SRP
Neural Networks

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Goals

Goals of the meeting:

- Give an interdisciplinary overview of artificial neural networks
- Present an application in machine learning
- Discuss the mathemtics behind the application

Outline

1. Neuroscience

2. Artificial Neural Networks
   - Feedforward Networks
     - Single-layer perceptrons
     - Multi-layer perceptrons
   - Recurrent Networks

3. Other Applications
   - Simulations

Mind

What is the mind?

- Neither scientists nor philosophers agree on a universal definition or specification.
- Colloquially, we understand the mind as a collection of processes of cognitive functions.
- Cognitive functions: perception, attention, learning and memory, emotions, symbolic representation, decision-making, reasoning, problem solving and consciousness.
  Human conscious mentation as opposed to unconscious processes
  (cognition = the faculty of knowing)
- The mind can integrate ambiguous information from sight, hearing, touch, taste, and smell; it can form spatio-temporal associations and abstract concepts; it can make decisions and initiate sophisticated coordinated actions.

Philosophy

Descartes' (1596-1650) dualism theory:

	Mind	Body (Brain)
essence	Thinking (consciousness) (res cogitans)	physical extension (having spatial dimensions) (res extensa)

⇝ Mind-Body problem:

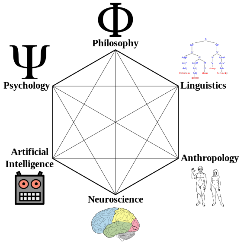
how can there be causal relationship between two completely different metaphysical realms?

Since then **Philosophy of Mind** is an active research field: functionalism, computationalism, connectivism, qualia, intentionality, non-reductivism

[Mind: A Brief Introduction, John Searle]

Cognitive Psychology

- How people acquire, process and apply knowledge or information.
  Closely related to interdisciplinary cognitive science (thinking understood in terms of representational structures and computational procedures that operate on them)



- Arose in the mid 1950s in opposition to behaviourism: objective, observable stimulus conditions determine by laws related observable behavior. No recourse to internal mental processes is assumed.
  No distinction between memory and performance and failed to account for complex learning.

- Research tools: experimental, cognitive computing, and neural cognitive psychology (brain imaging techniques and neurobiological methods, e.g., lesion patients)

Biology

Neuroscience (aka, neural science)

- is concerned with how the biological nervous systems of humans and other animals are organized and how they function
- The goal of neuroscience is to understand the anatomical, physiological and molecular bases of mental processes.
  How does the brain produce the remarkable individuality of human action?
  Are mental processes localized to specific regions of the brain, or do they represent emergent properties of the brain as an organ?
  The ultimate goal is to explain higher order brain functions such as cognitive functions (cognitive neuroscience)
- The specificity of the synaptic connections established during development underlie cognitive functions.
  It aims also at understanding both the innate (genetic) and environmental determinants of behavior.

Neuroscience

The physiological basis (= how cells and bio-molecules carry out chemical and physical function) of the elements of the brain are used to explain higher order functions of human brains.

The scientific methods include measuring firing rates and membrane potentials:

- electroencephalography (EEG)
  records electrical voltages from electrodes placed on the scalp,
  high temporal resolution (milliseconds), low spatial (centimeters)
- positron emission tomography
- functional magnetic resonance imaging (fMRI)
  activity causes variations in blookd oxigenation ⇒ magnetization,
  low temporal resolution (hundreds of ms), high spatial (mm)
- magneto-encephalography (MEG)
  records the magnetic field from SQUID sensors placed above the head
- trans-cranial magnetic stimulation

Cognitive Computing

- Strong artificial general intelligence
- system-level approach to synthesizing mind-like computers. (top-down, reductionism)
  Neuroscience takes a component-level approach to understanding how the mind arises from the wetware of the brain (bottom-up).
- Cognitive computing aims to develop a coherent, unified, universal mechanism inspired by the mind's capabilities.
  Rather than assemble a collection of piecemeal solutions, whereby different cognitive processes are each constructed via independent solutions, we seek to implement a unified computational theory of the mind.
- Symbols and reasonining, logic and search

Computational Neuroscience

Simulation from neuroscience data. Neurobiological data provide essential constraints on computational theories
$\rightsquigarrow$ narrowing the search space.

Goal: discover, demonstrate, and deliver the core algorithms of the brain and gain a deep scientific understanding of how the mind perceives, thinks, and acts.

Ultimately, this will lead to novel cognitive systems, computing architectures, programming paradigms, practical applications, and intelligent business machines.

Observations from computational neuroscience and cognitive computing:

- Neuroscientists: view them as a web of clues to the biological mechanisms of cognition. Testable hypothesis.
- Engineers: The brain is an example solution to the problem of cognitive computing

Neurophysiology

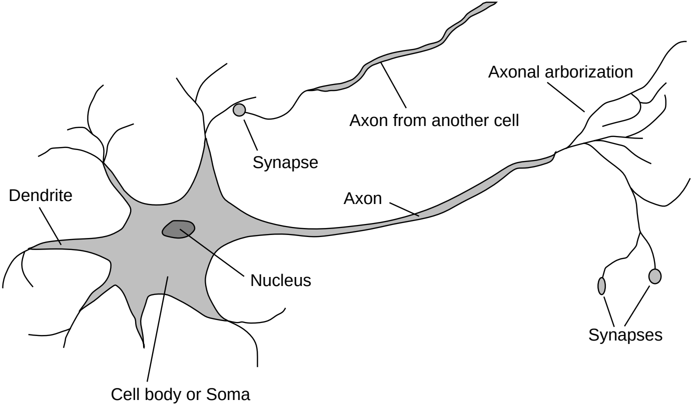
Neuron: adaptation of a biological cell into a structure capable of:

- receiving and integrating **input**,
- making a decision based on that input, and
- signaling other cells depending on the outcome of that decision

Three main structural components:

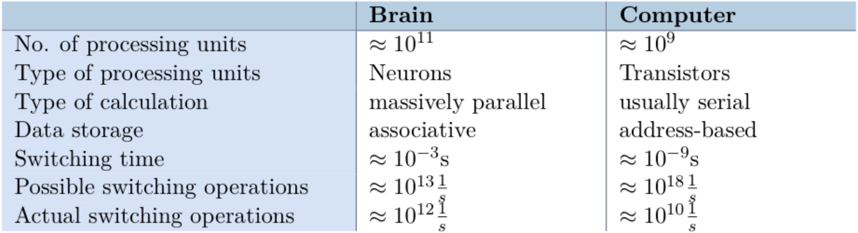
- dendrites, tree-like structures that receive and integrate inputs;
- a soma, where decisions based on these inputs are made;
- and an axon, a long narrow structure that transmits signals to other neurons near and far (can reach one meter length)

A neuron in a living biological system



Signals are noisy “spike trains” of electrical potential

In the brain: > 20 types of neurons with 10^{14} synapses



	Brain	**Computer**
No. of processing units	$\approx 10^{11}$	$\approx 10^{9}$
Type of processing units	Neurons	Transistors
Type of calculation	massively parallel	usually serial
Data storage	associative	address-based
Switching time	$\approx 10^{-3}$s	$\approx 10^{-9}$s
Possible switching operations	$\approx 10^{13} \frac{1}{s}$	$\approx 10^{18} \frac{1}{s}$
Actual switching operations	$\approx 10^{12} \frac{1}{s}$	$\approx 10^{10} \frac{1}{s}$

(compare with world population $= 7 \times 10^{9}$)
Additionally, brain is parallel and reorganizing while computers are serial and static
Brain is fault tolerant: neurons can be destroyed.

Signal integration and transmission within a neuron:

- Fluctuations in the neuron's membrane potential: voltage difference across the membrane that separates the interior and exterior of a cell.
- Fluctuations occur when ions cross the neuron's membrane through channels that can be opened and closed selectively.
- If the membrane potential crosses a critical threshold, the neuron generates a spike (its determination that it has received noteworthy input), which is a reliable, stereotyped electrochemical signal sent along its axon.
- Spikes are the essential information couriers of the brain
  e.g., used in the sensory signals the retina sends down the optic nerve in response to light; in the control signals the motor cortex sends down the spinal cord to actuate muscles, and in virtually every step in between.

- Synapses are tiny structures that bridge the axon of one neuron to the dendrite of the next, transducing the electrical signal of a spike into a chemical signal and back to electrical.

- The spiking neuron, called the presynaptic neuron, releases chemicals called neurotransmitters at the synapse that rapidly travel to the other neuron, called the postsynaptic neuron.

- The neurotransmitters trigger ion-channel openings on the surface of the post-synaptic cell, subsequently modifying the membrane potential of the receiving dendrite.

- These changes can be either excitatory, meaning they make target neurons more likely to fire, or inhibitory, making their targets less likely to fire.

- Both the input spike pattern received and the neuron type determine the final spiking pattern of the receiving neuron.

Thus:

- essentially digital electrical signal of the spike sent down one neuron
- is converted first into a chemical signal that can travel between neurons
- then into an analog electrical signal that can be integrated by the receiving neuron.

Intelligence is in the network was the initial credo in the early years. Other newer parameters have entered electrophysiology: **Plasticity** (mostly as Long Term Potentiation and Long Term Depression) and intrinsic electrical properties.

The magnitude of this analog post-synaptic activation, called synaptic strength, is not fixed over an organism's lifetime.

Widely believed among brain researchers that changes in synaptic strength underlie learning and memory, and hence that understanding synaptic plasticity could provide crucial insight into cognitive function.

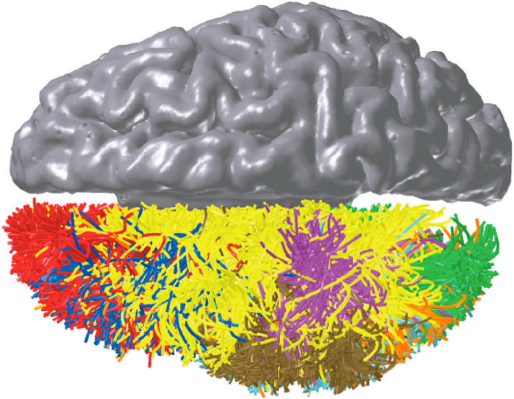
Donald O. Hebb's famous conjecture for synaptic plasticity is "neurons that fire together, wire together," i.e., that if neuron A and B commonly fire spikes at around the same time, they will increase the synaptic strength between them.

How much details of such a spiking message passing, like time dynamics of dendritic compartments, ion concentrations, and protein conformations, are relevant to the fundamental principles of cognition?

Neuroanatomy

At the surface of the brains of all mammals is a sheet of tissue a few millimeters thick called the cerebral cortex

Neurons are connected locally through gray-matter connections, as well as through long-range white-matter connections



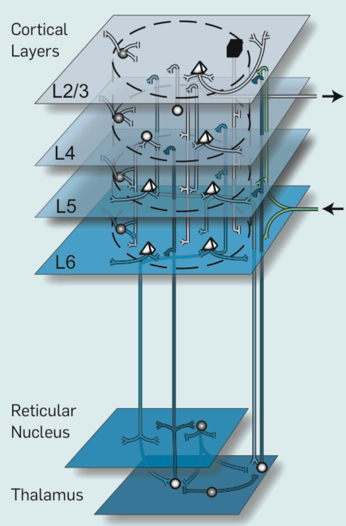
diffusion-weighted magnetic resonance imaging (Dw-Mri)

functional magnetic resonance imaging (fMRI)

Structure within cortex: six distinct horizontal layers spanning the thickness of the cortical sheet. interlaminar activity propagation

Cortical columns organize into cortical areas that are often several millimeters across and appear to be responsible for specific functions, including motor control, vision, and planning.

Scientists have focused on understanding the role each cortical area plays in brain function and how anatomy and connectivity of the area serve that function.



Structural plasticity For example, it has been demonstrated that an area normally specialized for audition can function as one specialized for vision, and vice versa, by rewiring the visual pathways in the white matter to auditory cortex and the auditory pathways to visual cortex

The existence of a canonical algorithm is a prominent hypothesis

At the coarsest scale of neuronal system organization, multiple cortical areas form networks to address complex functionality.

From Brains to Artificial Neural Networks

From neuroscience observations to artificial neurons

- The brain's neuronal network is a sparse, directed graph organized at multiple scales.
- Local, short-range connections can be described through statistical variations on a repeating canonical subcircuit,
- Global, long-range connections can be described through a specific, low-complexity blueprint.
- Repeating structure within an individual brain and a great deal of homology across species.

Thesis: computational building blocks of the brain (neurons and synapses) can be **described** by relatively compact, functional, phenomenological mathematical models, and their communication can be summarized in binary, asynchronous messages (spikes).

Key idea: behavior of the brain apparently emerges via non-random, correlated interactions between individual functional units, a key characteristic of organized complexity.

Such complex systems are often more amenable to computer modeling and simulation than to closed-form analysis and often resist piecemeal decomposition.

Outline

1. Neuroscience

2. Artificial Neural Networks
   - Feedforward Networks
     - Single-layer perceptrons
     - Multi-layer perceptrons
   - Recurrent Networks

3. Other Applications
   - Simulations

How to teach computers to carry out difficult tasks?
Get inspired from Biology and let computers learn themselves like children.
[A.M. Turing. Computing Machinery and Intelligence. Mind, Oxford University Press on behalf of the Mind Association, 1950, 59(236), 433-460]

Learning:

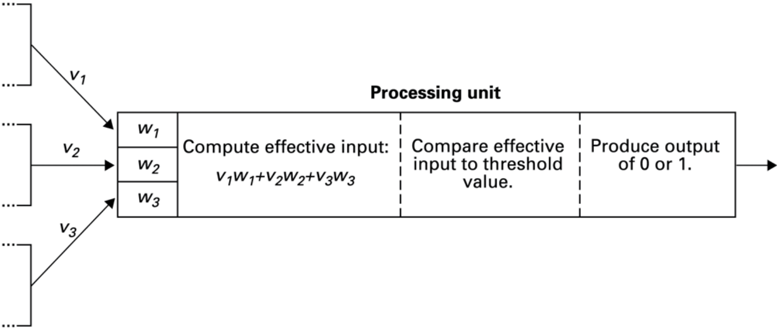
- Supervised Training (Imitation)
- Reinforcement
- Unsupervised

Artificial Neural Networks

$\rightsquigarrow$ "**The** neural network" does not exist. There are different paradigms for neural networks, how they are trained and where they are used.

- Artificial Neuron
  - Each input is multiplied by a weighting factor.
  - Output is 1 if sum of weighted inputs exceeds the threshold value; 0 otherwise.
- Network is programmed by adjusting weights using feedback from examples.

Activities within a processing unit

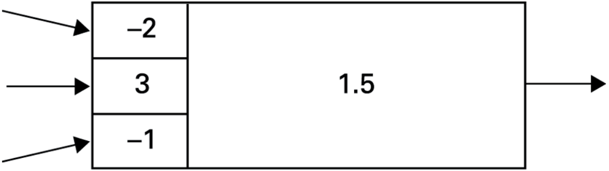


Processing unit

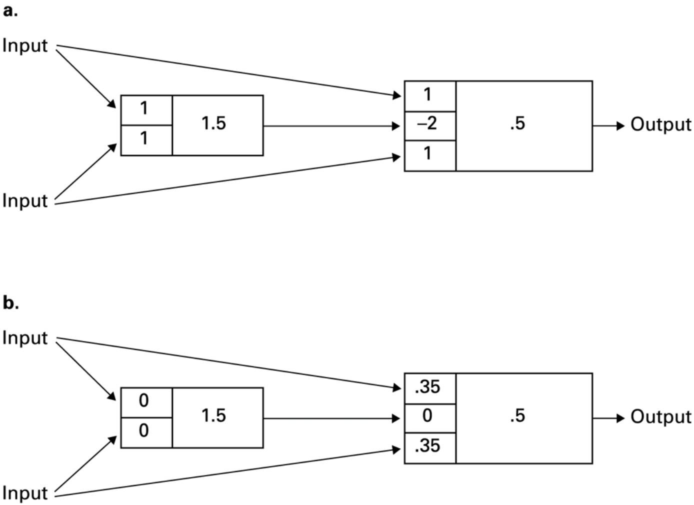
Compute effective input: $v_1w_1+v_2w_2+v_3w_3$

Compare effective input to threshold value.

Produce output of 0 or 1.



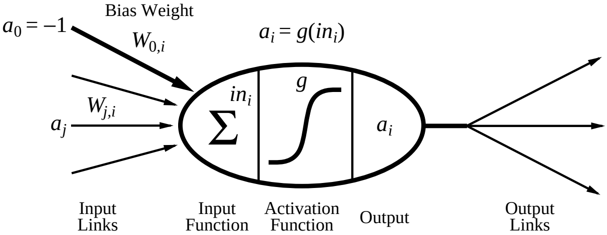
Neural Network with two layers



McCulloch–Pitts "unit" (1943)

Output is a function of weighted inputs:

$$a_i = g(in_i) = g\left(\sum_j W_{j,i} a_j\right)$$

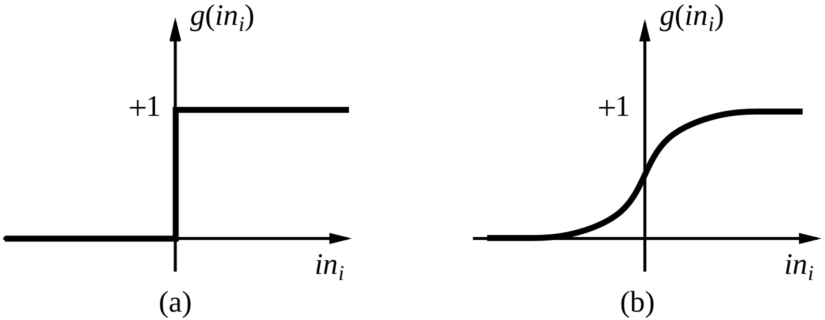


Changing the bias weight $W_{0,i}$ moves the threshold location

A gross oversimplification of real neurons, but its purpose is
to develop understanding of what networks of simple units can do

Activation functions

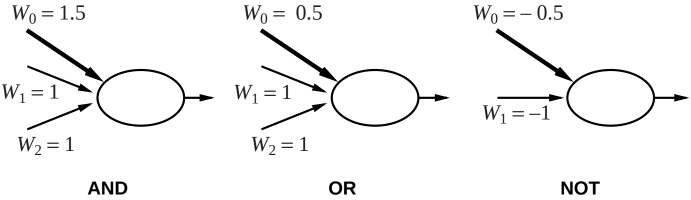
Non linear activation functions



(a) is a step function or threshold function
(mostly used in theoretical studies)

(b) is a continuous activation function, e.g., sigmoid function $1/(1+e^{-x})$
(mostly used in practical applications)

Implementing logical functions



McCulloch and Pitts: every Boolean function can be implemented

Network structures

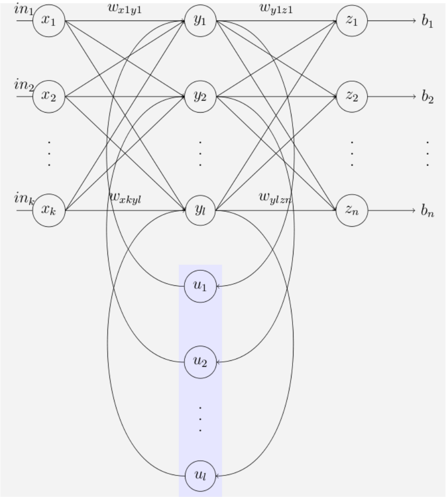
Architecture: definition of number of nodes and interconnection structures and activation functions σ but not weights.

- Feed-forward networks:
  no cycles in the connection graph
  - single-layer perceptrons (no hidden layer)
  - multi-layer perceptrons (one or more hidden layer)

  Feed-forward networks implement functions, have no internal state

- Recurrent networks:
  connections between units form a directed cycle.
  – internal state of the network
  exhibit dynamic temporal behavior (memory, apriori knoweldge)
  – Hopfield networks have symmetric weights ($W_{i,j} = W_{j,i}$)
  $\sigma(x) = \text{sign}(x)$, $a_i = \{1, 0\}$; **associative memory**

Elmann's Simple Recurrent Network

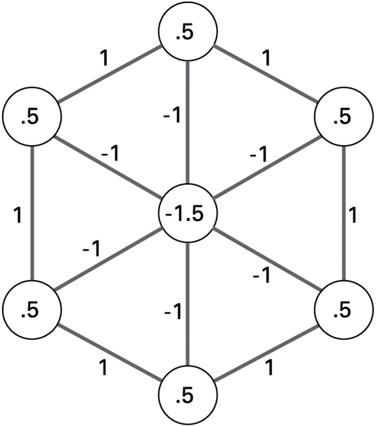


Good for sequence prediction

Hopfield Networks

An artificial neural network implementing an associative memory

- symmetric weights ($W_{i,j} = W_{j,i}$);
    - $g(x) = \text{sign}(x)$, $a_i = \{1, 0\}$;
- operates in synchronized discrete steps



Feed-Forward Networks – Use

Neural Networks are used in classification and regression

- Boolean classification:
  - value over 0.5 one class
  - value below 0.5 other class
- k-way classification
  - divide single output into k portions
  - k separate output unit
- continuous output
  - identity activation function in output unit

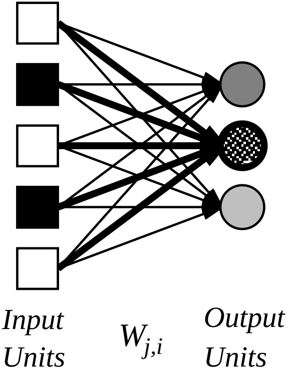
Outline

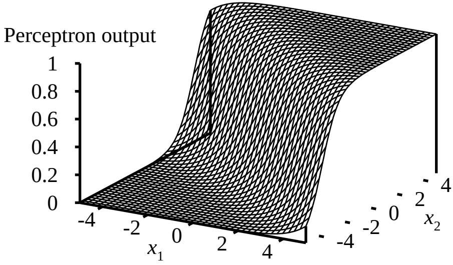
1. Neuroscience

2. Artificial Neural Networks
   - Feedforward Networks
     - Single-layer perceptrons
     - Multi-layer perceptrons
   - Recurrent Networks

3. Other Applications
   - Simulations

Single-layer NN (perceptrons)





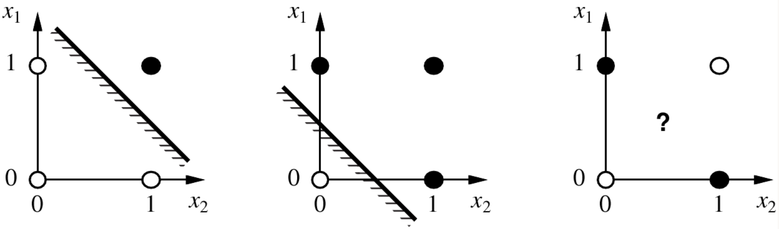
Output units all operate separately—no shared weights
Adjusting weights moves the location, orientation, and steepness of cliff

Expressiveness of perceptrons

Consider a perceptron with σ = step function (Rosenblatt, 1957, 1960)
The output is 1 when:

$$\sum_j W_j x_j > 0 \quad \text{or} \quad \mathbf{W} \cdot \mathbf{x} > 0$$

Hence, it represents a linear separator in input space:
- hyperplane in multidimensional space
- line in 2 dimensions



Minsky & Papert (1969) pricked the neural network balloon

Perceptron learning

Learn by adjusting weights to reduce error on training set
The squared error for an example with input $\mathbf{x}$ and true output y is

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_{\mathbf{W}}(\mathbf{x}))^2 \ ,$$

Find local optima for the minimization of the function $E(\mathbf{W})$ in the vector of variables $\mathbf{W}$ by gradient methods.

Note, the function E depends on constant values $\mathbf{x}$ that are the inputs to the perceptron.

The function E depends on h which is non-convex, hence the optimization problem cannot be solved just by solving $\nabla E(\mathbf{W}) = 0$

Digression: Gradient methods

Gradient methods are iterative approaches:

- find a descent direction with respect to the objective function E
- move $\mathbf{W}$ in that direction by a step size

The descent direction can be computed by various methods, such as gradient descent, Newton-Raphson method and others. The step size can be computed either exactly or loosely by solving a line search problem.

Example: gradient descent

```
1. Set iteration counter $t = 0$, and make an initial guess $\mathbf{W}_0$ for the minimum
2. Repeat:
3.     Compute a descent direction $\mathbf{p}_t = \nabla(E(\mathbf{W}_t))$
4.     Choose $\alpha_t$ to minimize $f(\alpha) = E(\mathbf{W}_t - \alpha \mathbf{p}_t)$ over $\alpha \in \mathbb{R}_+$
5.     Update $\mathbf{W}_{t+1} = \mathbf{W}_t - \alpha_t \mathbf{p}_t$, and $t = t + 1$
6. Until $\|\nabla f(\mathbf{W}_k)\| < tolerance$
```

Step 4 can be solved 'loosely' by taking a fixed small enough value $\alpha > 0$

Perceptron learning

In the specific case of the perceptron, the descent direction is computed by the gradient:

$$\begin{aligned}\frac{\partial E}{\partial W_j} &= Err \cdot \frac{\partial Err}{\partial W_j} = Err \cdot \frac{\partial}{\partial W_j}\left(y - \sigma(\sum_{j=0}^{n} W_j x_j)\right) \\ &= -Err \cdot \sigma'(in) \cdot x_j\end{aligned}$$

and the weight update rule (perceptron learning rule) in step 5 becomes:

$$W_j^{t+1} = W_j^t + \alpha \cdot Err \cdot \sigma'(in) \cdot x_j$$

For threshold perceptron, $\sigma'(in)$ is undefined: Original perceptron learning rule (Rosenblatt, 1957) simply omits $\sigma'(in)$

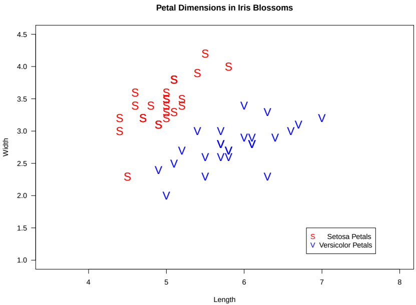
Perceptron learning contd.

```
Perceptron-Learning(examples,network)

input  : examples: a set of examples, each with input $\mathbf{x}^e = x_1^e, x_2^e, \ldots, x_n^e$ and
         output $y^e$
         network: a perceptron with weights $W_j, j = 0..n$ and activation function $g$
output : a network

repeat
    for $e$ in $examples$ do
        $in \leftarrow \sum_{j=0}^{n} W_j x_j[e]$
        $Err \leftarrow y[e] - g(in)$
        $W_j \leftarrow W_j + \alpha \cdot Err \cdot g'(in) \cdot x_j[e]$
until all examples correctly predicted or stopping criterion is reached ;
```

Perceptron learning rule converges to a consistent function
for any linearly separable data set

Numerical Example

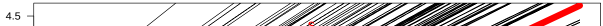
The (Fisher's or Anderson's) iris data set gives measurements in centimeters of the variables: sepal length and petal lenght and petal width for 50 flowers from 2 species of iris: "Iris setosa", and "versicolor".



```
> head(iris.data)

   Sepal.Length Sepal.Width    Species id
10          4.9         3.1     setosa -1
91          5.5         2.6 versicolor  1
85          5.4         3.0 versicolor  1
86          6.0         3.4 versicolor  1
29          5.2         3.4     setosa -1
83          5.8         2.7 versicolor  1
```

```
> ##iris.data <- D
> #D$V4 <- D$V3
> #plot(D[,1],D[,2])
> sigma <- function(w,point)
+   {
+     x <- c(point,1)
+     sign(w %*% x)
+   }
> w.0 <- c(runif(1),runif(1),runif(1))
> #w.0 <- c(1,-1,-1)
> #w.0 <- c(0.75,0.5,-0.6)
> w.t <- w.0
> for (j in 1:1000)
+   {
+     i <- (j-1)%%50 + 1
+     diff <- iris.data[i,4] - sigma(w.t, c(iris.data[i,1],iris.data[i,2]
+     w.t <- w.t+0.2*diff * c(iris.data[i,1],iris.data[i,2],1)
+     #cat(i," ",j," ",diff,w.t,"\n")
+ ##    matlines(x <- seq(0,10,0.1),(-w.t[1]*x-w.t[3])/w.t[2])
+ #    matlines(x <- seq(3.5,8,0.1),(-w.t[1]*x-w.t[3])/w.t[2])
+   }
> #
```

Petal Dimensions in Iris Blossoms



In Maple

Using Linear algebra to build the Perceptron

```
> with(linalg):
> x1 := vector([0.3,0.7, -1]);
> y := 1;
> w0 := vector([-0.6,0.8, 0.6]);
> i1 := dotprod(a1,w0);
> g := signum(i1);
> diff := y-1
> w1 := w0 + 0.2 * diff * x1
```

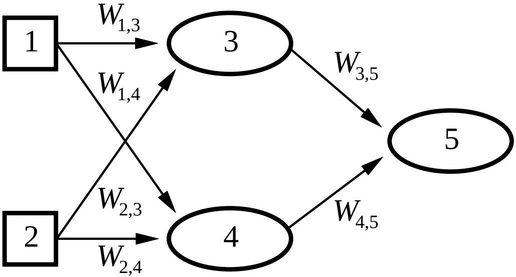
Outline

1. Neuroscience

2. Artificial Neural Networks
   - Feedforward Networks
     - Single-layer perceptrons
     - Multi-layer perceptrons
   - Recurrent Networks

3. Other Applications
   - Simulations

Multilayer Feed-forward

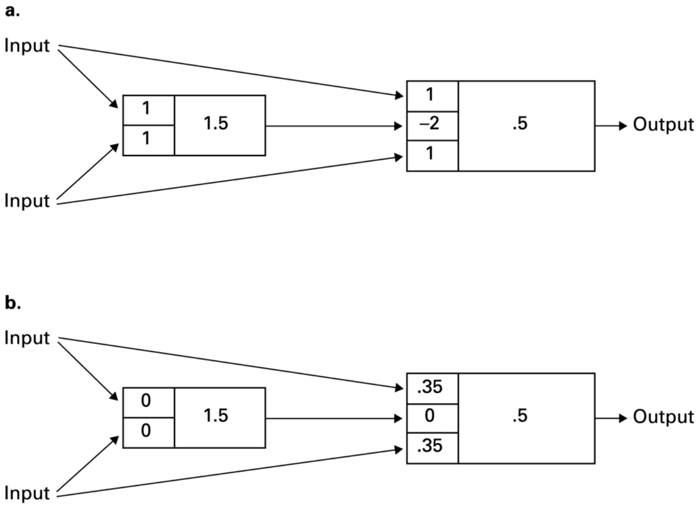


Feed-forward network = a parametrized family of nonlinear functions:

$$\begin{aligned} a_5 &= \sigma(W_{3,5} \cdot a_3 + W_{4,5} \cdot a_4) \\ &= \sigma(W_{3,5} \cdot \sigma(W_{1,3} \cdot a_1 + W_{2,3} \cdot a_2) + W_{4,5} \cdot \sigma(W_{1,4} \cdot a_1 + W_{2,4} \cdot a_2)) \end{aligned}$$

Adjusting weights changes the function: do learning this way!

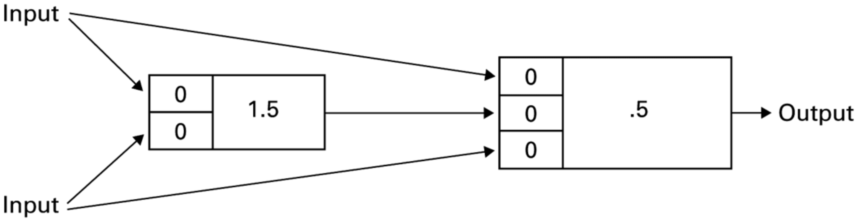
Neural Network with two layers



Multilayer Expressiveness

Basic Example

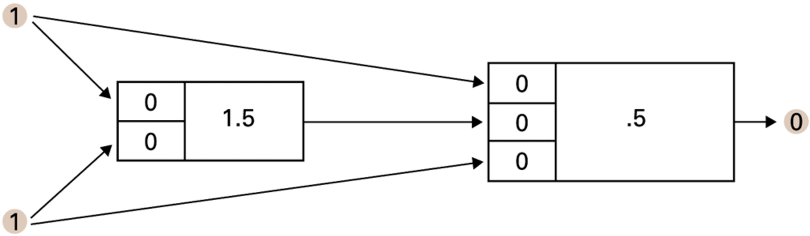
Exercise: set the weights in such a way that the network represents the XOR logical operator.



Multilayer Expressiveness

Basic Example

Exercise: set the weights in such a way that the network represents the XOR logical operator.

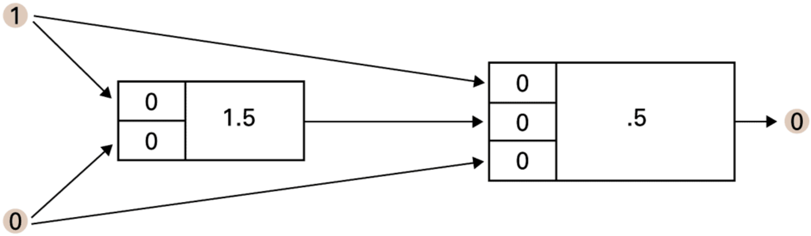


a. The network performs correctly for the input pattern 1, 1.

Multilayer Expressiveness

Basic Example

Exercise: set the weights in such a way that the network represents the XOR logical operator.

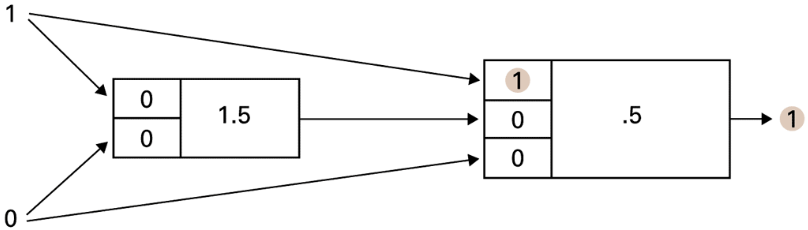


b. The network performs incorrectly for the input pattern 1, 0.

Multilayer Expressiveness

Basic Example

Exercise: set the weights in such a way that the network represents the XOR logical operator.

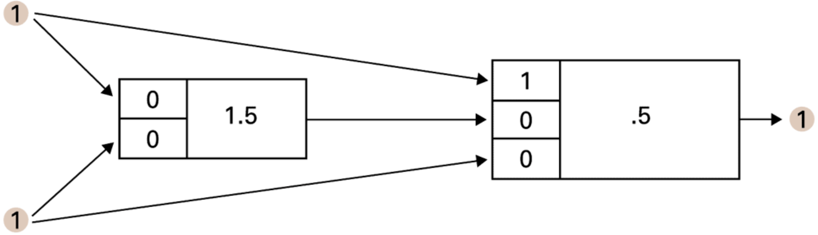


c. The upper weight in the second processing unit is adjusted.

Multilayer Expressiveness

Basic Example

Exercise: set the weights in such a way that the network represents the XOR logical operator.

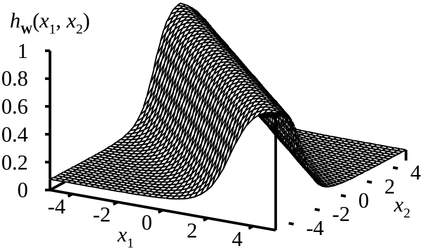


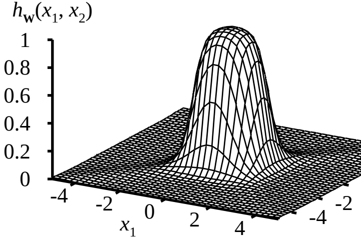
d. However, the network no longer performs correctly for the input pattern 1, 1.

...how should we continue?

Expressiveness of MLPs

All continuous functions with 2 layers, all functions with 3 layers





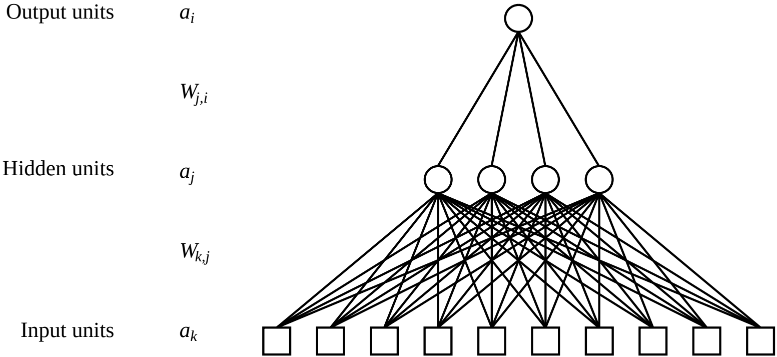
Combine two opposite-facing threshold functions to make a ridge
Combine two perpendicular ridges to make a bump
Add bumps of various sizes and locations to fit any surface
Proof requires exponentially many hidden units

Backpropagation Algorithm

- Supervised learning method to train multilayer feedforward NNs with diffrerentiable transfer functions.
- Adjust weights along the negative of the gradient of performance function.
- Forward-Backward pass.
- Sequential or batch mode
- Convergence time varies exponentially with number of inputs
- Avoid local minima by simulated annealing and other metaheuristics

Multilayer perceptrons

Layers are usually fully connected;
numbers of hidden units typically chosen by hand



Back-propagation learning

Output layer: same as for single-layer perceptron,

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$$

where $\Delta_i = Err_i \times g'(in_i)$.
Note: the general case has multiple output units hence: $\mathbf{Err} = (\mathbf{y} - \mathbf{h_w}(\mathbf{x}))$

Hidden layer: **back-propagate** the error from the output layer:

$$\Delta_j = g'(in_j) \sum_i W_{j,i} \Delta_i \qquad \text{(sum over the multiple output units)}$$

Update rule for weights in hidden layer:

$$W_{k,j} \leftarrow W_{k,j} + \alpha \times a_k \times \Delta_j \ .$$

(Most neuroscientists deny that back-propagation occurs in the brain)

Back-propagation derivation

The squared error on a single example is defined as

$$E = \frac{1}{2}\sum_i (y_i - a_i)^2 \ ,$$

where the sum is over the nodes in the output layer.

$$\begin{aligned}
\frac{\partial E}{\partial W_{j,i}} &= -(y_i - a_i)\frac{\partial a_i}{\partial W_{j,i}} = -(y_i - a_i)\frac{\partial g(in_i)}{\partial W_{j,i}} \\
&= -(y_i - a_i)g'(in_i)\frac{\partial in_i}{\partial W_{j,i}} = -(y_i - a_i)g'(in_i)\frac{\partial}{\partial W_{j,i}}\left(\sum_j W_{j,i}a_j\right) \\
&= -(y_i - a_i)g'(in_i)a_j = -a_j\Delta_i
\end{aligned}$$

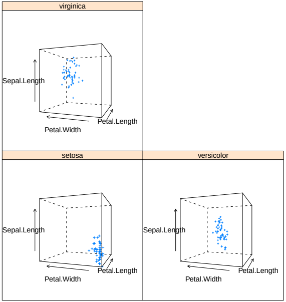
Back-propagation derivation contd.

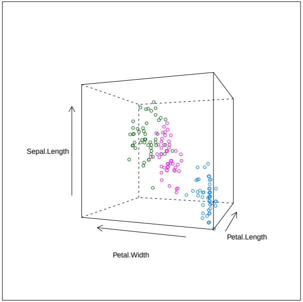
For the hidden layer:

$$\begin{aligned}
\frac{\partial E}{\partial W_{k,j}} &= -\sum_i (y_i - a_i)\frac{\partial a_i}{\partial W_{k,j}} = -\sum_i (y_i - a_i)\frac{\partial g(in_i)}{\partial W_{k,j}} \\
&= -\sum_i (y_i - a_i) g'(in_i)\frac{\partial in_i}{\partial W_{k,j}} = -\sum_i \Delta_i \frac{\partial}{\partial W_{k,j}}\left(\sum_j W_{j,i} a_j\right) \\
&= -\sum_i \Delta_i W_{j,i}\frac{\partial a_j}{\partial W_{k,j}} = -\sum_i \Delta_i W_{j,i}\frac{\partial g(in_j)}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial in_j}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial}{\partial W_{k,j}}\left(\sum_k W_{k,j} a_k\right) \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j) a_k = -a_k \Delta_j
\end{aligned}$$

Numerical Example

The (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables petal length and width, respectively, for 50 flowers from each of 2 species of iris. The species are "Iris setosa", and "versicolor".





Numerical Example

```
> samp <- c(sample(1:50, 25), sample(51:100, 25), sample(101:150, 25))
> Target <- class.ind(iris$Species)
> ir.nn <- nnet(Target ~ Sepal.Length * Petal.Length * Petal.Width, data
+     size = 2, rang = 0.1, decay = 5e-04, maxit = 200, trace = FALSE)
> test.cl <- function(true, pred) {
+     true <- max.col(true)
+     cres <- max.col(pred)
+     table(true, cres)
+ }
> test.cl(Target[-samp, ], predict(ir.nn, iris[-samp, c(1, 3, 4)]))
```

```
    cres
true  1  2  3
   1 25  0  0
   2  0 22  3
   3  0  2 23
```

Training and Assessment

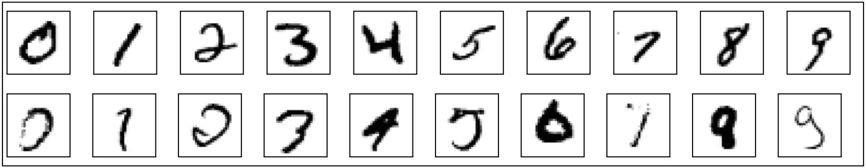
Use different data for different tasks:

- Training and Test data: holdout cross validation
- If little data: k-fold cross validation

Avoid peeking:

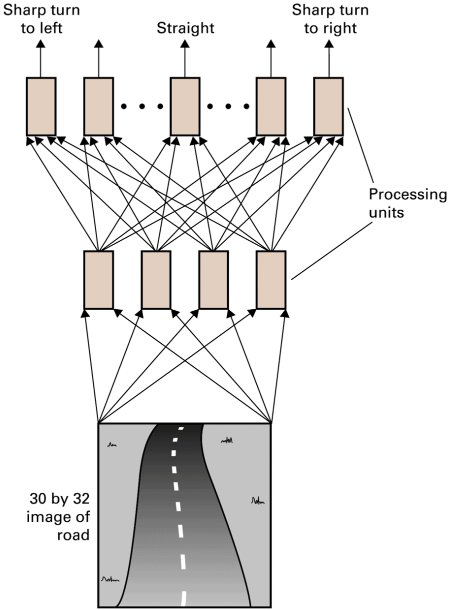
- Weights learned on training data.
- Parameters such as learning rate α and net topology compared on validation data
- Final assessment on test data

Handwritten digit recognition



- 400–300–10 unit MLP = 1.6% error
- LeNet: 768–192–30–10 unit MLP = 0.9% error
  http://yann.lecun.com/exdb/lenet/
- Current best (kernel machines, vision algorithms) ≈ 0.6% error
- Humans are at 0.2% – 2.5 % error

Another Practical Example



Directions of research in ANN

- Representational capability assuming unlimited number of neurons (no training)
- Numerical analysis or approximation theory: how many hidden units are necessary to achieve a certain approximation error? (no training) Results for single hidden layer and multiple hidden layers
- Sample complexity: how many samples are needed to characterize a certain unknown mapping.
- Efficient learning: backpropagation has the curse of dimensionality problem

Approximation properties

NNs with 2 hidden layers and arbitrarily many nodes can approximate any real-valued function up to any desired accuracy, using continuous activation functions

> *E.g.: required number of hidden units grows exponentially with number of inputs.*
> $2^n/n$ *hidden units needed to encode all Boolean functions of* n *inputs*

However proofs are not constructive.

More interest in efficiency issues: NNs with small size and depth

Size-depth trade off: more layers $\rightsquigarrow$ more costly to simulate

Recurrent Networks

Backpropagation through time solve temporal differentiable optimization problems with continuous variables

Recurrent Networks

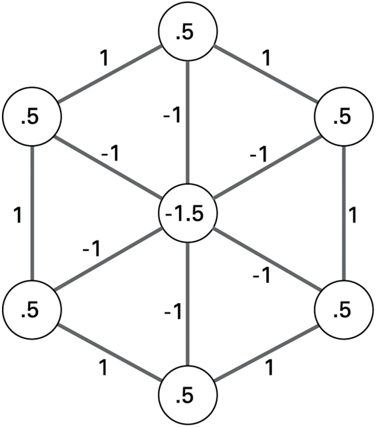
Associative Memory

- Associative memory: The retrieval of information relevant to the information at hand
- One direction of research seeks to build associative memory using neural networks that when given a partial pattern, transition themselves to a completed pattern.

Example

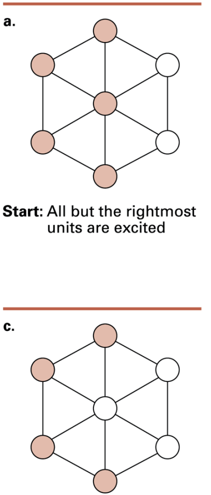
An artificial neural network implementing an associative memory

- symmetric weights ($W_{i,j} = W_{j,i}$);
  - $\sigma(x) = \text{sign}(x)$, $a_i = \{1, 0\}$;
- operates in synchronized discrete steps



Example

The steps leading to a stable configuration



Start: All but the rightmost units are excited

Step1: Only the leftmost units remain excited

Step 2: The top and bottom units become excited

Final: All the units on the perimeter are excited

Summary

- Supervised learning
- Perceptron learning rule: an algorithm for learning weights in single layered networks.
- Perceptrons: linear separators, insufficiently expressive
- Multi-layer networks are sufficiently expressive
- Many applications: speech, driving, handwriting, fraud detection, etc.
- Recurrent networks give rise to associative memory

Outline

1. Neuroscience

2. Artificial Neural Networks
   - Feedforward Networks
     - Single-layer perceptrons
     - Multi-layer perceptrons
   - Recurrent Networks

3. Other Applications
   - Simulations

Applications

- supervised learning: regression and classification
- associative memory
- optimization:

  R. Durbin and D. Willshaw. An analogue approach to the traveling salesman problem using and elastic net method. **Nature**, *326:689–691, 1987*

  J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. Biological Cybernetics. 52: 141–152,1985

- Unsupervised learning

  T. Kohonen. Self-Organizing and Associative Memory. Springer. Berlin 1988.

  (position of units incrementally adjusted – like weights in NNs – until sufficiently close to vertices.)

- grammatical induction, (aka, grammatical inference)
  e.g. in natural language processing
- noise filtering

Simulation of biological brains

Operationalize neuroscience data
Bottom-up approach
Cognitive computing

Simulations

Appropriate level of abstraction and resolution: the only solution is to experiment and explore as a community.

- AI: high levels of abstraction: cognitive science, visual information processing, connectionism, computational learning theory, and Bayesian belief networks
- Others: reductionist biological detail, exhaustive, biophysically accurate simulation.

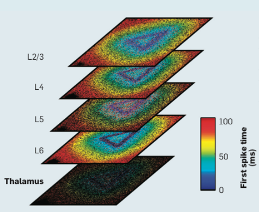
Mammalian-scale brain simulator

Neuroanatomy and neurophysiology, together, have produced a rich set of constraints on the structure and the dynamics of the brain.
Ingredients:

- phenomenological model of neurons exhibiting spiking communication, dynamic synaptic channels, plastic synapses, structural plasticity,
- multi-scale network architecture, including layers, minicolumns, hypercolumns, cortical areas, and multi-area networks,

Simultaneously achieving scale, speed, and detail in one simulation platform presents a formidable challenge with respect to the three primary resources of computing systems: memory, computation, and communication.

- Cortical simulation algorithms capable to simulate cat-scale cortex on Lawrence Livermore national Laboratory's Dawn Blue Gene/P supercomputer with 147,456 CPUs and 144TB of main memory.
- roughly equivalent to 4.5% of human scale
- The networks demonstrated self-organization of neurons into reproducible, time-locked, though not synchronous, groups.
- In a visual stimulation-like paradigm, the simulated network exhibited population-specific response latencies matching those observed in mammalian cortex.
- Figure outlines this activity, traveling from the thalamus to cortical layers four and six, then to layers two, three, and five, while simultaneously traveling laterally within each layer.



The realistic expectation:

- not that cognitive function will spontaneously emerge from these neurobiologically inspired simulations.
- rather, the simulator supplies a substrate, consistent with the brain, within which we can formulate and articulate theories of neural computation (mathematical theory of how the mind arises from the brain)
- it is a tool not the answer (a key integrative workbench for discovering algorithms of the brain)
- goal: building intelligent business machines.

Good news: human-scale cortical simulations are not only within reach but appear inevitable within a decade.
Bad news: the power and space requirements of such simulations may be many orders of magnitude greater than those of the biological brain.

Movement

- *Rodney Brooks (1989), "A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network", Neural Computation 1 (2): 253-262, doi:10.1162/neco.1989.1.2.253, http://people.csail.mit.edu/brooks/papers/AIM-1091.pdf*
- Asimo, 2006 http://www.youtube.com/watch?v=VTlV0Y5yAww
- Asimo, 2011 http://www.youtube.com/watch?v=eU93VmFyZbg
- Relevant applications in proteases

References

Brookshear J.G. (2009). **Computer Science - An Overview**. Pearson, 10th ed.

Kandel E.R., Schwartz J., and Jessell T. (eds.) (2000). **Principles of Neural Science**. McGraw-Hill, New York, US, 4th ed. 5th ed. expected for 2012 (ISBN 0-07-139011-1).

Luger G.F. (2009). **Artificial Intelligence: Structures and Strategies for Complex Problem Solving**. Addison-Wesley, Boston, MA, 6th ed.

Modha D.S., Ananthanarayanan R., Esser S.K., Ndirango A., Sherbondy A.J., and Singh R. (2011). **Cognitive computing**. Communication of the ACM, 54, pp. 62–71.

Russell S. and Norvig P. (2010). **Artificial Intelligence: A Modern Approach**. Prentice Hall, New Jersey, USA, third ed.

Searle J.R. (2004). **Mind: A Brief Introduction**. Oxford University Press.

Wikipedia (2011). **Gradient descent**.

Perception: understand how we construct subjective interpretations of proximal information from the environment.

Attention refers to the process by which organisms select a subset of available information upon which to focus for enhanced processing. orienting, filtering, and searching, and can either be focused upon a single information source or divided among several.

Learning: Learning improves the response of the organism to the environment. (i.e., habituation, conditioning, and instrumental, contingency, and associative learning)

Cognitive studies of implicit learning emphasize the largely automatic influence of prior experience on performance, and the nature of procedural knowledge

Memory is the record of experience represented in the brain. Long short term. All forms of memory are based on changes in synaptic connections within neural circuits of each memory system. multiple memory systems: elementary and non-associative types of memory, including habituation and sensitization, and reaching the most complex forms of associative memory, including episodic memory and semantic memory. distinct operating characteristics and brain pathways in which memories are embodied in the plasticity of information processing within the relevant neural circuit declarative memory, our ability for learning and consciously remembering everyday facts and events