Foundations of Artificial Neural Networks

Bernd Ulmann

Edinburgh, 04-JUN-2003

What are artifical neural networks (ANN)?

❶ A new paradigm of computing.

❷ A simplified implementation of biologically inspired neural networks.

Where ANNs are better than traditional computers

❶ Adapting to new environments by *learning*.

❷ Processing *fuzzy* (imprecise) data.

❸ Working with noisy or erroneous data.

❹ Performing classification tasks very quickly.

Biological neural networks – the basics

❶ A human brain contains about 10^{11} neurons.

❷ Each neuron has inputs (synapses connected to *dendrites*) and an output (*axon* ending in synapses).

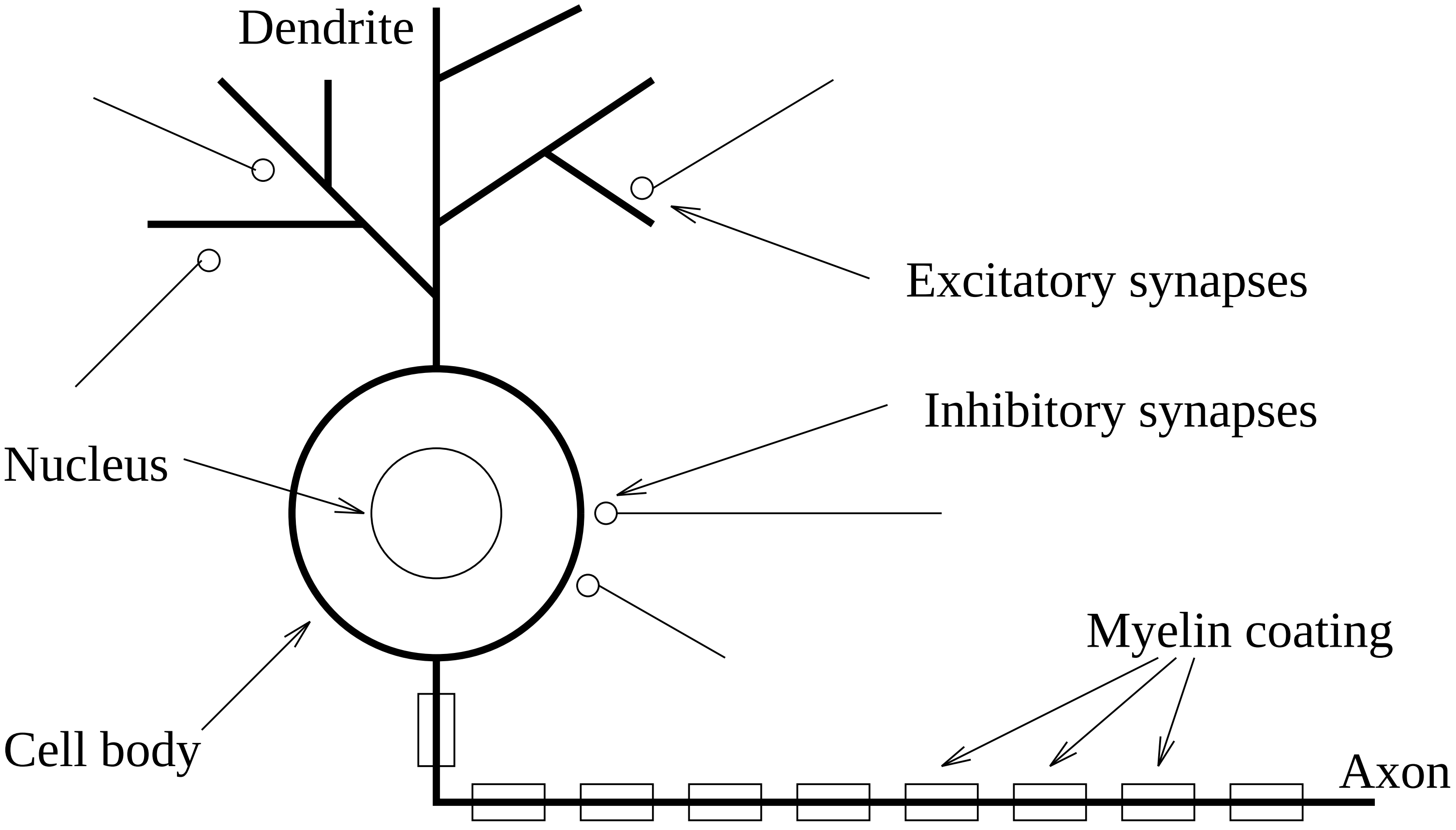
❸ One neuron generates input for $\approx 10^3 \ldots 10^4$ other neurons.

❹ Each neuron receives data from $\approx 10^3 \ldots 10^4$ neurons.

❺ *Synapses* connect neurons to each other.

❻ Signals are transmitted by releasing chemicals (*neurotransmitters*) from a sender neuron causing an electric potential in the receiving cell, etc.

Biological neurons



Simplified model of a neuron

❶ A neuron has a current *activation* $a_j(t)$.

❷ All inputs get numeric data $o_i(t)$ from other neurons.

❸ Inputs are multiplied by *weights* $w_{i,j}$.

❹ Generate a *net input* by employing a *propagation rule*:

$n_j(t) = \sum_i o_i(t) w_{i,j}$.

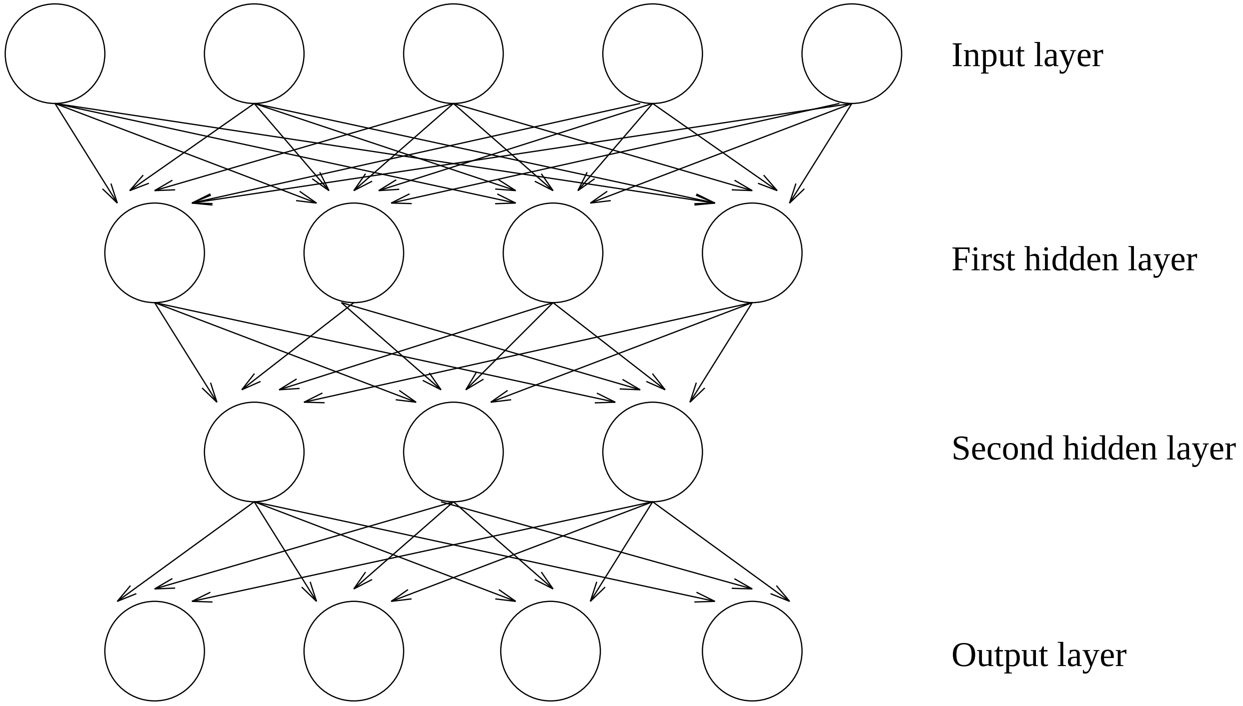
❺ Calculated new activation (Θ is a threshold value)

$a_j(t+1) = f_{\text{act}}\left(a_j(t), n_j(t), \Theta_j\right)$.

❻ A new output is calculated using an output function:

$o_j(t+1) = f_{\text{out}}\left(a_j(t+1)\right)$.

A simple feed forward ANN



Basic types of ANNs

❶ Feed forward ANNs

- Several layers of neurons are connected unidirectionally.
- Connections can skip layers.

❷ Feedback ANNs

- *Direct feedback* networks allow a neuron to have a feedback connection to itself → self amplification/attenuation.
- *Indirect feedback* networks have only feedback connections between layers → a method to control *attention* of the ANN.
- *Lateral feedback* networks have feedback connections only within a layer → *winner takes all* network.

Implementing ANNs

❶ Synchronous activation: All neurons change their states simultaneously → very useful for SIMD parallel computers.

❷ Asynchronous activation:

Fixed order activation**:** Order of activation is fixed.

Random order activation**:** Order is (pseudo)random – not every neuron is calculated, some are calculated multiply.

Random permutation activation**:** As before but every neuron is calcuted once and only once.

Topological order activation**:** Ideal for feedforward networks.

Training (teaching) ANNs by

❶ creating new connections/neurons,

❷ deleting existing connections/neurons,

❸ changing the weights $w_{i,j}$,

❹ changing the threshold Θ_j or by

❺ changing the functions $f_{\text{act}}()$ and/or $f_{\text{out}}()$.

In most cases, the weights $w_{i,j}$ will be modified – this can also be used to simulate additional or deleted connections/neurons.

Basic learning strategies

❶ *Supervised learning*: The learning algorithm receives the output generated by the ANN as well as the desired output and performs changes on the weights $w_{i,j}$. → Desired result has to be known in advance, sometimes not possible/realistic.

❷ *Reinforcement learning*: The ANN receives results in form of reward/punishment from actions based on its generated output.

❸ *Unsupervised learning*: The only goal is to build representations of the input data which can in turn be used for prediction/classification, etc.

The backpropagation learning rule

Idea: Change the weights $w_{i,j}$ according to the difference between the actual output $o_j(t)$ and the teaching value $t_j(t)$ with a somewhat complex mechanism as follows:

$$\Delta w_{i,j} = \eta o_i \delta_j \text{ with}$$

$$\delta_j = \begin{cases} f_{\text{act}}(n_j(t))(t_j(t) - o_j(t)) \text{ for output neurons} \\ f_{\text{act}}(n_j(t)) \sum_k (\delta_k w_{j,k}) \text{ for hidden neurons.} \end{cases}$$

Input data sets

❶ *Training set*: Examples used to train the ANN (for example by changing the weights $w_{i,j}$ according to the backpropagation rule).

❷ *Validation set*: Data set for tuning the ANN (for example by changing the number of hidden layers, etc.).

❸ *Test set*: Data set used to assess the performance of the ANN (obviously never to be used as a training set).

Sometimes the difference between the training and the validation set is not quite clear and depends on the applied learning rule.

References

[1] Klaus Peter Kratzer, *Neuronale Netze, Grundlagen und Anwendungen*, Carl Hanser Verlag 1991.

[2] Armando Freitas da Rocha, *Neural Nets, A Theory for Brains and Machines*, Lecture Notes in Artificial Intelligence 638, Springer-Verlag 1992.

[3] Raul Rojas, *Theorie der neuronalen Netze – Eine systematische Einf”uhrung*, 4., korrigierter Nachdruck, Springer-Verlag 1996.

[4] Warren S. Sarle, *ai-faq/neural-nets/part1*, etc., `ftp://ftp.sas.com/pub/neural/FAQ.html`.

[5] Andreas Zell, *Simulation Neuronaler Netze*, Addision-Wesley publishing company, 1994.