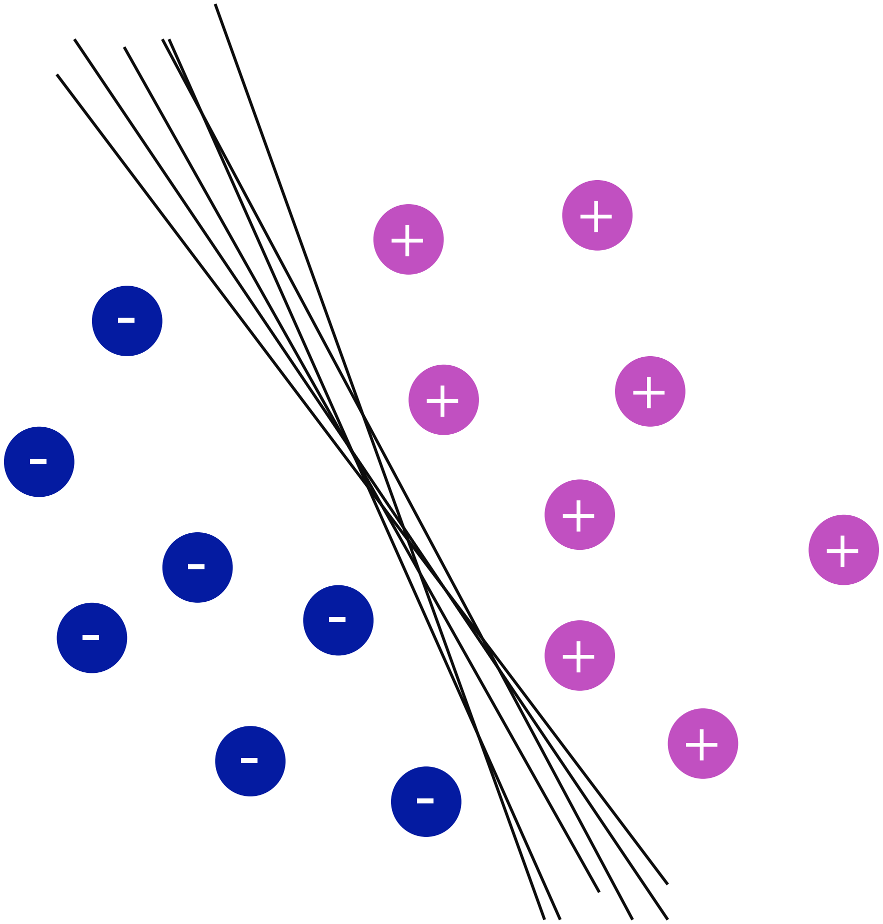
From perceptron to NN

- Neural networks use the ability of the perceptrons to represent elementary functions and combine them in a network of layers of elementary questions.

- However, a cascade of linear functions is still linear!

- And we want networks that represent highly non-linear functions.

Choice of the hyperplane



Lots of possible solutions!

Digression: Idea of SVM is to find the optimal solution.

Credit

- The elements of statistical learning. Data mining, inference, and prediction. 10th Edition 2009. T. Hastie, R. Tibshirani, J. Friedman.
- Machine Learning 1997. Tom Mitchell.