First lecture

Vadim Kulikov

MeiCogSci

04 February 2014

Me as a baby

Organisational issues

- Tuesdays 16-18 lectures
- Thursdays 16-17 exercise class
- Thursdays 17-18 lectures.

If you cannot make it on Thursday, it can be compensated by returning the exercises written on paper or by sending them to me in an e-mail before the following Tuesday lecture.
If you come to the exercise class, you will have to be ready to present your solutions at the black board.

Organisational issues

Those who have done 12 exercises by the end of the course, pass.
If you do less than 12, but at least 8, then you still have the opportunity to pass by completing an extra assignment.
If you get 12 done *and* complete the assignment, you will get 3 intead of 2 credit points!

Organisational issues

Those who have done 12 exercises by the end of the course, pass.
If you do less than 12, but at least 8, then you still have the opportunity to pass by completing an extra assignment.
If you get 12 done *and* complete the assignment, you will get 3 intead of 2 credit points!

General problem: Emergence

A general problem in cognitive science and philosophy of mind: How does meaning, knowledge, memory and consciousness *emerge* from the interconnections of neurons?

What is Emergence?

Is there a mathematical definition of emergence?

Maybe we can understand some of it by studying neural networks and their emergent properties.

More precisely:

What information processing capabilities emerge in hierarchical systems of primitive computing units? What can be computed with these networks? How can these networks determine their structure in a self-organizing manner?

What is Emergence?

Is there a mathematical definition of emergence?
Maybe we can understand some of it by studying neural networks and their emergent properties.

More precisely:
What information processing capabilities emerge in hierarchical systems of primitive computing units? What can be computed with these networks? How can these networks determine their structure in a self-organizing manner?

What is Emergence?

Is there a mathematical definition of emergence?
Maybe we can understand some of it by studying neural networks and their emergent properties.
More precisely:
What information processing capabilities emerge in hierarchical systems of primitive computing units? What can be computed with these networks? How can these networks determine their structure in a self-organizing manner?

Some more motivation

Understanding abstract neural networks has a twofold benefit:

- It contributes to our understanding of human brain and how high-level phenomena emerge from low-level phenomena.
- Provides algorithms to be used (pattern recognition, etc)

Locomotion simulation

Model of the Brain?

It seems to be a constant in the history of science that the brain has always been compared to the most complicated contemporary artefact produced by human industry. In ancient times the brain was compared to a pneumatic machine, in the Renaissance to a clockwork, and at the end of the last century to the telephone network. There are some today who consider computers the paradigm par excellence of a nervous system. It is rather paradoxical that when John von Neumann wrote his classical description of future universal computers, he tried to choose terms that would describe computers in terms of brains, not brains in terms of computers.

Model of the Brain?

It seems to be a constant in the history of science that the brain has always been compared to the most complicated contemporary artefact produced by human industry. In ancient times the brain was compared to a pneumatic machine, in the Renaissance to a clockwork, and at the end of the last century to the telephone network. There are some today who consider computers the paradigm par excellence of a nervous system. It is rather paradoxical that when John von Neumann wrote his classical description of future universal computers, he tried to choose terms that would describe computers in terms of brains, not brains in terms of computers.

Model of the Brain?

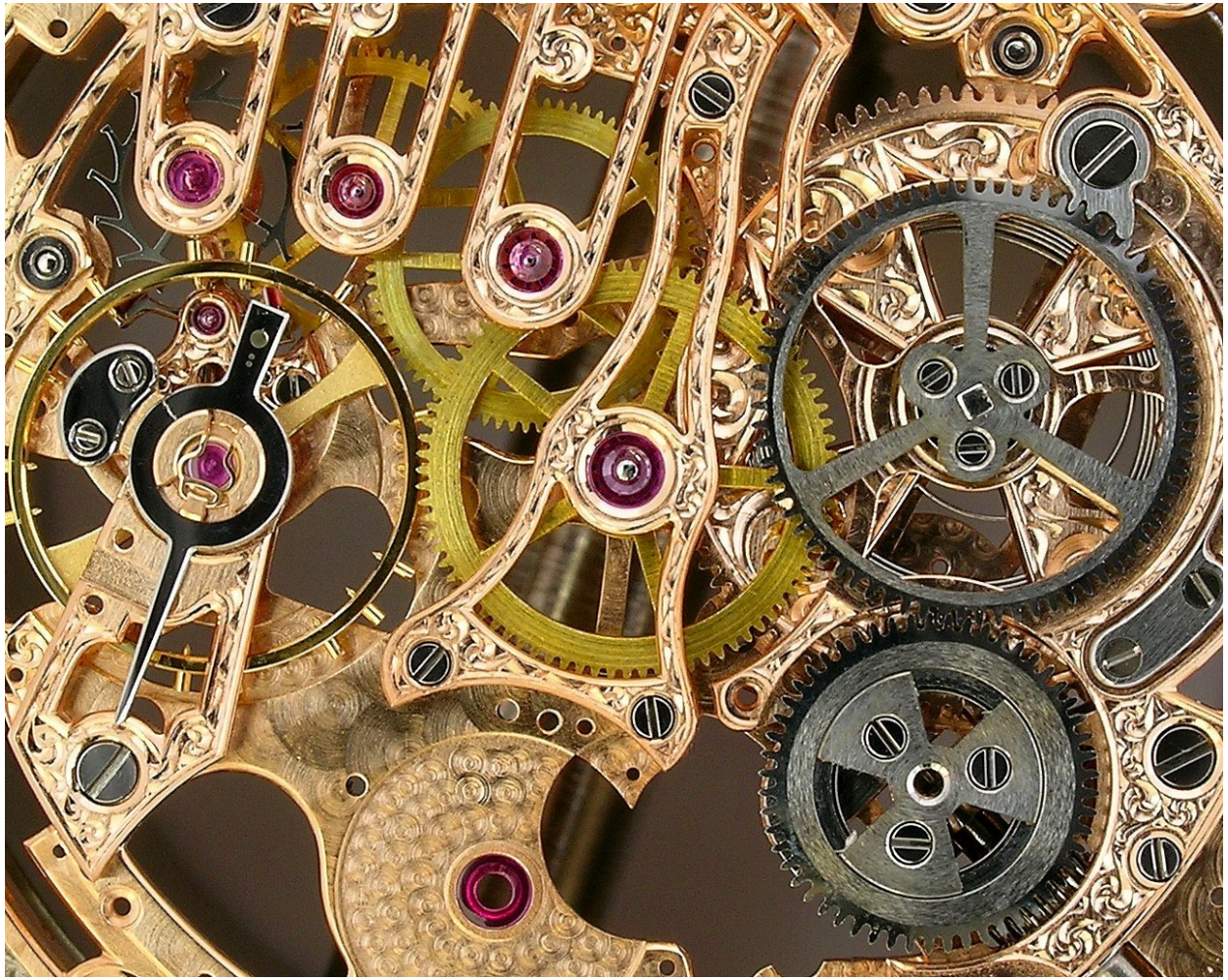
It seems to be a constant in the history of science that the brain has always been compared to the most complicated contemporary artefact produced by human industry. In ancient times the brain was compared to a pneumatic machine, in the Renaissance to a clockwork, and at the end of the last century to the telephone network. There are some today who consider computers the paradigm par excellence of a nervous system. It is rather paradoxical that when John von Neumann wrote his classical description of future universal computers, he tried to choose terms that would describe computers in terms of brains, not brains in terms of computers.

Model of the Brain?

It seems to be a constant in the history of science that the brain has always been compared to the most complicated contemporary artefact produced by human industry. In ancient times the brain was compared to a pneumatic machine, in the Renaissance to a clockwork, and at the end of the last century to the telephone network. There are some today who consider computers the paradigm par excellence of a nervous system. It is rather paradoxical that when John von Neumann wrote his classical description of future universal computers, he tried to choose terms that would describe computers in terms of brains, not brains in terms of computers.

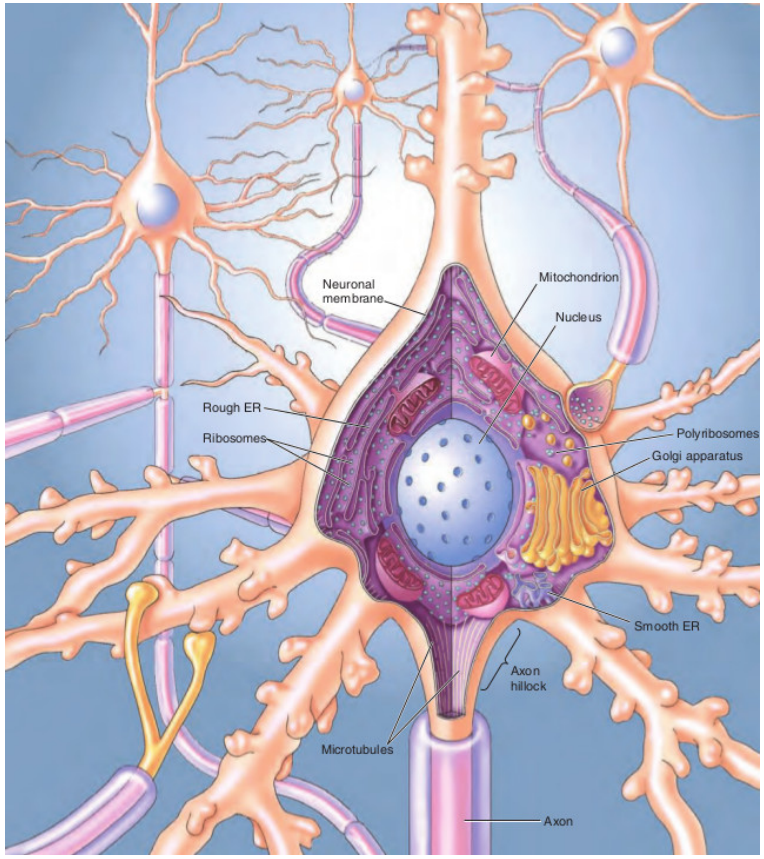
Model of the Brain?

Model of the Brain?



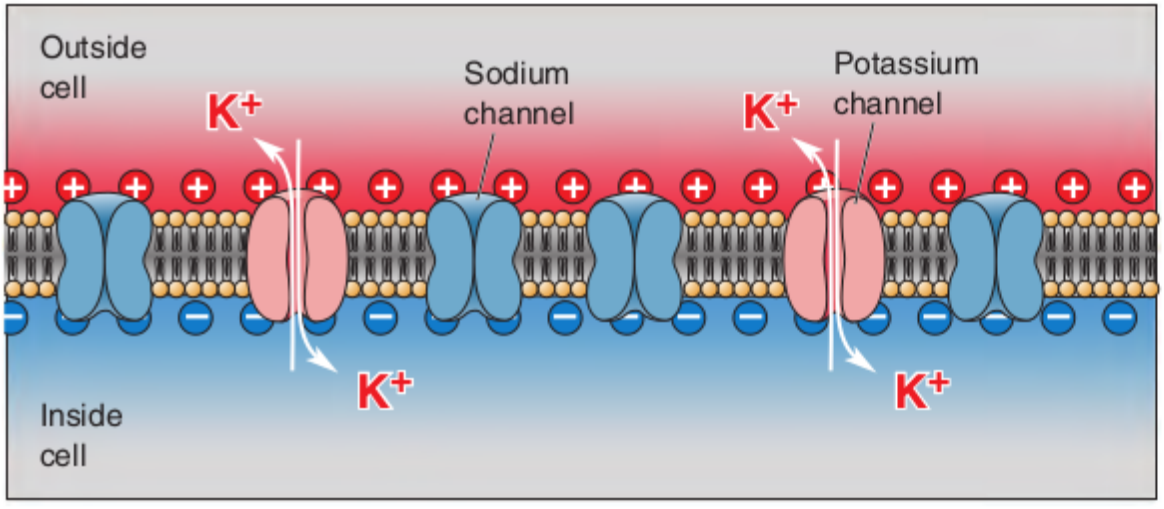
Neurons: Biological perspective

A neuron is a cell which has dendrites and one axon:

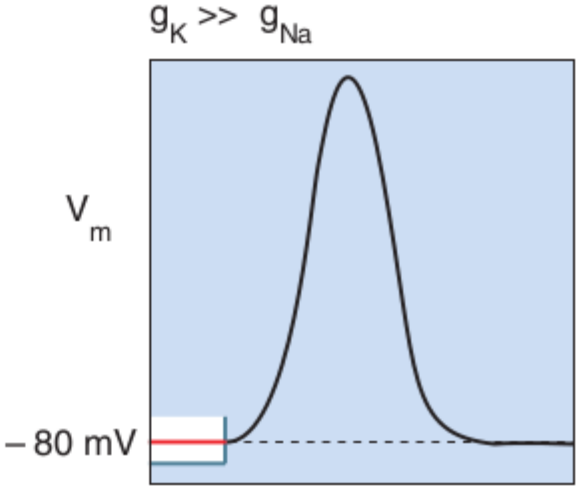


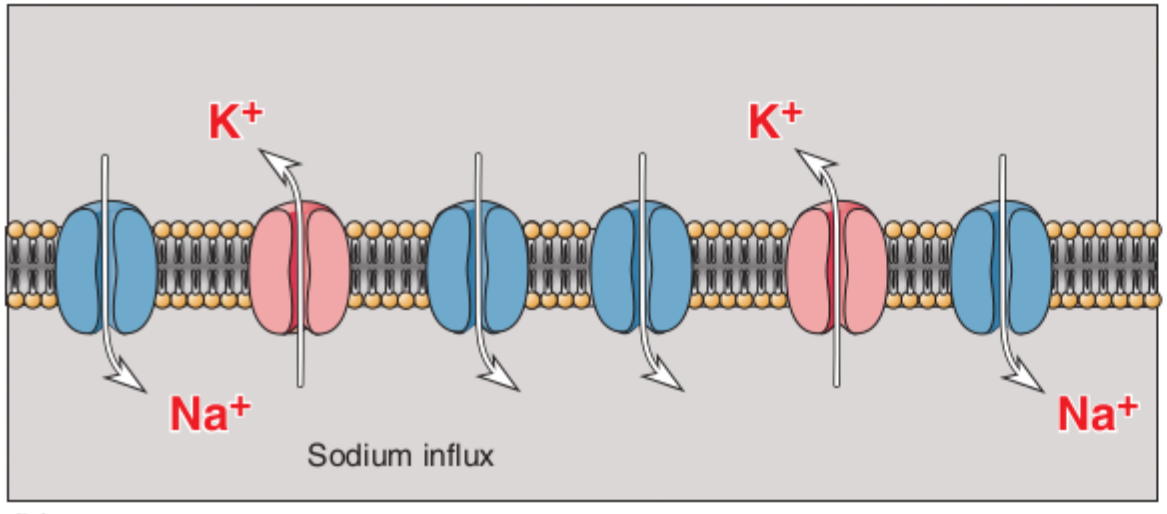
The axon often splits and connects via synapses to dendrites of other neurons.

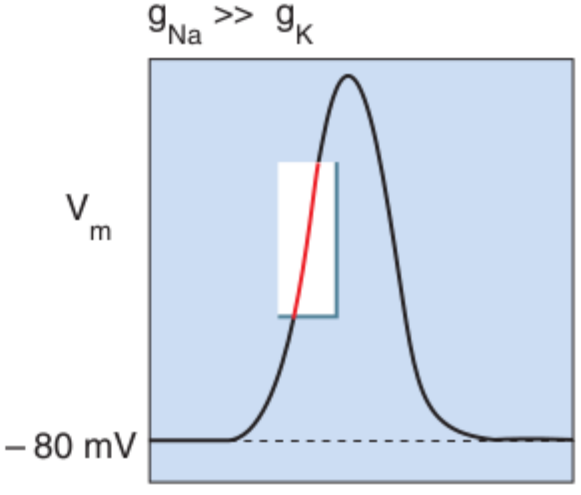
Neurons: Biological perspective



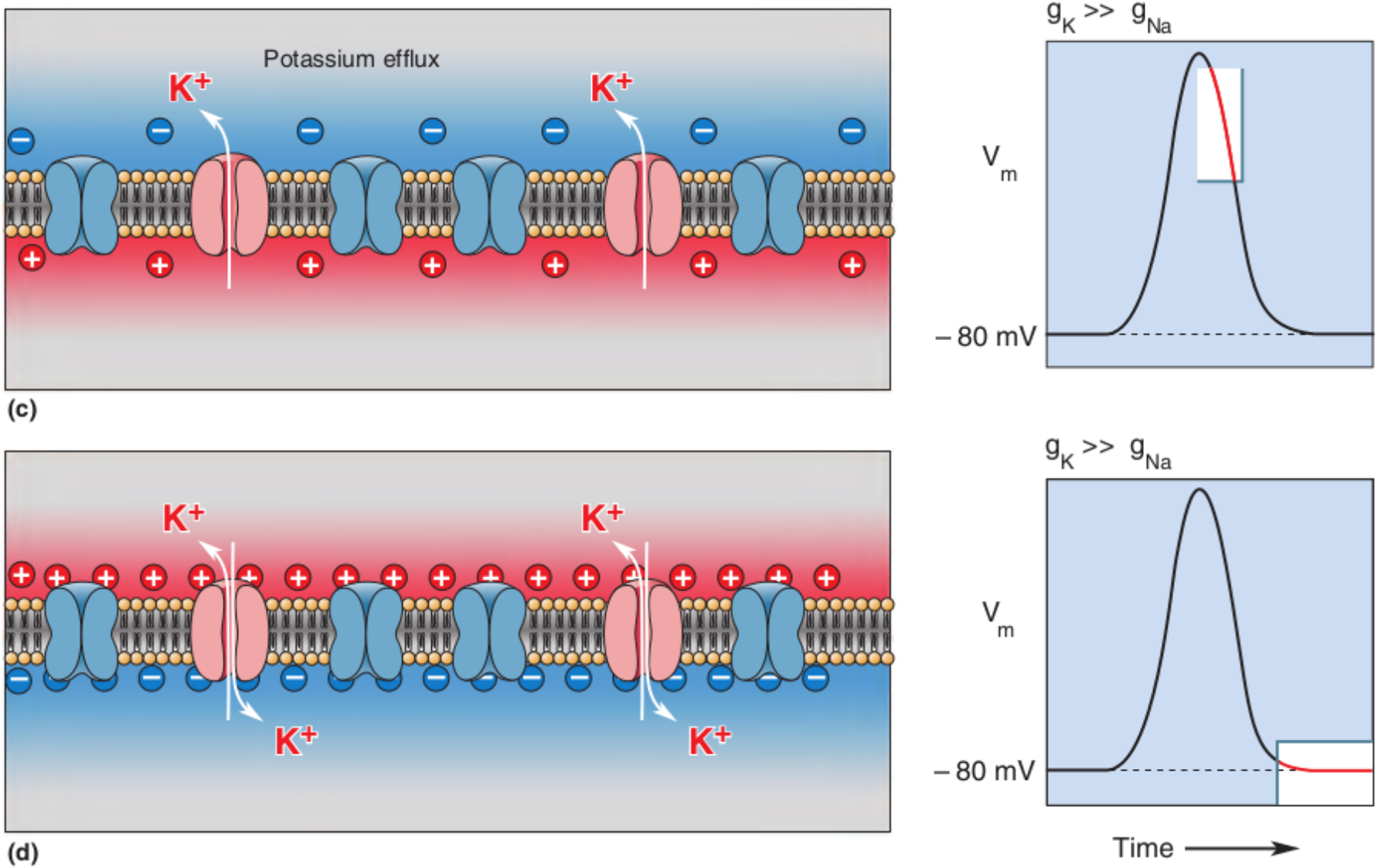
(a)







Neurons: Biological perspective



Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Neurons: Biological perspective

- The incoming signals from other synapses can depolarise or hyperpolarise the membrane.
- When the depolarisation of the membrane at the axon hillock reaches a certain threshold, action potential is generated.
- Refractory period prevents neuron from firing arbitrarily often
- All-or-none digital-type of information transmission. However if one looks at the *frequency* of firing instead, then a continuous model can be reasonable.
- Also sensory organs initiate action potentials.

Abstract neurons

In reality one neuron is quite complicated. The Human Brain Project acknowledges that simulating one neuron requires approximately the power of one laptop.

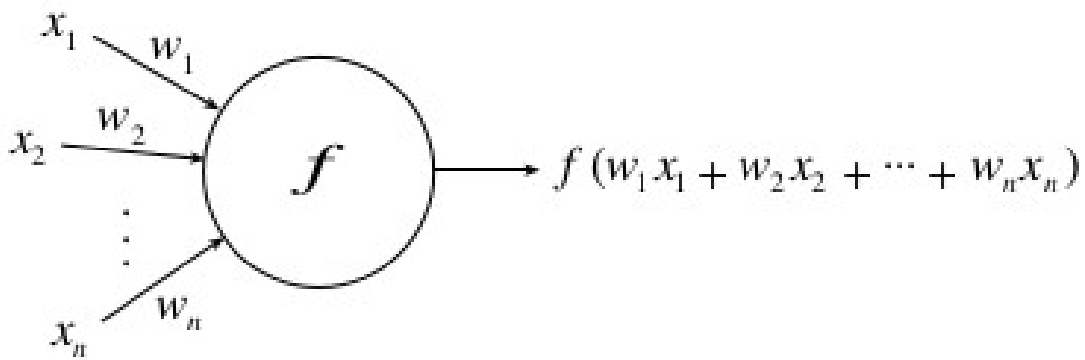
However for the purpose of mathematical analysis the neurons are often simplified:

Abstract neurons

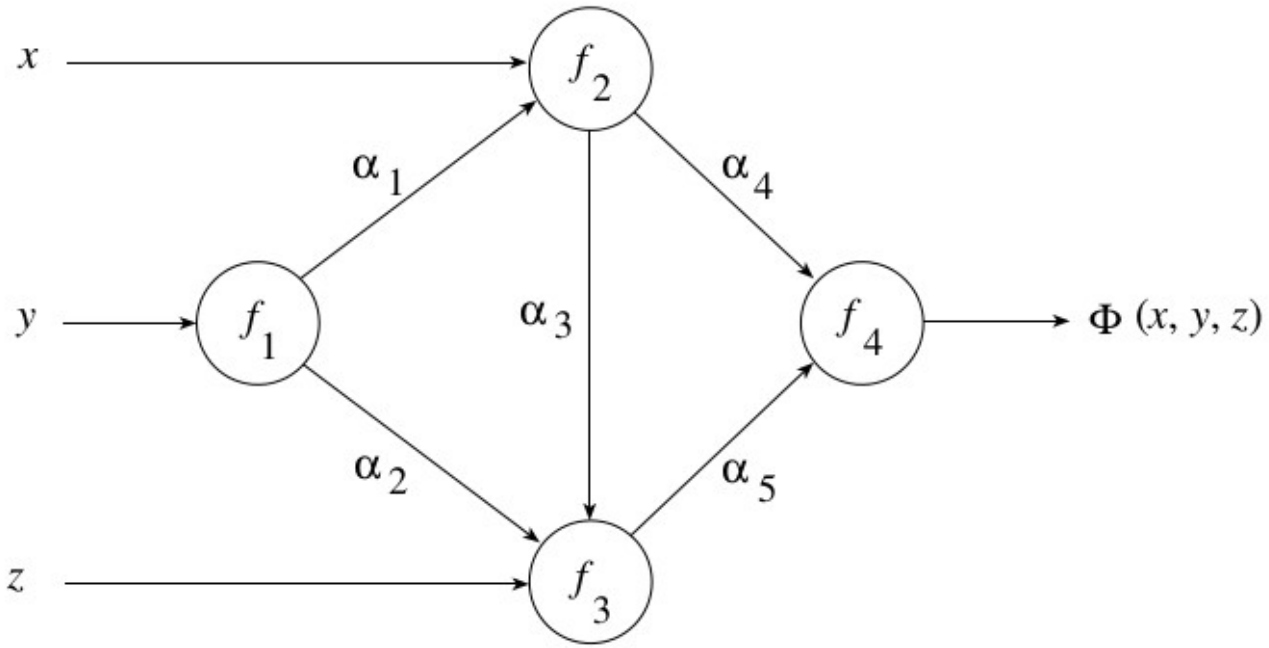
In reality one neuron is quite complicated. The Human Brain Project acknowledges that simulating one neuron requires approximately the power of one laptop.
However for the purpose of mathematical analysis the neurons are often simplified:

Abstract neurons

In reality one neuron is quite complicated. The Human Brain Project acknowledges that simulating one neuron requires approximately the power of one laptop.
However for the purpose of mathematical analysis the neurons are often simplified:



An Example of a Neuronetwork



Architectural differences: Neuronetworks vs. Turing machine

Artificial neural networks can be simulated with a computer. But *a priori* there is no reason why neuronetworks wouldn't be more efficient.

Architectural differences: Neuronetworks vs. Turing machine

The main difference between a Turing machine (a general model for a modern computer) and neuronetworks are:

- Neuronetworks admit massive parallelism,
- Redundancy in neuronetworks with which they deal with the unreliability of the individual computing units,
- Neuronetworks are (or can be) self-organising systems. Biologically, even individual neurons are complex self-organising systems.
- Neuronetworks are not handed "a program" but they adaptively *learn* to implement different tasks.

Architectural differences: Neuronetworks vs. Turing machine

The main difference between a Turing machine (a general model for a modern computer) and neuronetworks are:

- Neuronetworks admit massive parallelism,
- Redundancy in neuronetworks with which they deal with the unreliability of the individual computing units,
- Neuronetworks are (or can be) self-organising systems. Biologically, even individual neurons are complex self-organising systems.
- Neuronetworks are not handed "a program" but they adaptively *learn* to implement different tasks.

Architectural differences: Neuronetworks vs. Turing machine

The main difference between a Turing machine (a general model for a modern computer) and neuronetworks are:

- Neuronetworks admit massive parallelism,
- Redundancy in neuronetworks with which they deal with the unreliability of the individual computing units,
- Neuronetworks are (or can be) self-organising systems. Biologically, even individual neurons are complex self-organising systems.
- Neuronetworks are not handed "a program" but they adaptively *learn* to implement different tasks.

Architectural differences: Neuronetworks vs. Turing machine

The main difference between a Turing machine (a general model for a modern computer) and neuronetworks are:

- Neuronetworks admit massive parallelism,
- Redundancy in neuronetworks with which they deal with the unreliability of the individual computing units,
- Neuronetworks are (or can be) self-organising systems. Biologically, even individual neurons are complex self-organising systems.
- Neuronetworks are not handed "a program" but they adaptively *learn* to implement different tasks.

Architectural differences: Neuronetworks vs. Turing machine

The main difference between a Turing machine (a general model for a modern computer) and neuronetworks are:

- Neuronetworks admit massive parallelism,
- Redundancy in neuronetworks with which they deal with the unreliability of the individual computing units,
- Neuronetworks are (or can be) self-organising systems. Biologically, even individual neurons are complex self-organising systems.
- Neuronetworks are not handed "a program" but they adaptively *learn* to implement different tasks.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Neural Networks of this Course

- Simple perceptron learning,
- Associative learning and feedback networks,
- Hopfield networks,
- Feed-forward networks and backpropagation,
- Unsupervised learning.

Uninhibtory McCulloch-Pitts Units

The simplest model of a neuron is the uninhibitory McCulloch-Pitts unit:

Definition

Gets an input $\bar{x} \in \{0,1\}^n$ and outputs 1 if $x_1 + \cdots + x_n > \theta$ and otherwise 0.

[Geometric interpretations and logical functions]

Example: logical gates

Theorem (Chapter 2 Proposition 1)

Not all logical functions can be implemented with only uninhibitory McCulloch-Pitts units. (Only monotone ones can.)

General McCulloch-Pitts Units

Definition

General input: $(\bar{x}, \bar{y}) \in \{0,1\}^n \times \{0,1\}^m$ where y are *inhibitory* type inputs and x are *uninhibitory* type. The output is 1 if and only if $\bar{y} = \bar{0}$ and $x_1 + \cdots + x_n > \theta$ and can be either of inhibitory or uninhibory type (depending on the unit).

Logical gates

Theorem (Chapter 2 Proposition 2)

All logical functions can be implemented using also inhibitory connections.

Perceptron

Perceptron is a generalisation of the McCulloch-Pitts units. It has n inputs $(x_1, \ldots, x_n)$ and a weight w_i associated with each input channel i. The output is

$$f_\theta(w_1 x_1 + \cdots + w_n x_n)$$

where f_θ is the step function $f_\theta(x) = 1$ if $x \geqslant \theta$ and $f_\theta(x) = 0$ if $x < \theta$.

Perceptron

Exercise

A McCulloch-Pitts unit can be simulated by a perceptron.

Theorem (Proposition 4, Chapter 2)

Networks of McCulloch-Pitts units are equivalent to networks with relative inhibition.

Exercise

A perceptron can be "simulated" (replaced) by a network of McCulloch-Pitts units (with inhibitory connections allowed) that separates the same sets.

Example: linear separation

Definition

Two sets $A, B \subset \mathbb{R}^n$ are (absolutely) linearly separably if there exists $\bar{w} \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ such that for all $\bar{a} \in A$

$$\bar{w} \cdot \bar{a} > \theta$$

all $\bar{b} \in B$

$$\bar{w} \cdot \bar{b} < \theta$$

Perceptron is a linear separator

Perceptron outputs

$$f_\theta(w_1 x_1 + \cdots + w_n x_n)$$

which is equivalent to saying that it outputs 1 if

$$\bar{w} \cdot \bar{x} > \theta$$

Thus perceptron divides the space into two half-spaces. This makes it possible to make a classifier perceptron: if A and B are subsets of $\mathbb{R}^n$ that are linearly separable, then there are weights $\bar{w} = (w_1, \ldots, w_n)$ and a threshold θ such that the perceptron with corresponding values separates A from B.
Conversely, if A and B are not linearly separable, then they cannot be separated by a perceptron.

Example: linear separation

- One perceptron acts as a linear separator,
- XOR-function cannot be implemented with a single perceptron,
- Every logical function can be implemented with a two-layer network of perceptrons.

Example: linear separation

- One perceptron acts as a linear separator,
- XOR-function cannot be implemented with a single perceptron,
- Every logical function can be implemented with a two-layer network of perceptrons.

Example: linear separation

- One perceptron acts as a linear separator,
- XOR-function cannot be implemented with a single perceptron,
- Every logical function can be implemented with a two-layer network of perceptrons.

Examples

Another example: edge detection. Using multiple layers amounts to this: first layer detects local features like edges, horizontal, vertical lines, contrasts etc. Next layer finds less local features combined from these etc.

Perceptron learning

start: The weight vector $\mathbf{w}_0$ is generated randomly, set $t := 0$

test: A vector $\mathbf{x} \in P \cup N$ is selected randomly,

if $\mathbf{x} \in P$ and $\mathbf{w}_t \cdot \mathbf{x} > 0$ go to *test*,
if $\mathbf{x} \in P$ and $\mathbf{w}_t \cdot \mathbf{x} \leq 0$ go to *add*,
if $\mathbf{x} \in N$ and $\mathbf{w}_t \cdot \mathbf{x} < 0$ go to *test*,
if $\mathbf{x} \in N$ and $\mathbf{w}_t \cdot \mathbf{x} \geq 0$ go to *subtract*.

add: set $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{x}$ and $t := t + 1$, goto *test*

subtract: set $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{x}$ and $t := t + 1$, goto *test*

Perceptron learning

Theorem (Chapter 4, Proposition 8)

If N and P are finite and linearly separable, then the perceptron learning algorithm finds a solution in a finite number of steps.

Curious facts

Curious Fact

Nobody knows a formula for the number of linearly separable partitions of $\{0,1\}^n$. The value is known only up to $n = 9$.

Curious Fact

Finding the right weights (i.e. learning) in a neural network is an NP-complete problem.

Curious facts

Curious Fact

Nobody knows a formula for the number of linearly separable partitions of $\{0,1\}^n$. The value is known only up to $n = 9$.

Curious Fact

Finding the right weights (i.e. learning) in a neural network is an NP-complete problem.

Perceptron learning

Exercise

Show that the parity function of $n > 2$ bits cannot be computed by a single perceptron. Hint: Without loss of generality you may assume that the neuron fires when input contains an odd number of ones and that the threshold is positive.

Exercise

Let $n > 2$. Construct a neural network consisting of perceptrons which computes the parity function. I.e. it takes a vector $\bar{x} \in \{0, 1\}^n$ as an input and outputs 1 if the number of 1's in x is odd and otherwise outputs 0.