UNIVERSITY OF LEEDS

Class: Machine Learning

Neural Networks: Perceptron

Instructor: Matteo Leonetti

Learning outcomes

UNIVERSITY OF LEEDS

- Describe the biological principles that inspired neural networks.
- Draw the diagram of the McCulloch and Pitts's neuron.
- Distinguish between generative and discriminative learning models.

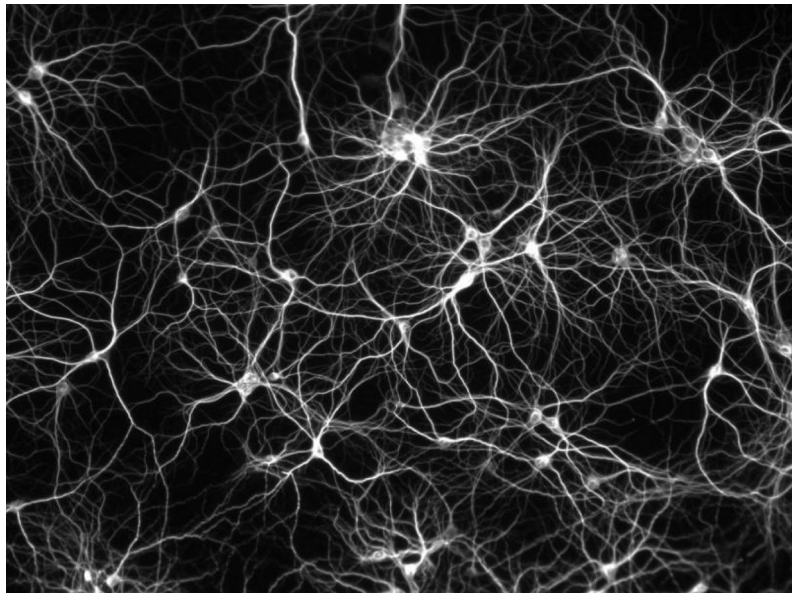
Demo

UNIVERSITY OF LEEDS

https://cs.stanford.edu/people/karpathy/convnetjs/

Our brain

UNIVERSITY OF LEEDS



Neural networks are models for classification and regression there were originally inspired by the brain.

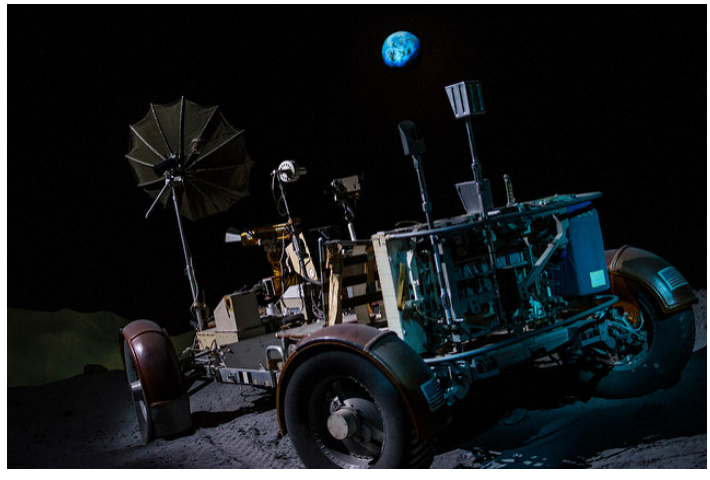
Research on Neural Networks in computer science, however, has since significantly diverged from neuroscience.

Neurons

UNIVERSITY OF LEEDS

10^{11} neurons 10^{15} synapses

Enough fibres to cover the distance to the moon and back

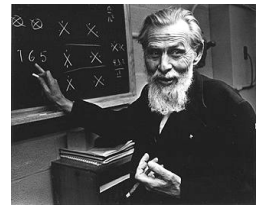


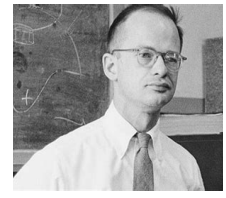
History

UNIVERSITY OF LEEDS

- 1943 – McCulloch and Pitts

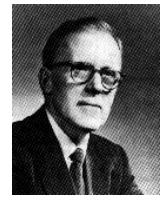
  A Logical Calculus of the Ideas Immanent in Nervous Activity





- 1949 Donald Hebb

  The Organization of Behavior



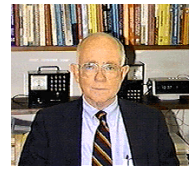
- 1958 Frank Rosenblatt
  The perceptron: A probabilistic model for information storage and organization in the brain.

History

UNIVERSITY OF LEEDS

- 1962, Widrow and Hoff

  Adaptive Switching Circuits



- *1965 Alexey Ivakhnenko*

- 1975, Paul Werbos

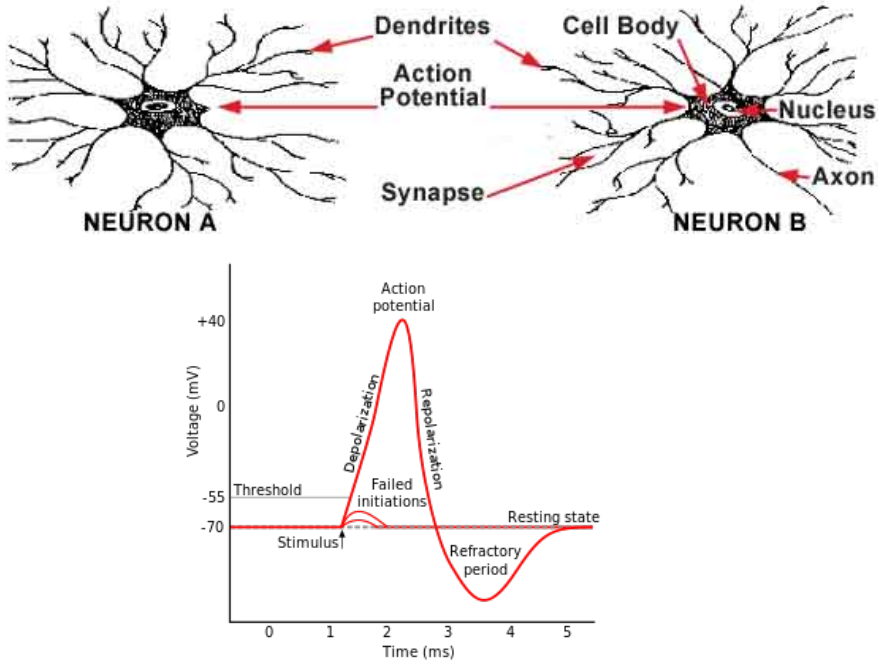
  Backpropagation

- …

- 2009- now, Deep Learning: Dayan (Max Planck Tubingen), Schmidhuber (IDISA), Hinton (Google, U Toronto) , Bengio (U Montreal), LeCun (FAIR), … and more!

Firing

UNIVERSITY OF LEEDS

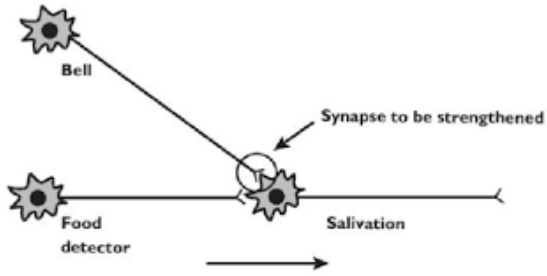


Neurons are connected by dendrites.

The fundamental mechanism that artificial neural networks imitate is the spike generated when the stimulus applied to a neuron overcomes a given threshold.

Hebbian learning

UNIVERSITY OF LEEDS



When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased

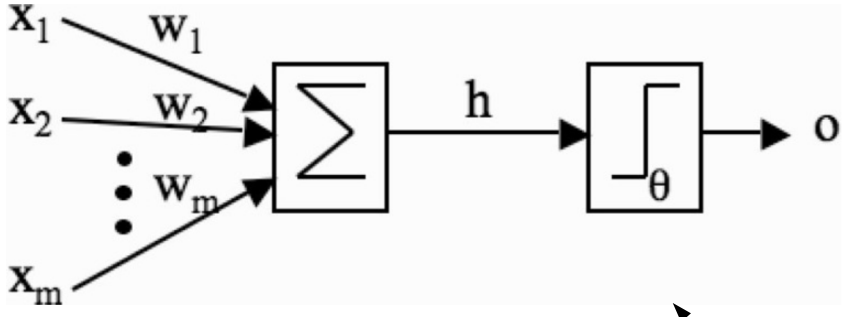
Donald Hebb (1949)

Stimuli that repeatedly appear together cause the connections of the corresponding neurons to become stronger.

This principle is behind Hebbian learning, and Pavlov's "classical conditioning".

McCulloch and Pitts Neuron

UNIVERSITY OF LEEDS



$$h_w(\boldsymbol{x}) = \sum_i w_i x_i = \boldsymbol{w} \cdot \boldsymbol{x}$$

$$o(h_w) = \begin{cases} 1 & \text{if } h_w > \theta \\ 0 & \text{if } h_w \leq \theta \end{cases}$$

Activation function

The implementation of such principles in software takes the form of McCulloch and Pitts's neuron.

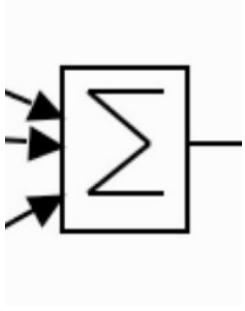
Each input is multiplied by a weight, and all the results are summed together.

The sum is then fed to an activation function, which outputs 1 if the sum is above a threshold, and 1 otherwise.

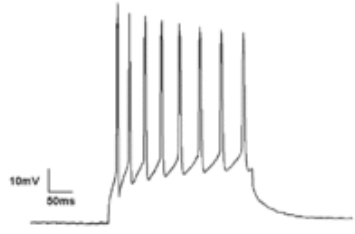
Model Critique

UNIVERSITY OF LEEDS

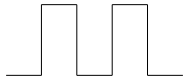
?



Single threshold vs spike train



No clock → asynchronous



Is this model realistic?

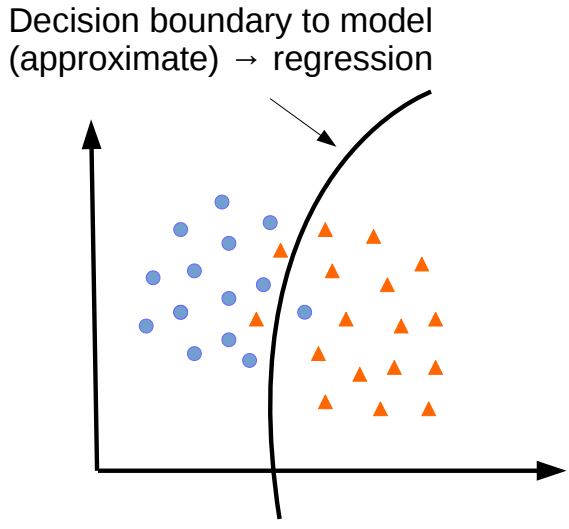
It has not been established that what neurons compute is actually a sum.

The modelled neurons sends a single spike down the axon, while real neurons send trains of spikes, and the frequency encodes information.

Real neural networks are asynchronous, while in our model all inputs must be present for a neuron to compute its output.

A function approximator

UNIVERSITY OF LEEDS



Generative vs Discriminative model

We use neural networks to classify the input, which means learning a boundary function between the classes.

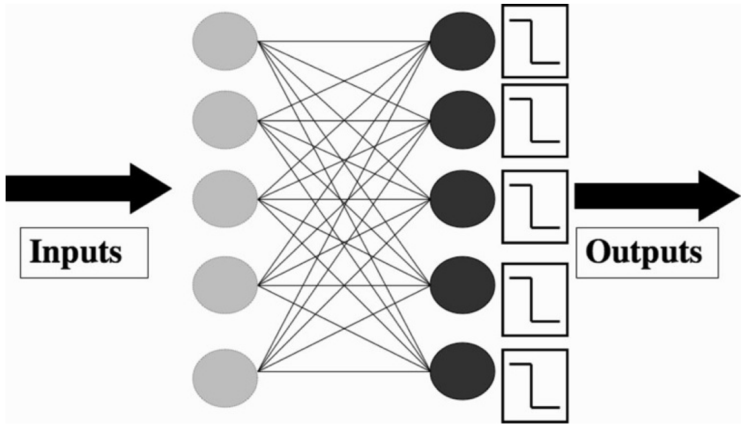
A generative model can, as the name suggests, generate new data. Typically, generative models learn a distribution of the data, so that new data can be sampled from it (example: regression over the parameters of a certain distribution).

A discriminative model learns the decision boundary between the classes, and cannot be used to generate new data points.

Neural Networks (NNs) can be used both as a generative and a discriminative model. In this module, we will only study discriminative NNs.

Perceptron

UNIVERSITY OF LEEDS



A perceptron is a model with a number of independent neurons.

Learning adjusts the weights of the neurons.

Typically, there is one output neuron per class.

Training the perceptron

UNIVERSITY OF LEEDS

Learning happens through **optimisation**.

We define an error function, and then an optimisation algorithm finds the parameters that obtain the minimum error.

For example, the error function is the total number of mistakes:

$$E(\boldsymbol{X}) = \sum_{\vec{x}_n \in \boldsymbol{X}} |y_n - t_n|$$

Where y_n is the output of the perceptron on point n, and $t_n \in \{0,1\}$ is the desired class, and $\boldsymbol{X}$ is the dataset.

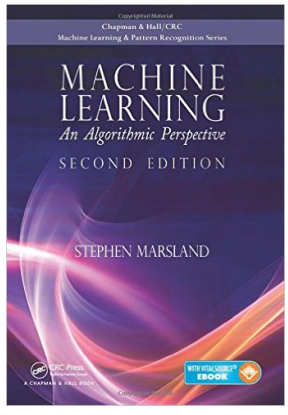
Conclusion

Local methods stop when the gradient is close to zero, which means that they are close to a stationary point.

There is no guarantee that such a point is the *global* minimum. Local methods will, in general, converge to a *local* minimum of the objective function.

A local minimum is a point such that all the points around it have a higher value of the objective function.

UNIVERSITY OF LEEDS



Chapter 3, up to 3.3