

Tutorial 3: Supervised learning and feedforward neural networks

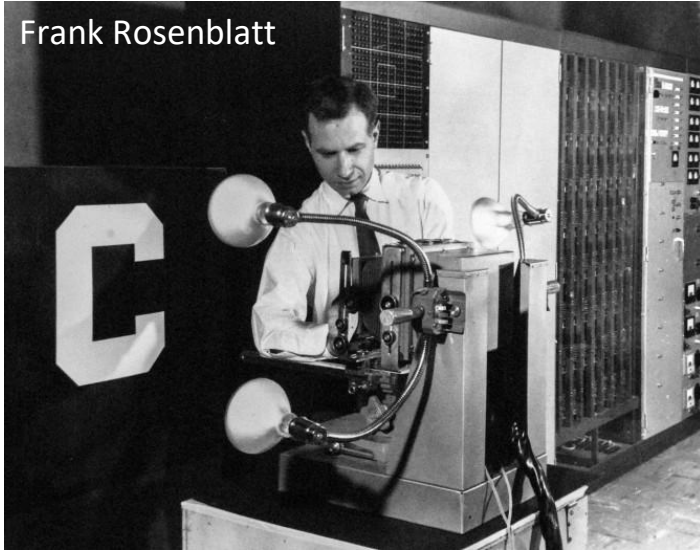
Plan

- Perceptron.
- Multi-layer Perceptron.
- Convolutional Neural Network and digit recognition
- Analogy between convolutional networks and visual cortex

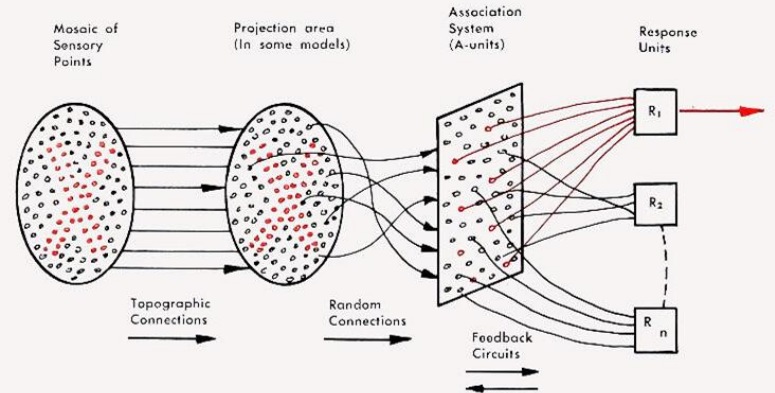


Perceptron

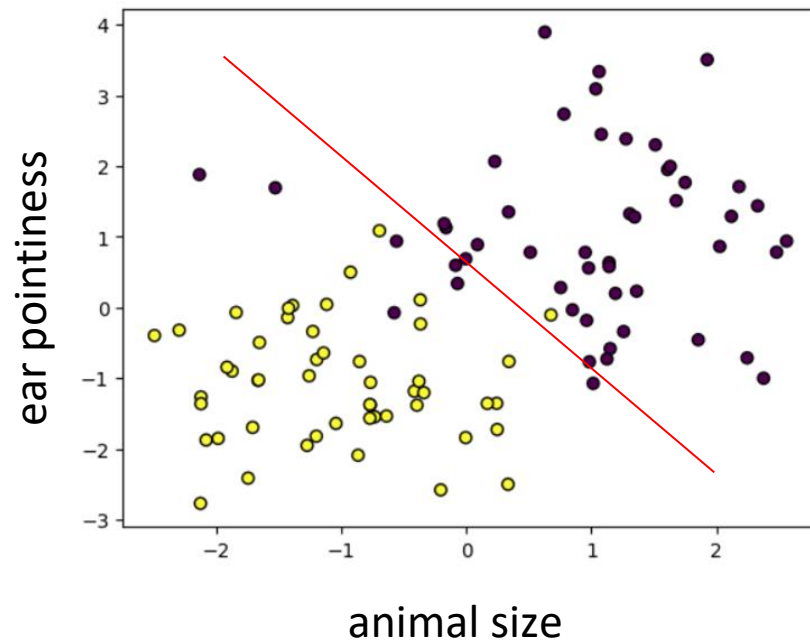
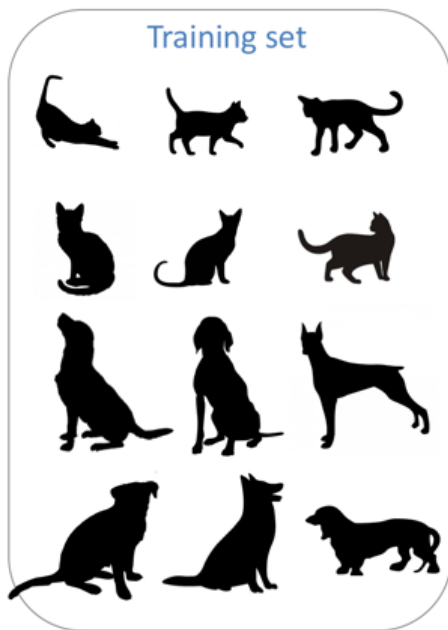
Frank Rosenblatt



1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

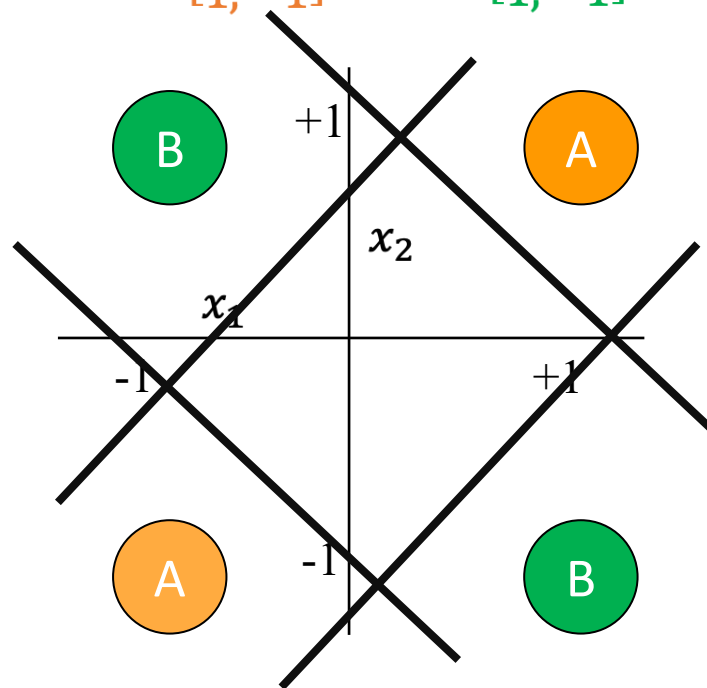


Linearly separable data

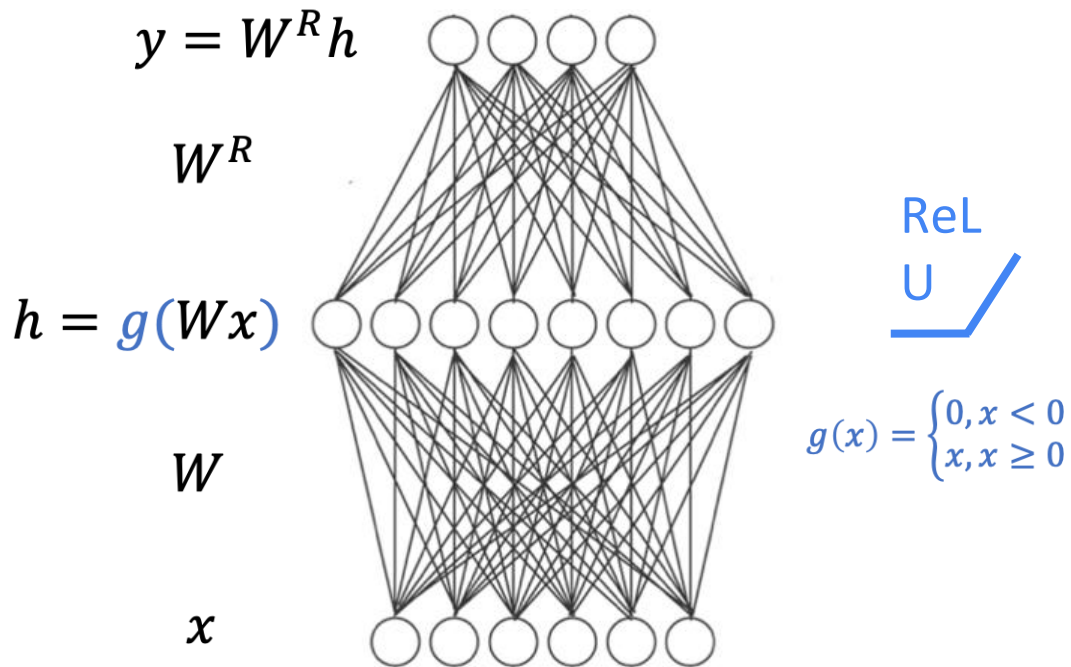


The XOR problem

$$x^A = \begin{bmatrix} 1, -1 \\ 1, -1 \end{bmatrix} \quad x^B = \begin{bmatrix} -1 & 1 \\ 1, -1 \end{bmatrix}$$



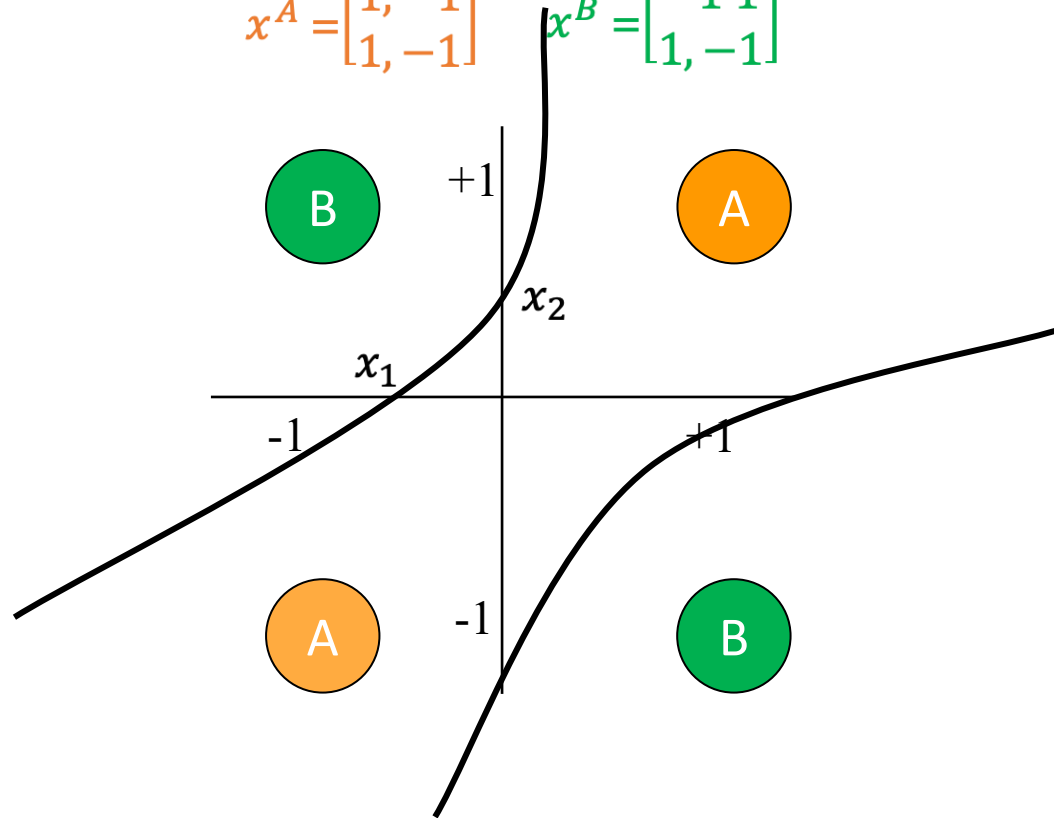
Multi-layer perceptron



deep nonlinear network

The XOR problem

$$x^A = \begin{bmatrix} 1, -1 \\ 1, -1 \end{bmatrix} \quad x^B = \begin{bmatrix} -1, 1 \\ 1, -1 \end{bmatrix}$$

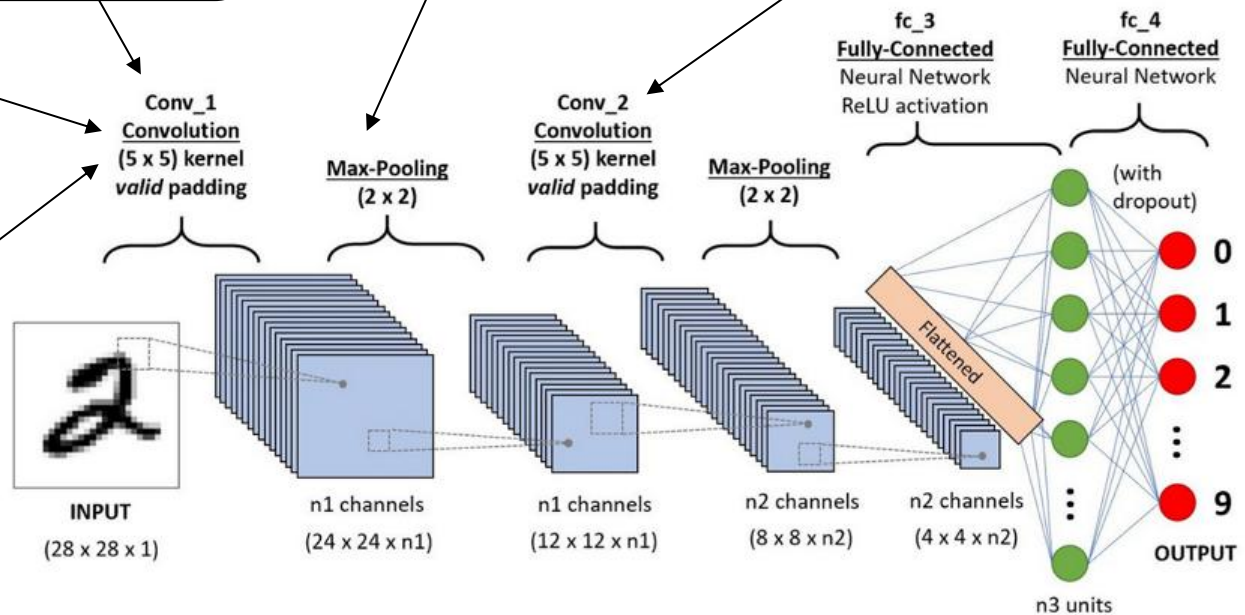
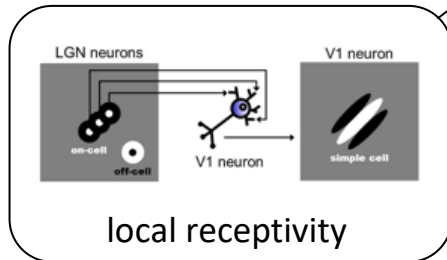
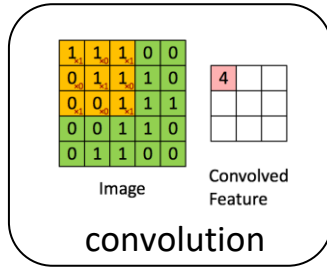
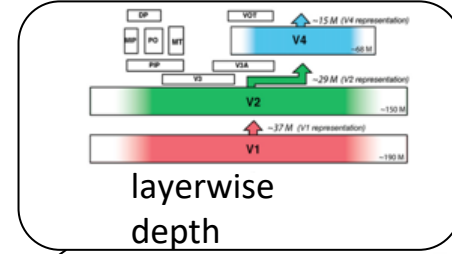
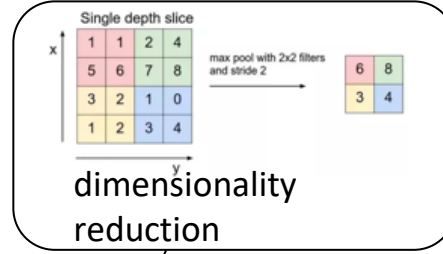




Convolutional neural networks

$$b_i = \frac{a_i}{(k + \alpha \cdot \sum a_j^2)^\beta}$$

divisive
normalisation





```
import torch, torch.nn as nn
```

```
x = torch.tensor([[1.0], [2.0]])
```

```
y = torch.tensor([[3.0], [5.0]])
```

```
model = nn.Linear(1, 1)
```