Artificial Neural Networks

Lecture 1: Introduction

Tudor Berariu
tudor.berariu@gmail.com

Faculty of Automatic Control and Computers
University Politehnica of Bucharest

Lecture : 7^{th} of October, 2015
Last Updated: 30^{th} of December, 2015

Today's Outline

Today's Outline

Today's Outline

1. About this Course
   - What are Artificial Neural Networks?
   - Why study Artificial Neural Networks?
   - Course Syllabus
   - Course Goals
   - Resources
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch

What are Artificial Neural Networks?

Definition

Artificial Neural Networks are

What are Artificial Neural Networks?

Definition

Artificial Neural Networks are machines designed to model the information processing capabilities of animal nervous systems.

What *information processing capabilities* of the CNS are we interested in?

What do we know about the brain?

- **The brain** (nervous system) has a totally different architecture from the conventional computer.
- **The brain** is a highly complex, nonlinear, and parallel computer (information-processing system). [Hay09, page 1]

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a massively parallel distributed processor

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a massively parallel distributed processor made up of simple processing units

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.
It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.

What are artificial neural networks?

... back to the definition

Definition

An **artificial neural network** is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.
It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

[Hay09, page 2]

Artificial Neural Networks Description

An ANN consists of a large number of connected computing units:

- input units (correspond to sensory neurons)
- hidden units
- output units (correspond to motor neurons)

An example

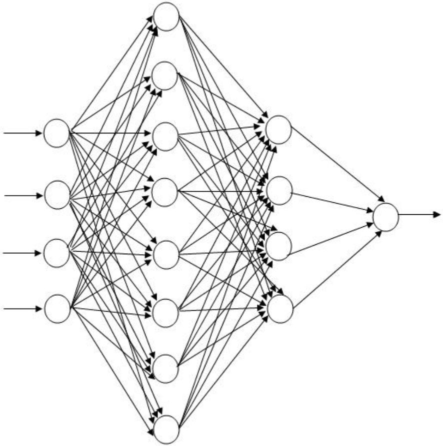


Figure: Multi-Layer Feedforward Network

What are ANNs composed of

The three elements important in any model of ANN:

1. the functions of the nodes
2. the topology of the network
3. the learning algorithm

Today's Outline

Why study on ANN begun?

The brain is many times **faster than** any **computer** on some specific tasks (although its synapses are slow compared to electronic logic gates):

- natural language processing
- sensorial perception
- motor control and movement planning

What else do we like about the human brain?

Appealing features of the brain:

- adaptivity (learning capabilities)
- robustness and fault tolerance
- its capability of dealing with fuzzy, noisy, and even inconsistent information

"Single learning algorithm" hypothesis

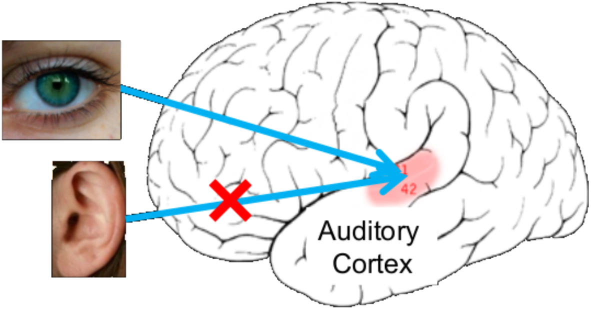


Figure: Rewiring in ferrets [RPKS92]

"Single learning algorithm" hypothesis

A blind guy climbing a wall?

"Single learning algorithm" hypothesis

He sees with his tongue...

source: http://www.bbc.com/news/health-13358608

"Single learning algorithm" hypothesis


THE TIMES OF INDIA

Science

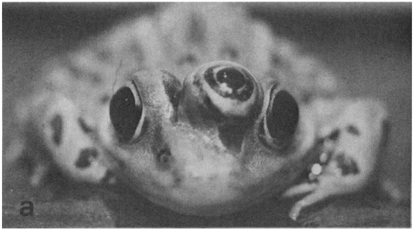
Home | City | India | World | Business | Tech | Sports | Cricket | Entertainment | TV | Life

Auto | Polls | Speak Out | Science | Environment | Education | STOI | Headlines | Specials

You are here: Home » Science

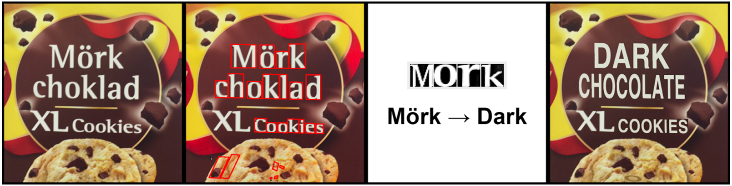
Rajasthan scientists produce 3-eyed frog

PTI | Jan 28, 2005, 11.42PM IST



Median third eye was found to develop from transplanted pineal gland [JSS+05]

Applications: Automatic Translation in Images



(source: `http://googleresearch.blogspot.ro/2015/07/how-google-translate-squeezes-deep.html`)

Applications: Speech Recognition

TECH 5/19/2014 @ 1:02PM | 5,840 views

Baidu's Coup: Ng Aims To Build Silicon Valley's New Brain Trust

+ Comment Now + Follow Comments

Baidu Inc. is one of world's Internet heavyweights, sporting a $55 billion market capitalization and the status of being the planet's fifth most popular website, thanks to the popularity of its Chinese-language search engine. Until now, though, Baidu has been practically invisible in Silicon Valley.

That's about to change.

Last week, Baidu announced that it is hiring Andrew Ng, a renowned Stanford computer-science professor who also has led artificial intelligence projects at Google and has cofounded Coursera, the online-education company. Ng will take charge of a five-year, $300 million research initiative for Baidu, spanning both China and Silicon Valley. Ng will be based in Sunnyvale, Calif., where Baidu has big plans to build up its own brain trust.

(source: `http://www.forbes.com/sites/georgeanders/2014/05/19/baidus-coup-ng-aims-to-build-silicon-valleys-new-brain-trust/`)

Applications: Speech Recognition

- Baidu used RNN to build a speech recognition system: DeepSpeech

TECH 12/18/2014 @ 9:00AM | 44,582 views

Baidu Announces Breakthrough In Speech Recognition, Claiming To Top Google And Apple

+ Comment Now + Follow Comments

When artificial-intelligence guru Andrew Ng joined Chinese Internet pioneer Baidu last May as chief scientist, he was a little cagey about what he and his team might work on at a newly opened lab in Sunnyvale, Calif. But he couldn't help revealing better speech recognition as a key area of interest in the age of the smartphone.

Today, Baidu, often called China's Google GOOGL -0.69%, unveiled the first results of what the former Google researcher, Stanford professor and Coursera cofounder had in mind. In a paper published today on Cornell University Library's arXiv.org site, Ng and 10 members of his Baidu Research team led by research scientist Awni Hannun said they've come up with a new method of more accurately recognizing speech, an increasingly important feature used in Apple's AAPL -2.46% Siri and Dictation services as well as Google's voice search. Baidu's Deep Speech beat other methods such as those offered by Google and Apple on standard benchmarks that measure the error rate of speech recognition systems, according to Ng.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., *Deepspeech: Scaling up end-to-end speech recognition*, arXiv preprint arXiv:1412.5567 (2014)

(source: `https://gigaom.com/2014/12/18/baidu-claims-deep-learning-breakthrough-with-deep-speech/`)

Applications: Face detection

Figure: Face detection (source: http://www.cs.nyu.edu/~yanqn/research/cface/)

Applications: Object Recognition

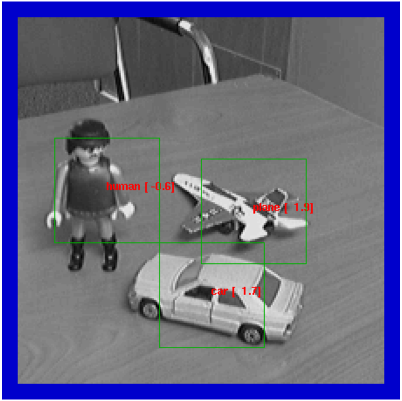
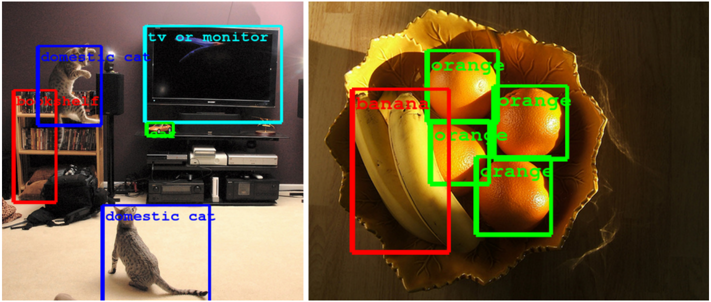


Figure: Object detection and recognition (source: http://www.cs.nyu.edu/~yann/research/norb/)

GoogLeNet @ ILSVRC 2014



- 6.7% error rate for Hit@5 classification test
- http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
- http://googleresearch.blogspot.ro/2014/09/building-deeper-understanding-of-images.html

Today's Outline

1. About this Course
   - What are Artificial Neural Networks?
   - Why study Artificial Neural Networks?
   - Course Syllabus
   - Course Goals
   - Resources
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch

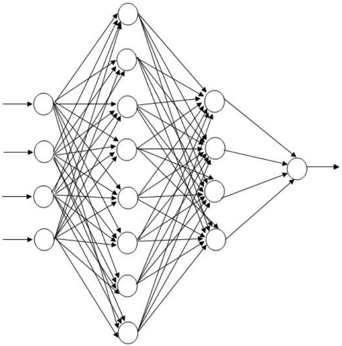
Course syllabus

- ADALINE and the Perceptron

- Both are Single Layer Networks
- Algorithms:
  - Least Mean Squares Rule (Widrow, Hoff, 1960)
  - Perceptron learning (Rosenblatt, 1956)
- Both solve only linearly separable problems

Course syllabus

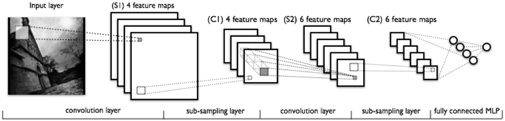
- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks



- Error Backpropagation
- Learning Algorithms

Course syllabus

- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks

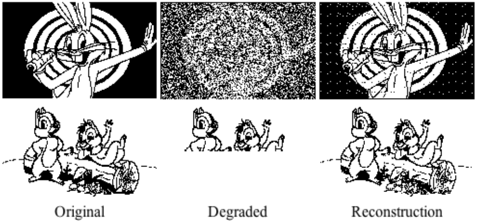


(source: `http://deeplearning.net/tutorial/lenet.html`)

- CNNs are used for computer vision problems
- Recently CNNs improved record scores in image recognition contests
- CNNs use receptive fields, weight sharing

Course syllabus

- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks

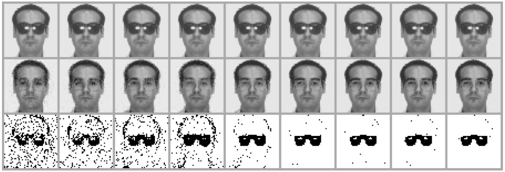


(source: `fourier.eng.hmc.edu/e161/lectures/nn/node5.html`)

- Used for pattern recognition
- Auto-associative memories

Course syllabus

- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks
- Boltzmann Machines

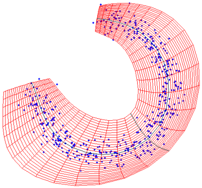


(image from [TSH12])

- Stochastic version of the Hopfield Network with hidden units
- Boltzmann Machines escape local minima where Hopfield Networks get stuck

Course syllabus

- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks
- Boltzmann Machines
- Autoencoders



(source: http://www.nlpca.org/)

- PCA - finding an lower-dimensional space in which the data has the most variance
- PCA with (deep) neural networks

Course syllabus

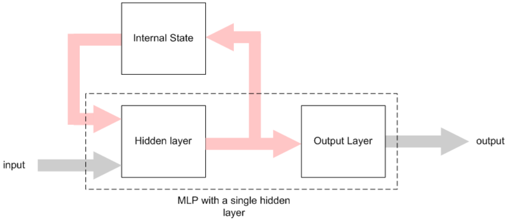
- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks
- Boltzmann Machines
- Autoencoders
- Self-Organizing Maps

(source: https://www.youtube.com/watch?v=71wmOT4lHWc)

- Kohonen, 1982
- Extract structure from data (Unsupervised learning method)
- Competitive learning

Course syllabus

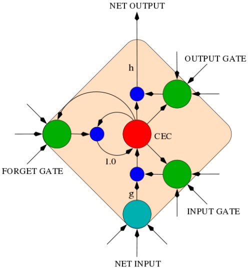
- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks
- Boltzmann Machines
- Autoencoders
- Self-Organizing Maps
- Recurrent Networks



- feedback loops
- internal states
- good for sequential data (time series, signal processing)
- hard to train

Course syllabus

- ADALINE and the Perceptron
- Multi-Layer Feedforward Networks
- Convolutional Neural Networks
- Hopfield Networks
- Boltzmann Machines
- Autoencoders
- Self-Organizing Maps
- Recurrent Networks
- Memory Networks



(source: `http://people.idsia.ch/~juergen/rnn.html`)

- LSTM cells [HS97]
- Neural Turing Machines [GWD14]
- Memory Networks [WCB14]

Today's Outline

1. About this Course
   - What are Artificial Neural Networks?
   - Why study Artificial Neural Networks?
   - Course Syllabus
   - Course Goals
   - Resources
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch

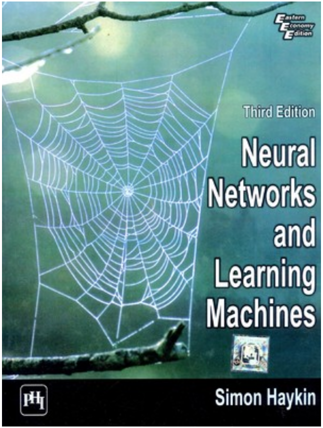
Course Goals

- understand the **fundamental principles** of artificial neural networks and their relation to biological systems
- understand the **different architectures** of neural networks
- learn how to **train** neural networks
- know which **problems** are suitable for neural networks
- familiarize yourself with **Torch**

Today's Outline

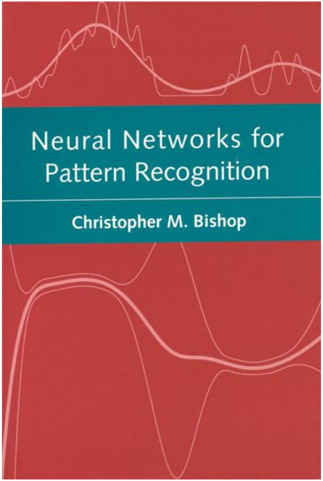
1. About this Course
   - What are Artificial Neural Networks?
   - Why study Artificial Neural Networks?
   - Course Syllabus
   - Course Goals
   - Resources
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch

Haykin, 2009



Simon S. Haykin, *Neural networks and learning machines*, Prentice Hall, 2009

Bishop, 1995



Christopher M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, Inc., New York, NY, USA, 1995

More recent resources for deep learning

Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, Book in preparation for MIT Press, 2015

`http://www.deeplearningbook.org/`

... plus various recent papers

Today's Outline

Today's Outline

What is Machine Learning?

Definition

Machine Learning

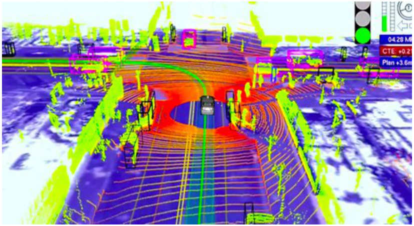
What is Machine Learning?

Definition

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. (Arthur Samuel, 1959)

Applications of Machine Learning

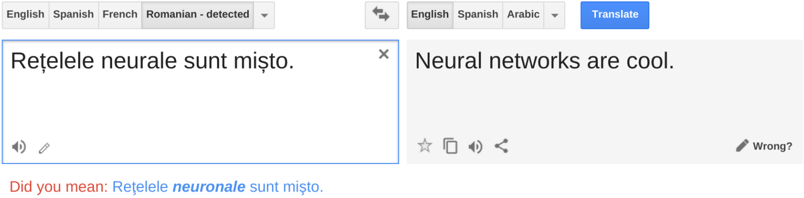
- Self-Driving Car: Google Car
- Machine Translation: Google Translate
- Recommender Systems
  - Movies: ImDB, NetFlix
  - Intelligent Advertising: Google Ads, Facebook Ads



images from www.nytimes.com

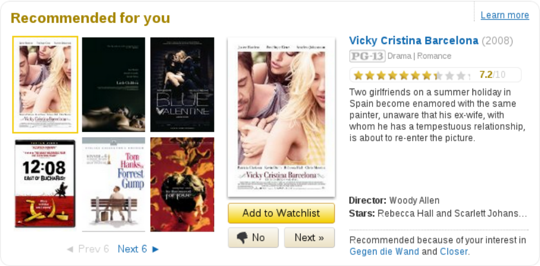
Applications of Machine Learning

- Self-Driving Car: Google Car
- Machine Translation: Google Translate
- Recommender Systems
  - Movies: ImDB, NetFlix
  - Intelligent Advertising: Google Ads, Facebook Ads



Applications of Machine Learning

- Self-Driving Car: Google Car
- Machine Translation: Google Translate
- Recommender Systems
    - Movies: ImDB, NetFlix
    - Intelligent Advertising: Google Ads, Facebook Ads



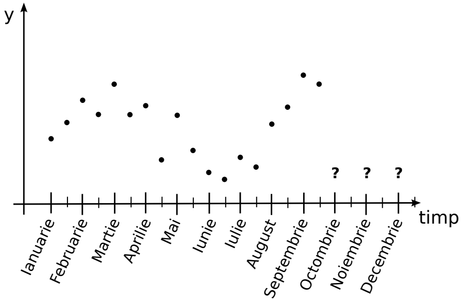
Applications of Machine Learning

- Self-Driving Car: Google Car
- Machine Translation: Google Translate
- Recommender Systems
  - Movies: ImDB, NetFlix
  - Intelligent Advertising: Google Ads, Facebook Ads

Machine Learning: Types of Problems

Problem Types

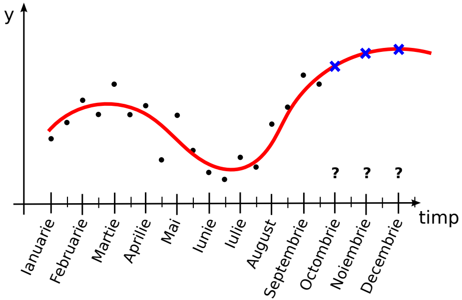
- Regression
  - predicting the market price of a good
- Classification
  - object classification in an image



Machine Learning: Types of Problems

Problem Types

- Regression
  - predicting the market price of a good
- Classification
  - object classification in an image



Machine Learning: Types of Problems

Problem Types

- Regression
  - predicting the market price of a good
- Classification
  - object classification in an image

image from Albert-Ludwigs-Universität,

Lehrstuhl für Mustererkennung und Bildverarbeitung

Machine Learning: Types of Problems

Problem Types

- Regression
  - predicting the market price of a good
- Classification
  - object classification in an image

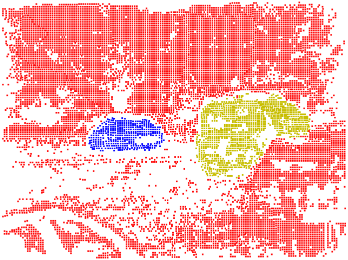


image from Albert-Ludwigs-Universität,

Lehrstuhl für Mustererkennung und Bildverarbeitung

Types of machine learning

Definition

supervised learning

Definition

unsupervised learning

Definition

reinforcement learning

Types of machine learning

Definition

Problems in which training data comprising of input-target pairs is available are called supervised learning problems.

Definition

Problems in which training data consists of input vectors without any target labels are called unsupervised learning problems.

Definition

Problems in which an agent learns actions to take in order to maximize a [long-term] reward are known as reinforcement learning problems.

Machine Learning goals

Phases of a Machine Learning algorithm:

- training a model
- testing the model

Definition

generalization

Machine Learning goals

Phases of a Machine Learning algorithm:

- training a model
- testing the model

Definition

The ability of a model to perform well on the testing set is called generalization.

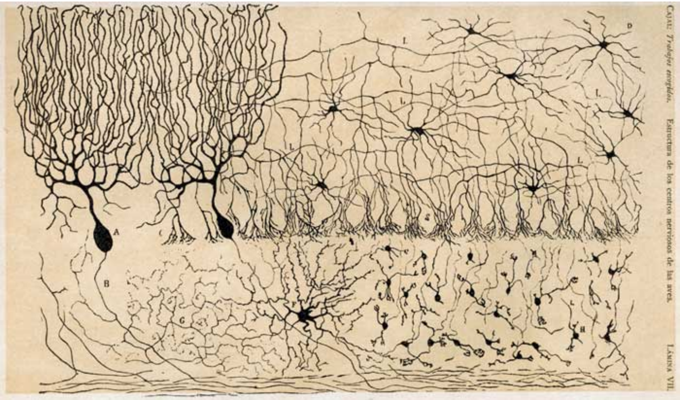
How do we evaluate a model?

- training set
- validation set
- test set
- cross-validation
- overfitting, underfitting

Today's Outline

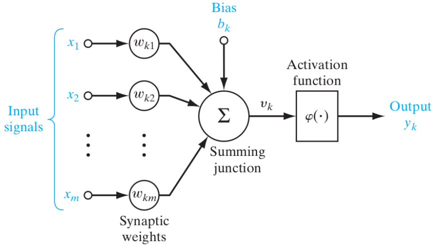
The neuron theory vs. the reticular theory

- The nervous system was not properly studied until late 1800s.
- Camillo Golgi and **Ramón y Cajal** ⟶ *the neuron doctrine*.



McCulloch and Pitts

- Origins of both connectionist and symbolic AI paradigms:
  - Warren S McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5** (1943), no. 4, 115–133
- Each neuron's spike represents the truth value of a proposition.
- A neuron combines the truth values of other propositions (neurons) in order to compute its own.



Timeline

- '40-'60 McCulloch and Pitts, Alan Turing, Frank Rosenblatt
- '69 Paper and Minsky - *Perceptrons*
- '70 - '80 Neural Network *Dark ages*
- '86 Rumelhart, Hinton and Williams (multi-layer perceptrons, error backpropagation)
- '00 Era of Kernel Methods (SVM, Kernel-PCA)
- last years Deep Learning

Today's Outline

1. About this Course
2. Neural Networks for Machine Learning
   - Machine Learning Refresher
   - History of research in ANN
   - Important researchers
3. Biological Neurons and Artificial Neurons
4. Torch

Conferences

There are two major annual conferences:

- Neural Information Processing Systems (`https://nips.cc/`)
- International Conference on Machine Learning (`http://icml.cc/`)

Hinton's group @ University of Toronto

- Geoffrey E. Hinton (`http://www.cs.toronto.edu/~hinton/`)
  - `https://www.coursera.org/course/neuralnets`
- Alex Graves (`http://www.cs.toronto.edu/~graves/`)
- Alex Krizhevsky (`http://www.cs.toronto.edu/~kriz/`)
- Tijmen Tieleman (`http://www.cs.toronto.edu/~tijmen/`)
- Ruslan Salakhutdinov (`http://www.cs.toronto.edu/~rsalakhu/index.html`)

Why?

- Hinton invented most of the algorithms used today in NN.
- They are leading researchers in the field.

Yann Lecun

- Yann Lecun (Director of AI Research at Facebook; NY University)
  - `http://yann.lecun.com/`
  - `https://www.youtube.com/watch?v=M7smwHwdOIA`

Why?

- one of the founding fathers of convolutional neural nets
- *optimal brain damage*

Yoshua Bengio

- http://www.iro.umontreal.ca/~bengioy/yoshua_en/index.html
- Book still to be published:
  Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, Book in preparation for MIT Press, 2015
- Read this paper by Bengio, Hinton and Lecun:
  http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html
- Read this article:

  Yoshua Bengio, *Deep learning of representations: Looking forward*, Proceedings of the First International Conference on Statistical Language and Speech Processing (Berlin, Heidelberg), SLSP'13, Springer-Verlag, 2013, pp. 1–37

IDSIA

- Jürgen Schmidhuber (http://people.idsia.ch/~juergen/)
  - Together with Hochreiter, inventors of LSTM [HS97]
  - One of the pioneers of Deep Learning
    http://people.idsia.ch/~juergen/deeplearning.html
- Dan Cireșan (http://people.idsia.ch/~ciresan/)

Google's Deep Mind

`http://deepmind.com/`

- Demis Hassabis
- Volodymyr Mnih (`https://www.cs.toronto.edu/~vmnih/`)
- Alex Graves
- David Silver (`http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Home.html`)
- Nando de Freitas
- ...

Important results:

- Deep Reinforcement Learning [MKS+15]
- Neural Turing Machines [GWD14]
- Spatial Transformer Networks [JSZK15]

Some blogs

- Christopher Olah (Google)
  - http://colah.github.io/
- Andrej Karpathy (Stanford)
  - http://karpathy.github.io/
  - http://cs.stanford.edu/people/karpathy/
- Yarin Gal (Cambridge)
  - http://mlg.eng.cam.ac.uk/yarin/blog.html
- Recent publications related to Deep Learning
  - http://memkite.com/deep-learning-bibliography/

Today's Outline

Today's Outline

1. About this Course
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
   - The Neuron
   - Neuronal models
4. Torch

The real neuron

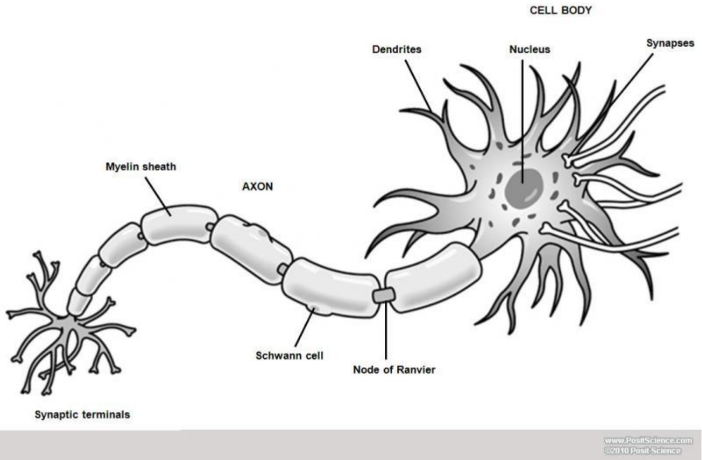


Figure: A neuron from the CNS

The McCulloch-Pitts neuron

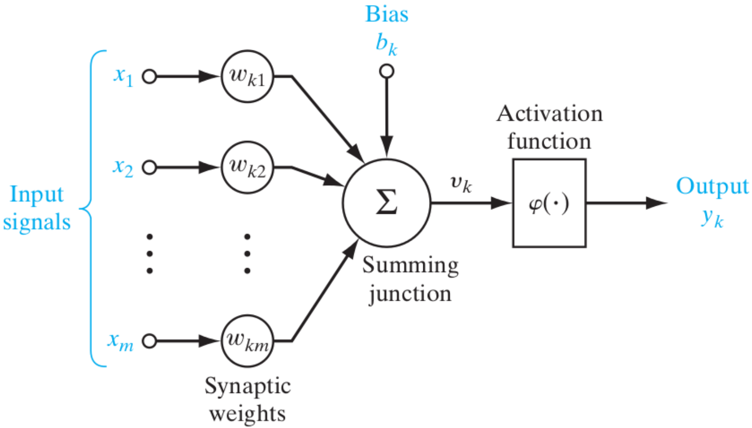


Figure: The McCulloch-Pitts neuron model (taken from [Hay09])

Neuron Models

- when building ANNs we do not consider the full complexity of real biological neurons
- models exclude details that are not useful
- on simplified models we can apply mathematics

Today's Outline

Elements of the neural model

The basic elements of the neural model:

connecting links - each characterized by a synaptic weight

adder - a function that sums the inputs (*linear combiner)*

activation function - limits the amplitude of the neuron

Linear neurons

Definition

Linear neurons:

$$y_k = b_k + \sum_i x_i \cdot w_{ki} \tag{1}$$

y_k - output of the neuron

b_k - bias

i - index over input synapses

x_i - i^{th} input

w_{ki} - weight for connection between neuron k and input i

Linear neurons

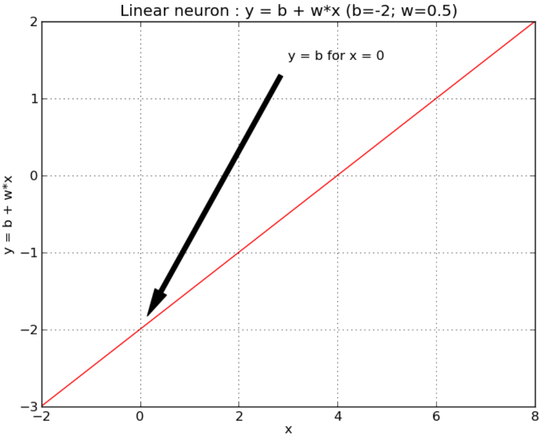


Figure: Linear neuron $y = b + w \cdot x$ for $w = 0.5, b = -2$

Binary Threshold Neurons

Definition

Binary threshold neurons (McCulloch and Pitts, 1943):

$$z_k = b_k + \sum_i x_i \cdot w_{ki} \tag{2}$$

$$y_k = \begin{cases} 0 & \text{if } z_k < 0 \\ 1 & \text{if } z_k \geq 0 \end{cases} \tag{3}$$

y_k - output of the neuron

b_k - bias

x_i - i^{th} input

w_{ki} - weight for connection between neuron k and input i

z_k - induced local field

Binary Threshold Neuron

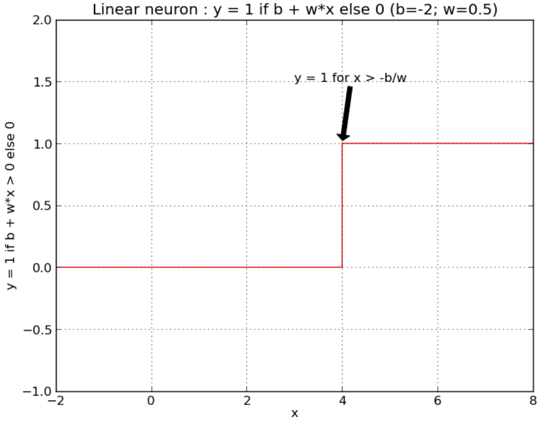


Figure: Binary Threshold Neuron

Rectified Linear Neurons

Definition

Rectified linear neurons:

$$z_k = b_k + \sum_i x_i \cdot w_{ki} \tag{4}$$

$$y_k = \begin{cases} z_k & \text{if } z_k \geq 0 \\ 0 & \text{if } z_k < 0 \end{cases} \tag{5}$$

- y_k - output of the neuron
- b_k - bias
- x_i - i^{th} input
- w_{ki} - weight for connection between neuron k and input i
- z_k - induced local field

Rectified Linear Neuron

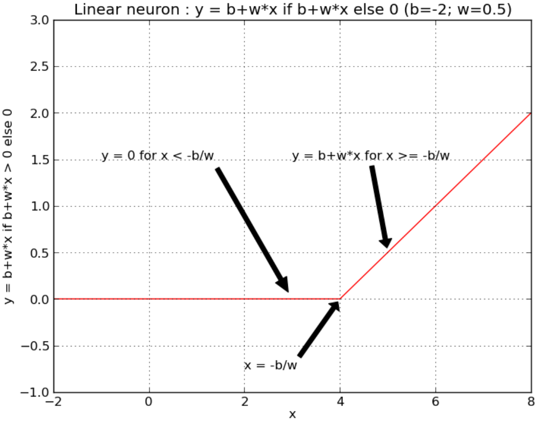


Figure: Rectified Linear Neuron

Sigmoid neurons I

- the binary threshold neuron activation function is not differentiable
- smooth, continuous functions that approximate the Heaviside step function would be nice
- easy to compute derivatives would be great too

Sigmoid neurons II

Definition

$$z_k = b_k + \sum_i x_i \cdot w_{ki} \tag{6}$$

The logistic function for activation:

$$y_k = \frac{1}{1 + e^{\alpha \cdot z_k}} \tag{7}$$

- Here is the nice derivative:

$$\frac{d}{dx} f(x) = f(x) \cdot (1 - f(x)) \tag{8}$$

Logistic function I

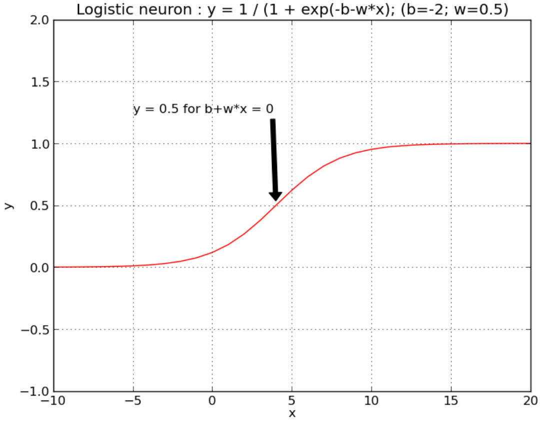


Figure: Logistic function

Logistic function II

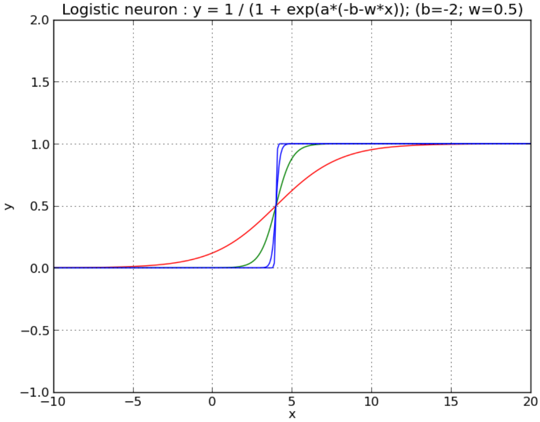


Figure: Logistic function for different values for α

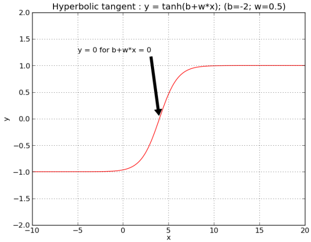
Hyperbolic tangent

Definition

$$z_k = b_k + \sum_i x_i \cdot w_{ki} \tag{9}$$

The logistic function for activation:

$$y_k = \frac{e^{z_k} - e^{-z_k}}{e^{z_k} + e^{-z_k}} \tag{10}$$



Stochastic neurons

- use the same logistic activation function, but as a probability of firing

Definition

Stochastic binary neurons:

$$z_k = b_k + \sum_i x_i \cdot w_{ki} \tag{11}$$

The probability of being active:

$$p(y_k = 1) = \frac{1}{1 + e^{z_k}} \tag{12}$$

Today's Outline

Today's Outline

1. About this Course
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch
   - Why Torch?
   - Short Introduction to Lua

What is Torch?

- Torch7 (`http://torch.ch/`)
- scientific computing framework for machine learning
- uses a fast scripting language: Lua

Why Torch?

- really fast neural networks and optimization libraries
- large list of community-driven packages
- automatic parallelization over CPUs and GPUs
- contributors:
  - researchers @ Deep Mind
  - researchers @ Facebook https://research.facebook.com/blog/879898285375829/fair-open-sources-deep-learning-modules-for-torch/
  - Twitter, NVIDIA, AMD, etc.

- Read this comparison between various machine learning platforms: https://github.com/zer0n/deepframeworks/blob/master/README.md

Today's Outline

1. About this Course
2. Neural Networks for Machine Learning
3. Biological Neurons and Artificial Neurons
4. Torch
   - Why Torch?
   - Short Introduction to Lua

Introduction to Lua

The following slides follow the Lua tutorial from here: `http://tylerneylon.com/a/learn-lua/`.

Read the full tutorial!

Comments and Variables in Lua

```lua
-- Two dashes start a one-line comment.

--[[
   Adding two ['s and ]'s makes it a
   multi-line comment.
--]]

x = 23
```

source: `http://tylerneylon.com/a/learn-lua/`.

Data Types in Lua; Garbage Collection

```lua
num = 42  -- All numbers are doubles.
-- Don't freak out, 64-bit doubles have 52 bits for
-- storing exact int values; machine precision is
-- not a problem for ints that need < 52 bits.

s = 'walternate'  -- Immutable strings like Python.
t = "double-quotes are also fine"
u = [[ Double brackets
       start and end
       multi-line strings.]]
t = nil  -- Undefines t; Lua has garbage collection.
```

source: http://tylerneylon.com/a/learn-lua/.

Flow Control (I)

```lua
-- Blocks are denoted with keywords like do/end:
while num < 50 do
  num = num + 1  -- No ++ or += type operators.
end
```

source: http://tylerneylon.com/a/learn-lua/.

Flow Control (II)

```lua
-- If clauses:
if num > 40 then
  print('over 40')
elseif s ~= 'walternate' then  -- ~= is not equals.
  -- Equality check is == like Python; ok for strs.
  io.write('not over 40\n')  -- Defaults to stdout.
else
  -- Variables are global by default.
  thisIsGlobal = 5  -- Camel case is common.
  -- How to make a variable local:
  local line = io.read()  -- Reads next stdin line.
  -- String concatenation uses the .. operator:
  print('Winter is coming, ' .. line)
end
```

Falsy values

```
-- Undefined variables return nil.
-- This is not an error:
foo = anUnknownVariable  -- Now foo = nil.

aBoolValue = false

-- Only nil and false are falsy; 0 and '' are true!
if not aBoolValue then print('twas false') end

-- 'or' and 'and' are short-circuited.
ans = aBoolValue and 'yes' or 'no'  --> 'no'
```

source: `http://tylerneylon.com/a/learn-lua/`.

Falsy values

```
karlSum = 0
for i = 1, 100 do  -- The range includes both ends.
  karlSum = karlSum + i
end
-- Use "100, 1, -1" as the range to count down:
fredSum = 0
for j = 100, 1, -1 do fredSum = fredSum + j end
```

source: http://tylerneylon.com/a/learn-lua/.

Another loop construct

```
repeat
  print('the way of the future')
  num = num - 1
until num == 0
```

source: http://tylerneylon.com/a/learn-lua/.

Functions

```lua
function fib(n)
  if n < 2 then return 1 end
  return fib(n - 2) + fib(n - 1)
end
```

source: http://tylerneylon.com/a/learn-lua/.

Anonymous functions

```lua
-- Closures and anonymous functions are ok:
function adder(x)
  -- The returned function is created when adder is
  -- called, and remembers the value of x:
  return function (y) return x + y end
end
a1 = adder(9)
a2 = adder(36)
print(a1(16))  --> 25
print(a2(64))  --> 100
```

source: `http://tylerneylon.com/a/learn-lua/`.

Functions' senders and receivers

```lua
-- Returns, func calls, and assignments all work
-- with lists that may be mismatched in length.
-- Unmatched receivers are nil;
-- unmatched senders are discarded.

x, y, z = 1, 2, 3, 4
-- Now x = 1, y = 2, z = 3, and 4 is thrown away.

function bar(a, b, c)
  print(a, b, c)
  return 4, 8, 15, 16, 23, 42
end
x, y = bar('zaphod')  --> prints "zaphod  nil nil"
-- Now x = 4, y = 8, values 15..42 are discarded.
```

Functions are first class values

```lua
-- Functions are first-class, may be local/global.
-- These are the same:
function f(x) return x * x end
f = function (x) return x * x end

-- And so are these:
local function g(x) return math.sin(x) end
local g; g  = function (x) return math.sin(x) end
-- the 'local g' decl makes g-self-references ok.
```

source: `http://tylerneylon.com/a/learn-lua/`.

Another loop construct

```lua
-- Tables = Lua's only compound data structure;
--          they are associative arrays.
-- Similar to php arrays or js objects, they are
-- hash-lookup dicts that can also be used as lists.

-- Using tables as dictionaries / maps:

-- Dict literals have string keys by default:
t = {key1 = 'value1', key2 = false}
```

source: `http://tylerneylon.com/a/learn-lua/`.

Another loop construct

```lua
-- Dict literals have string keys by default:
t = {key1 = 'value1', key2 = false}

-- String keys can use js-like dot notation:
print(t.key1)  -- Prints 'value1'.
t.newKey = {}  -- Adds a new key/value pair.
t.key2 = nil   -- Removes key2 from the table.

-- Literal notation for any (non-nil) value as key:
u = {['@!#'] = 'qbert', [{}] = 1729, [6.28] = 'tau'}
print(u[6.28])  -- prints "tau"
```

source: `http://tylerneylon.com/a/learn-lua/`.

Another loop construct

```lua
-- Key matching is basically by value for numbers
-- and strings, but by identity for tables.
a = u['@!#']  -- Now a = 'qbert'.
b = u[{}]     -- We might expect 1729, but it's nil:
-- b = nil since the lookup fails. It fails
-- because the key we used is not the same object
-- as the one used to store the original value. So
-- strings & numbers are more portable keys.
```

source: `http://tylerneylon.com/a/learn-lua/`.

Another loop construct

```lua
-- A one-table-param function call needs no parens:
function h(x) print(x.key1) end
h{key1 = 'Sonmi~451'}  -- Prints 'Sonmi~451'.

for key, val in pairs(u) do  -- Table iteration.
  print(key, val)
end
```

source: `http://tylerneylon.com/a/learn-lua/`.

Another loop construct

```lua
-- _G is a special table of all globals.
print(_G['_G'] == _G)  -- Prints 'true'.
```

source: http://tylerneylon.com/a/learn-lua/.

Another loop construct

```lua
-- Using tables as lists / arrays:

-- List literals implicitly set up int keys:
v = {'value1', 'value2', 1.21, 'gigawatts'}
for i = 1, #v do  -- #v is the size of v for lists.
  print(v[i])  -- Indices start at 1 !! SO CRAZY!
end
-- A 'list' is not a real type. v is just a table
-- with consecutive integer keys, treated as a list.
```

source: `http://tylerneylon.com/a/learn-lua/`.

Today's Outline

What to read

Read ...

- ... the *Introduction* chapter from [Hay09]

Topics for the exam

- For the exam you should be able to:
  - ... define basic machine learning concepts (e.g. regression, classification, cross-validation);
  - ... give a general description of artificial neural networks (why is it worth studying them, how do they resemble biological neural networks);
  - ... know the usual activation functions used in artificial neural networks.

Today's Outline

5 For the exam

6 References

References I

- Yoshua Bengio, *Deep learning of representations: Looking forward*, Proceedings of the First International Conference on Statistical Language and Speech Processing (Berlin, Heidelberg), SLSP'13, Springer-Verlag, 2013, pp. 1–37.
- Christopher M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, Inc., New York, NY, USA, 1995.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, Book in preparation for MIT Press, 2015.
- Alex Graves, Greg Wayne, and Ivo Danihelka, *Neural turing machines*, arXiv preprint arXiv:1410.5401 (2014).
- Simon S. Haykin, *Neural networks and learning machines*, Prentice Hall, 2009.

References II

- Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., *Deepspeech: Scaling up end-to-end speech recognition*, arXiv preprint arXiv:1412.5567 (2014).
- Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.
- OP Jangir, P Suthar, DVS Shekhawat, P Acharya, KK Swami, and Manshi Sharma, *The" third eye"-a new concept of trans-differentiation of pineal gland into median eye in amphibian tadpoles of bufo melanostictus*, Indian journal of experimental biology **43** (2005), no. 8, 671.

References III

- Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, *Spatial transformer networks*, arXiv preprint arXiv:1506.02025 (2015).

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, *Human-level control through deep reinforcement learning*, Nature **518** (2015), no. 7540, 529–533.

- Warren S McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5** (1943), no. 4, 115–133.

References IV

- Anna W Roe, Sarah L Pallas, Young H Kwon, and Mriganka Sur, *Visual projections routed to the auditory pathway in ferrets: receptive fields of visual neurons in primary auditory cortex*, The Journal of neuroscience **12** (1992), no. 9, 3651–3664.
- Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton, *Robust boltzmann machines for recognition and denoising*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2264–2271.
- Jason Weston, Sumit Chopra, and Antoine Bordes, *Memory networks*, arXiv preprint arXiv:1410.3916 (2014).