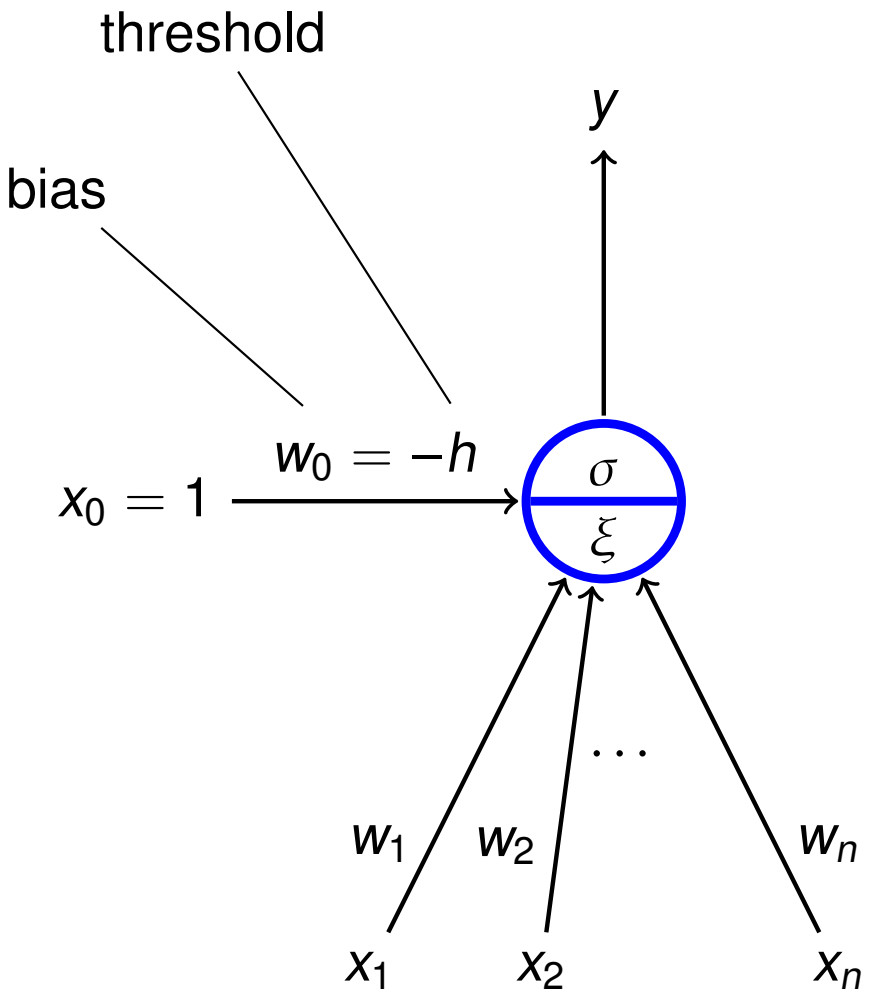
Formal neuron (with bias)



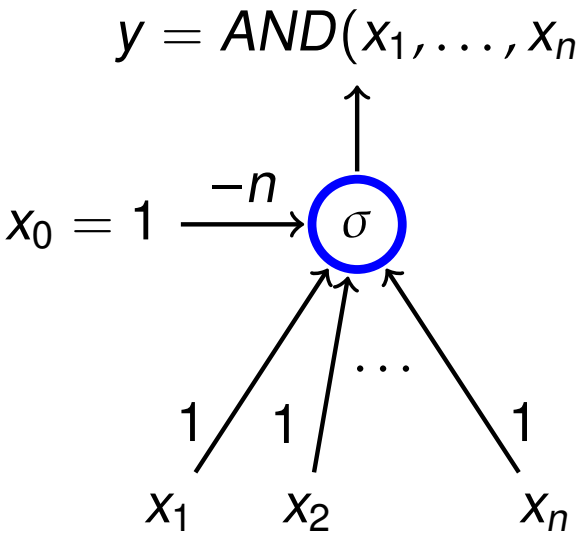
- $x_0 = 1, x_1, \ldots, x_n \in \mathbb{R}$ are **inputs**
- $w_0, w_1, \ldots, w_n \in \mathbb{R}$ are **weights**
- ξ is an **inner potential**; almost always $\xi = w_0 + \sum_{i=1}^{n} w_i x_i$
- y is an **output** given by $y = \sigma(\xi)$ where σ is an **activation function**;

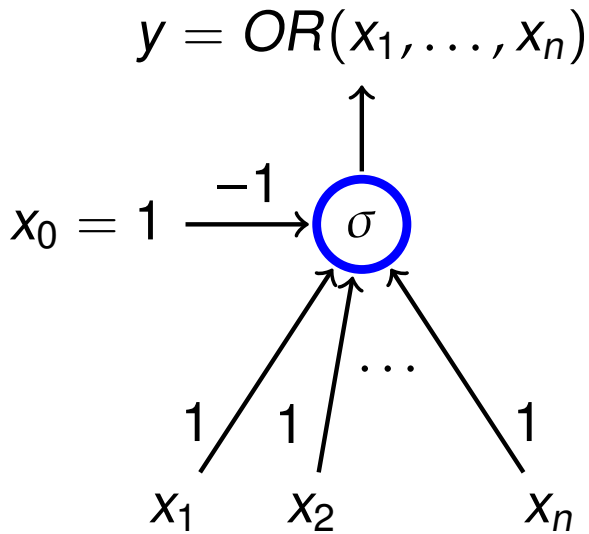
  e.g. a *unit step function*

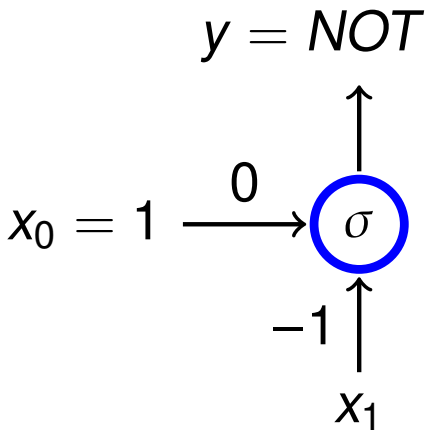
$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0\,; \\ 0 & \xi < 0. \end{cases}$$

Boolean functions

Activation function: *unit step function* $\sigma(\xi) = \begin{cases} 1 & \xi \geq 0\,; \\ 0 & \xi < 0. \end{cases}$







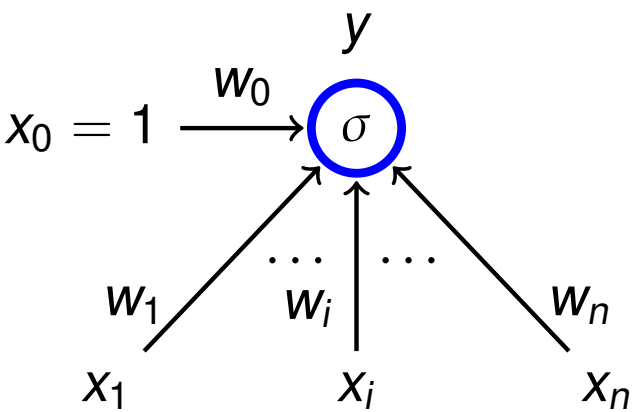
Boolean functions

Theorem

Let σ be the unit step function. Two layer MLPs, where each neuron has σ as the activation function, are able to compute all functions of the form $F : \{0,1\}^n \to \{0,1\}$.

Proof.

- Given a vector $\vec{v} = (v_1, \ldots, v_n) \in \{0,1\}^n$, consider a neuron $N_{\vec{v}}$ whose output is 1 iff the input is $\vec{v}$:

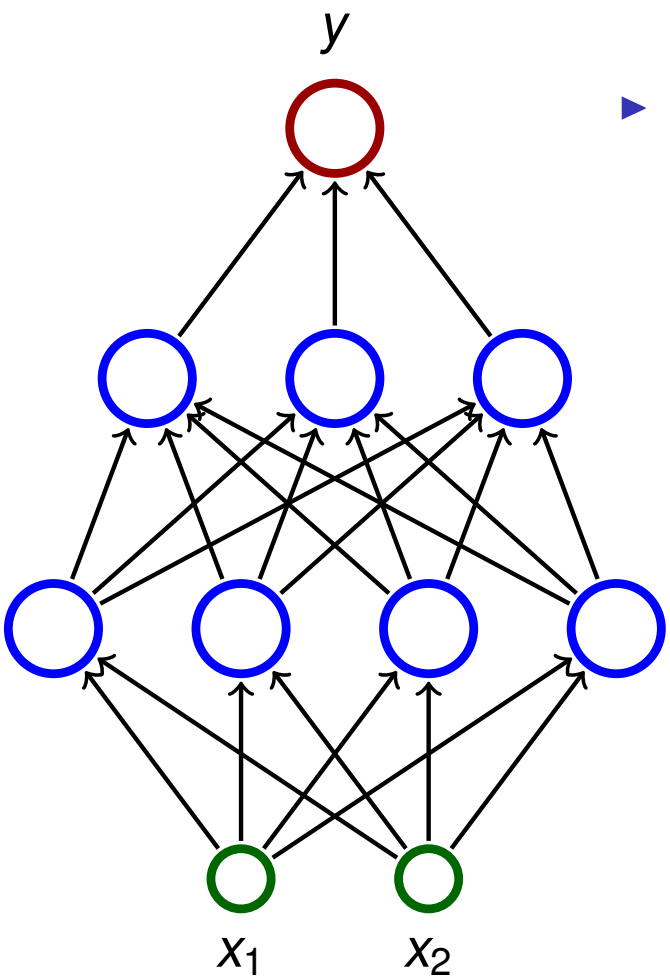


$$w_0 = -\sum_{i=1}^{n} v_i$$

$$w_i = \begin{cases} 1 & v_i = 1 \\ -1 & v_i = 0 \end{cases}$$

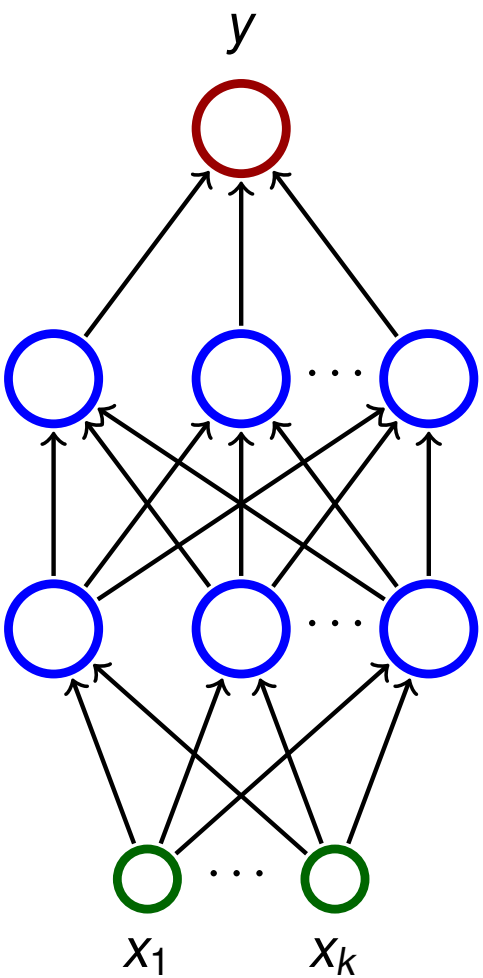
- Now let us connect all outputs of all neurons $N_{\vec{v}}$ satisfying $F(\vec{v}) = 1$ using a neuron implementing *OR*. □

Non-linear separation



- Consider a three layer network; each neuron has the unit step activation function.
- The network divides the input space in two subspaces according to the output (0 or 1).
  - The first (hidden) layer divides the input space into half-spaces.
  - The second layer may e.g. make intersections of the half-spaces $\Rightarrow$ convex sets.
  - The third layer may e.g. make unions of some convex sets.

Non-linear separation – illustration



- Consider three layer networks; each neuron has the unit step activation function.
- Three layer nets are capable of "approximating" any "reasonable" subset A of the input space $\mathbb{R}^k$.
  - Cover A with hypercubes (in 2D squares, in 3D cubes, ...)
  - Each hypercube K can be separated using a two layer network N_K
    (i.e. a function computed by N_K gives 1 for points in K and 0 for the rest).
  - Finally, connect outputs of the nets N_K satisfying $K \cap A \neq \emptyset$ using a neuron implementing OR.

Non-linear separation - sigmoid

Theorem (Cybenko 1989 - informal version)

Let σ be a continuous function which is sigmoidal, i.e. satisfies

$$\sigma(x) = \begin{cases} 1 & \text{pro } x \to +\infty \\ 0 & \text{pro } x \to -\infty \end{cases}$$

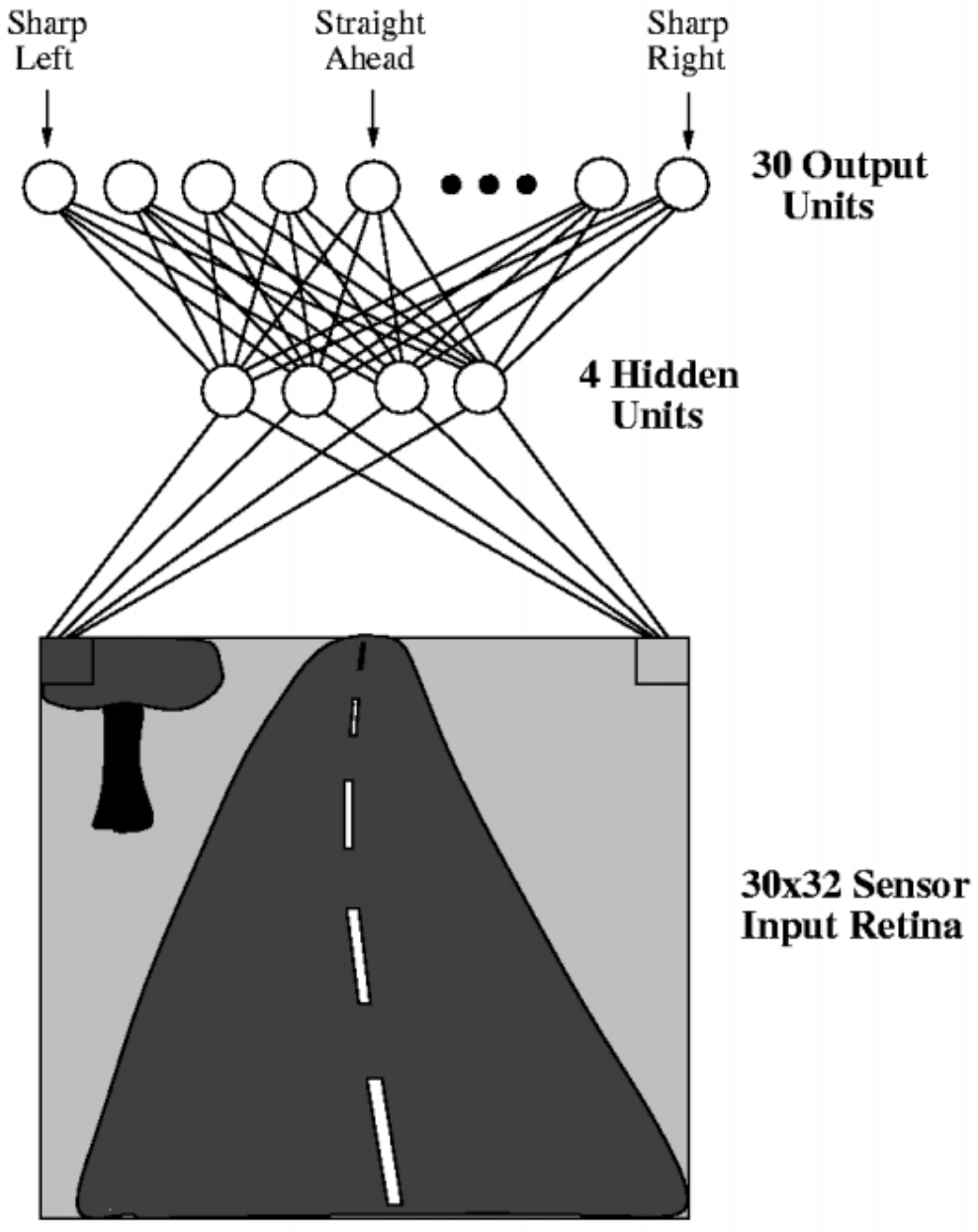
For every reasonable set $A \subseteq [0,1]^n$, there is a **two layer network** *where each hidden neuron has the activation function σ (output neurons are linear), that satisfies the following:*

For most vectors $\vec{v} \in [0,1]^n$ we have that $\vec{v} \in A$ iff the network output is > 0 for the input $\vec{v}$.

For mathematically oriented:

- "reasonable" means Lebesgue measurable
- "most" means that the set of incorrectly classified vectors has the Lebesgue measure smaller than a given $\varepsilon > 0$

Non-linear separation - practical illustration



- ALVINN drives a car
- The net has $30 \times 32 = 960$ inputs (the input space is thus $\mathbb{R}^{960}$)
- Input values correspond to shades of gray of pixels.
- Output neurons "classify" images of the road based on their "curvature".

Zdroj obrázku: `http://jmvidal.cse.sc.edu/talks/ann/alvin.html`

Function approximation - three layers

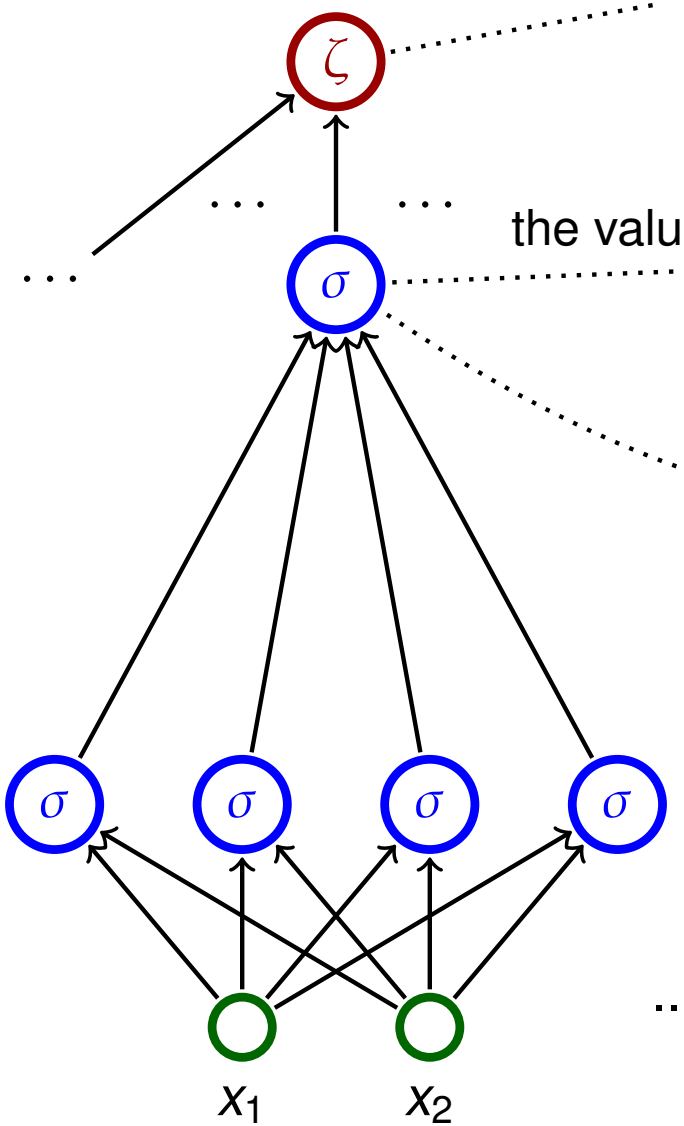
Let σ be a logistic sigmoid, i.e.

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}}$$

For every continuous function $f : [0,1]^n \to [0,1]$ and $\varepsilon > 0$ there is a three-layer network computing a function $F : [0,1]^n \to [0,1]$ such that

- there is a linear activation in the output layer, i.e. the value of the output neuron is its inner potential ξ,
- the remaining neurons have the logistic sigmoid σ as their activation,
- for every $\vec{v} \in [0,1]^n$ we have that $|F(\vec{v}) - f(\vec{v})| < \varepsilon$.

Function approximation – three layer networks



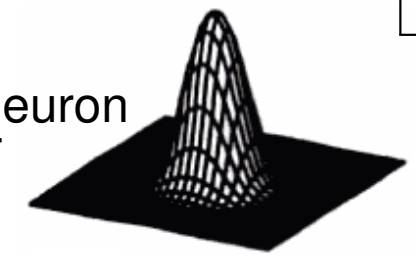
weighted sum of "spikes"

$$\sigma(\xi) = \frac{1}{1+e^{-\xi}}$$

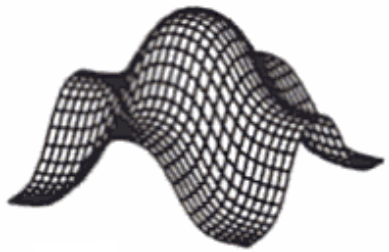
$$\zeta(\xi) = \xi$$

the value of the neuron

..... a "spike"



inner potential



... + the other two 90 degree rotations

Function approximation - two-layer networks

Theorem (Cybenko 1989)

Let σ be a continuous function which is sigmoidal, i.e. is increasing and satisfies

$$\sigma(x) = \begin{cases} 1 & \text{pro } x \to +\infty \\ 0 & \text{pro } x \to -\infty \end{cases}$$

For every continuous function $f : [0,1]^n \to [0,1]$ and every $\varepsilon > 0$ there is a function $F : [0,1]^n \to [0,1]$ computed by a **two layer network** *where each hidden neuron has the activation function σ (output neurons are linear), that satisfies the following*

$$|f(\vec{v}) - F(\vec{v})| < \varepsilon \qquad \text{pro každé } \vec{v} \in [0,1]^n.$$

Neural networks and computability

- Consider recurrent networks (i.e. containing cycles)
  - with real weights (in general);
  - one input neuron and one output neuron (the network computes a function $F : A \to \mathbb{R}$ where $A \subseteq \mathbb{R}$ contains all inputs on which the network stops);
  - parallel activity rule (output values of all neurons are recomputed in every step);
  - activation function

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0\,; \\ \xi & 0 \leq \xi \leq 1\,; \\ 0 & \xi < 0. \end{cases}$$

- We encode words $\omega \in \{0,1\}^+$ into numbers as follows:

$$\delta(\omega) = \sum_{i=1}^{|\omega|} \frac{\omega(i)}{2^i} + \frac{1}{2^{|\omega|+1}}$$

E.g. $\omega = 11001$ gives $\delta(\omega) = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^5} + \frac{1}{2^6}$ ($= 0.110011$ in binary form).

Neural networks and computability

A network **recognizes** a language $L \subseteq \{0,1\}^+$ if it computes a function $F : A \to \mathbb{R}$ ($A \subseteq \mathbb{R}$) such that

$$\omega \in L \text{ iff } \delta(\omega) \in A \text{ and } F(\delta(\omega)) > 0.$$

- Recurrent networks with rational weights are equivalent to Turing machines
  - For every recursively enumerable language $L \subseteq \{0,1\}^+$ there is a recurrent network with rational weights and less than 1000 neurons, which recognizes L.
  - The halting problem is undecidable for networks with at least 25 neurons and rational weights.
  - There is "universal" network (equivalent of the universal Turing machine)
- Recurrent networks are super-Turing powerful
  - For **every** language $L \subseteq \{0,1\}^+$ there is a recurrent network with less than 1000 nerons which recognizes L.

Summary of theoretical results

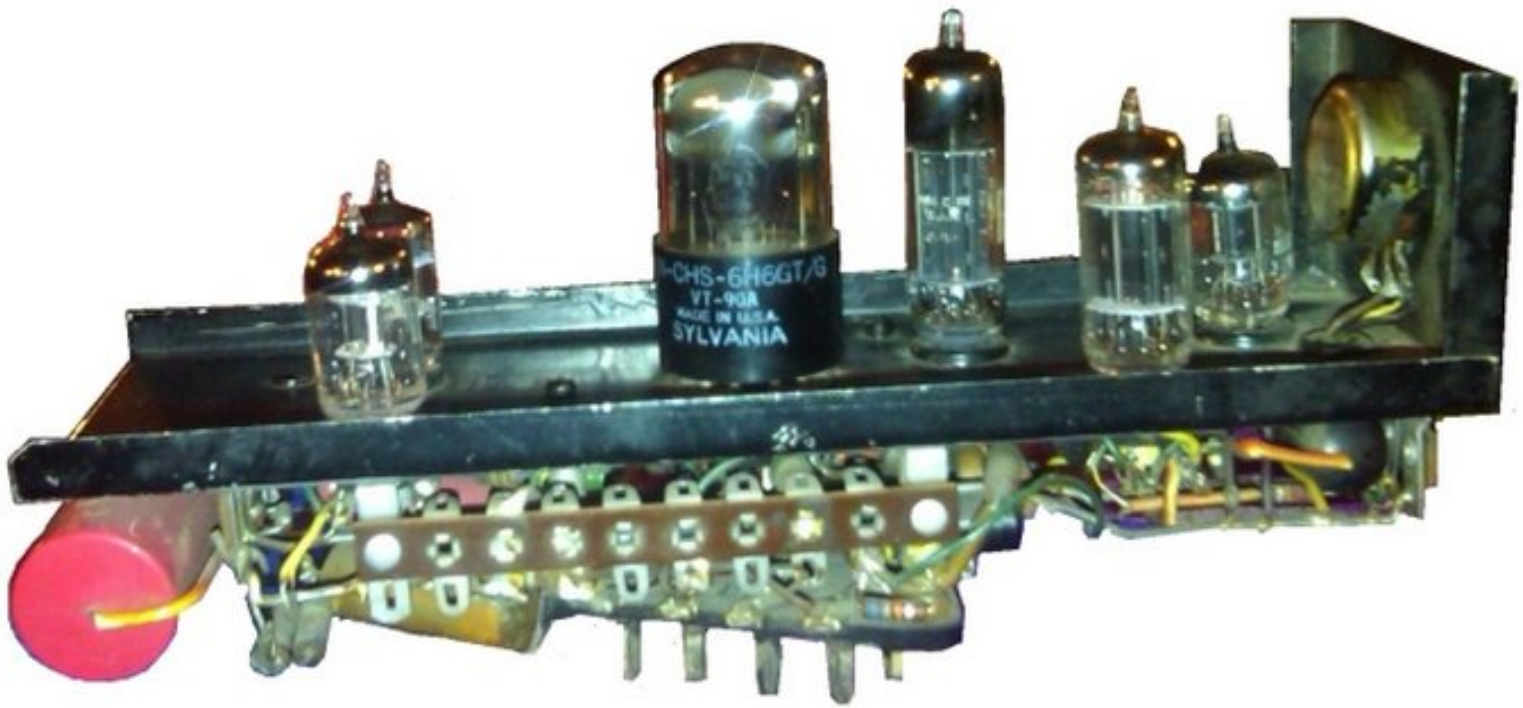
- Neural networks are very strong from the point of view of theory:
  - All Boolean functions can be expressed using two-layer networks.
  - Two-layer networks may approximate any continuous function.
  - Recurrent networks are at least as strong as Turing machines.
- These results are purely theoretical!
  - "Theoretical" networks are extremely huge.
  - It is very difficult to handcraft them even for simplest problems.
- From practical point of view, the most important advantage of neural networks are: learning, generalization, robustness.

Neural networks vs classical computers

	Neural networks	Classical computers
Data	implicitly in weights	explicitly
Computation	naturally parallel	sequential, localized
Robustness	robust w.r.t. input corruption & damage	changing one bit may completely crash the computation
Precision	imprecise, network recalls a training example "similar" to the input	(typically) precise
Programming	learning	manual

History of neurocomputers

- 1951: SNARC (Minski et al)
  - the first implementation of neural network
  - a rat strives to exit a maze
  - 40 artificial neurons (300 vacuum tubes, engines, etc.)



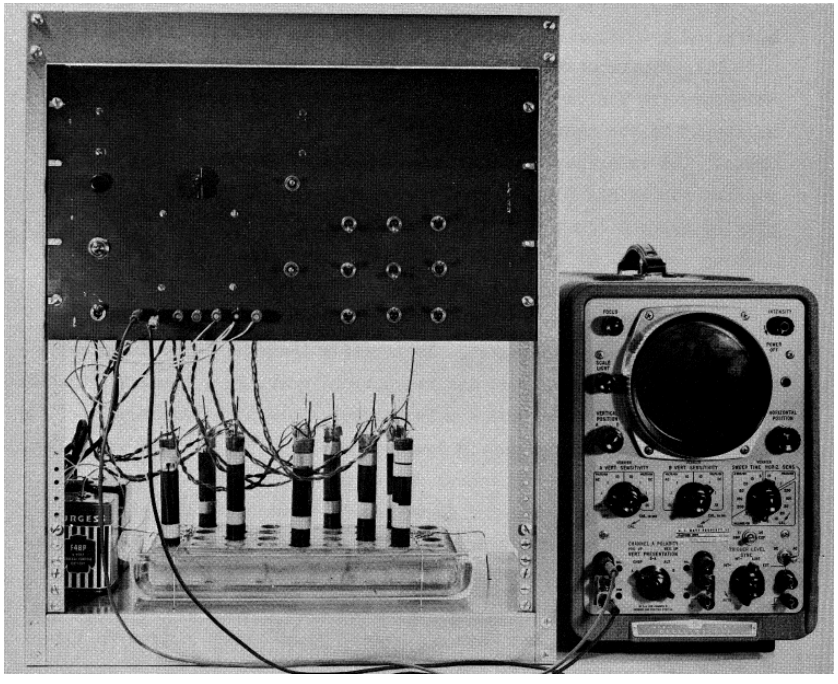
History of neurocomputers

- 1957: Mark I Perceptron (Rosenblatt et al) - the first successful network for image recognition

  - single layer network
  - image represented by 20×20 photocells
  - intensity of pixels was treated as the input to a perceptron (basically the formal neuron), which recognized figures
  - weights were implemented using potentiometers, each set by its own engine
  - it was possible to arbitrarily reconnect inputs to neurons to demonstrate adaptability

History of neurocomputers

- 1960: ADALINE (Widrow & Hof)



  - single layer neural network
  - weights stored in a newly invented electronic component **memistor**, which remembers history of electric current in the form of resistance.
  - Widrow founded a company Memistor Corporation, which sold implementations of neural networks.
- 1960-66: several companies concerned with neural networks were founded.

History of neurocomputers

- 1967-82: dead still after publication of a book by Minski & Papert (published 1969, title *Perceptrons*)
- 1983-end of 90s: revival of neural networks
  - many attempts at hardware implementations
    - application specific chips (ASIC)
    - programmable hardware (FPGA)
  - hw implementations typically not better than "software" implementations on universal computers (problems with weight storage, size, speed, cost of production etc.)
- end of 90s-cca 2005: NN suppressed by other machine learning methods (support vector machines (SVM))
- 2006-now: The boom of neural networks!
  - deep networks – often better than any other method
  - GPU implementations
  - ... some specialized hw implementations (not yet widespread)

History in waves ...

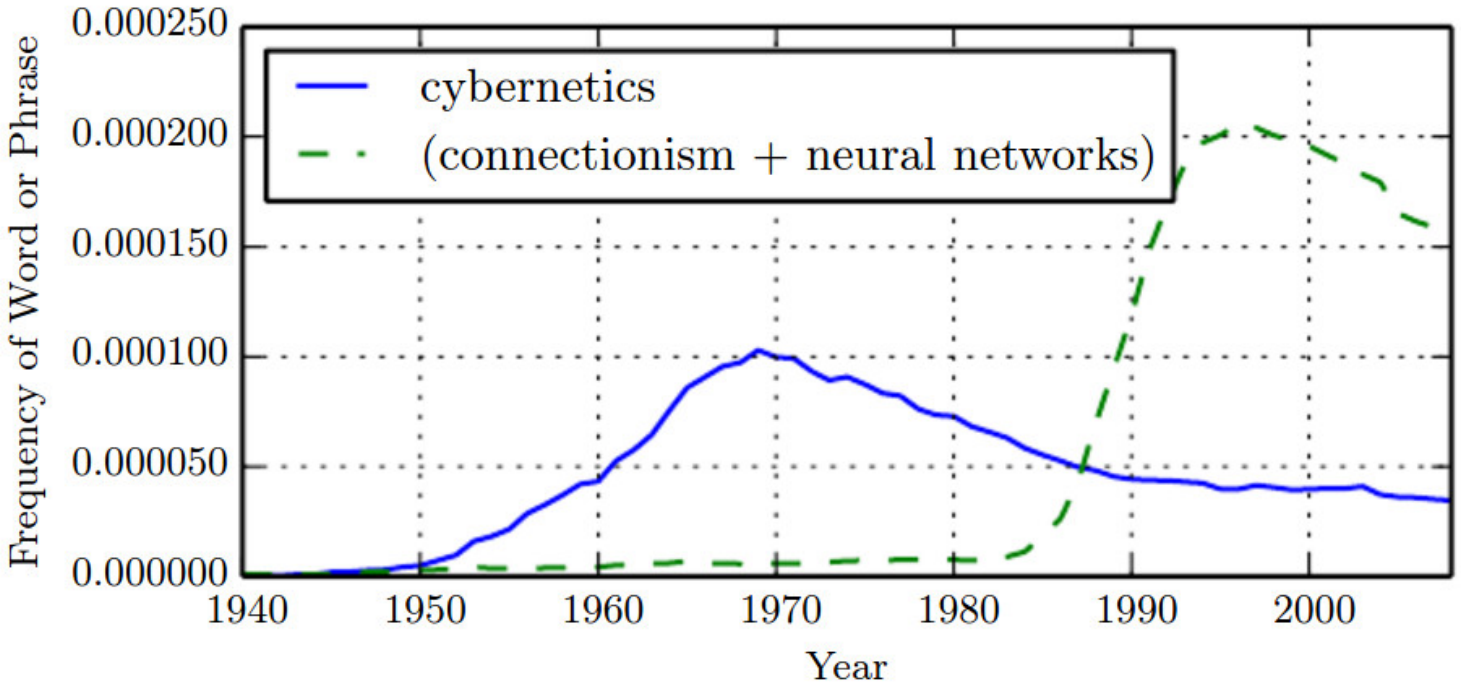
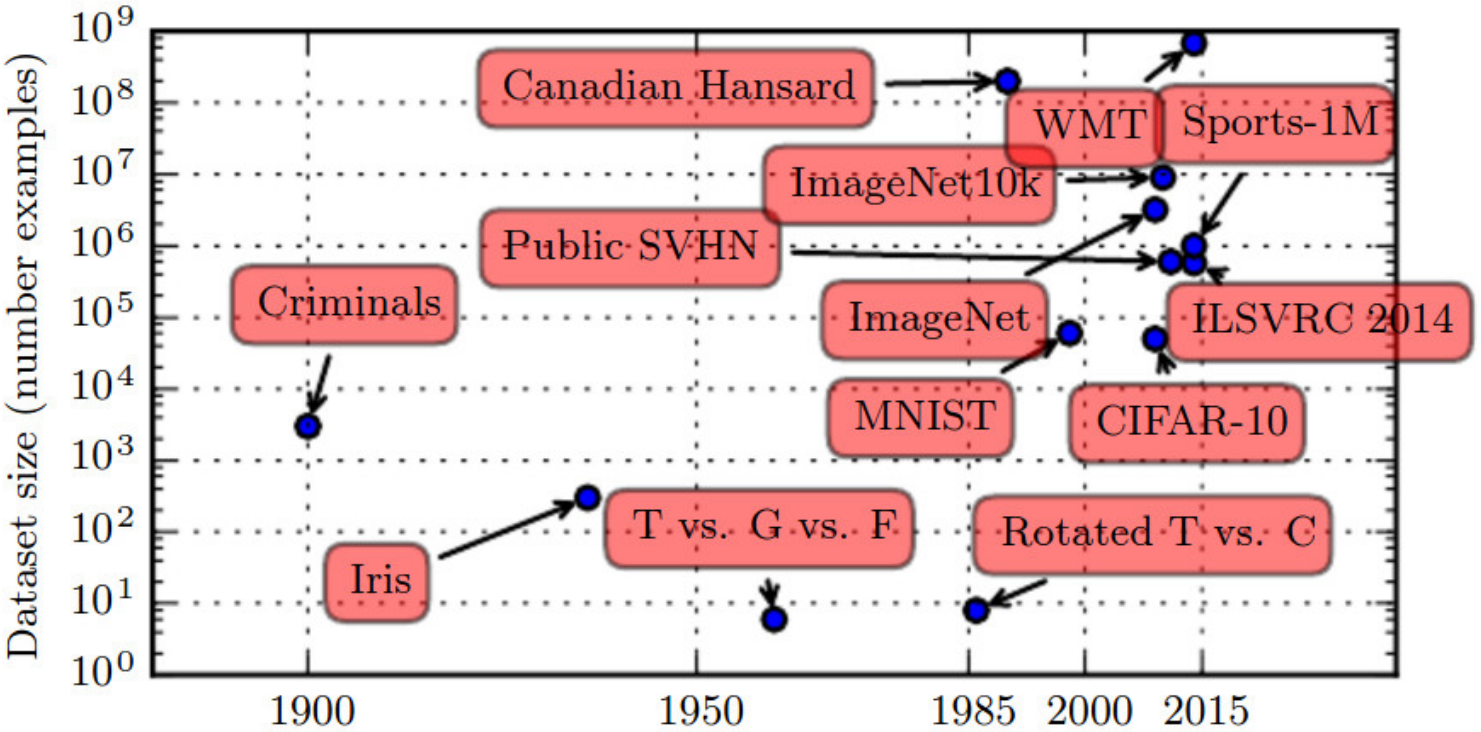


Figure: The figure shows two of the three historical waves of articial neural nets research, as measured by the frequency of the phrases “cybernetics” and “connectionism” or “neural networks” according to Google Books (the third wave is too recent to appear).

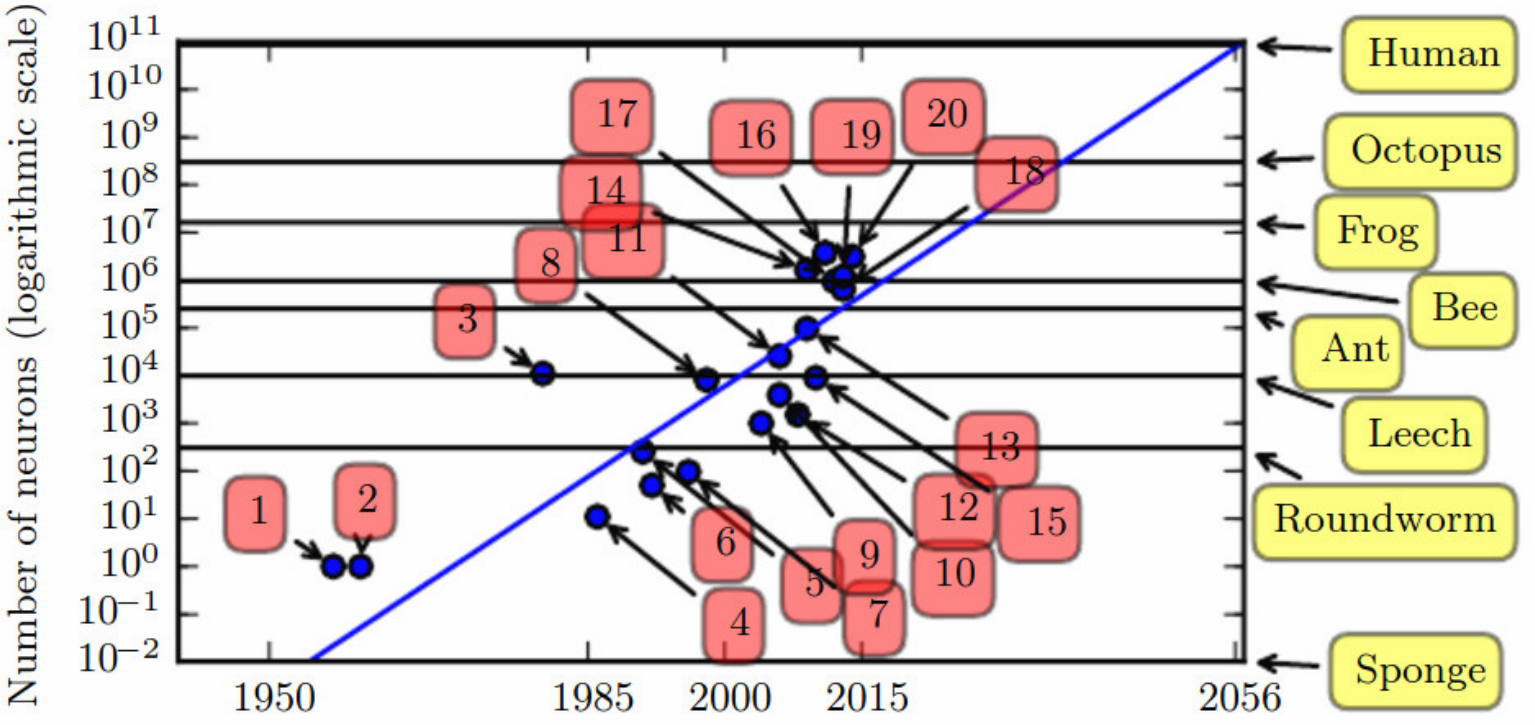
Current hardware – What do we face?

Increasing dataset size ...



Current hardware – What do we face?

... and thus increasing size of neural networks ...



2. ADALINE
4. Early back-propagation network (Rumelhart et al., 1986b)
8. Image recognition: LeNet-5 (LeCun et al., 1998b)
10. Dimensionality reduction: Deep belief network (Hinton et al., 2006)
    ... here the third "wave" of neural networks started
15. Digit recognition: GPU-accelerated multilayer perceptron (Ciresan et al., 2010)
18. Image recognition (AlexNet): Multi-GPU convolutional network (Krizhevsky et al., 2012)
20. Image recognition: GoogLeNet (Szegedy et al., 2014a)

Current hardware – What do we face?

... as a reward we get this ...

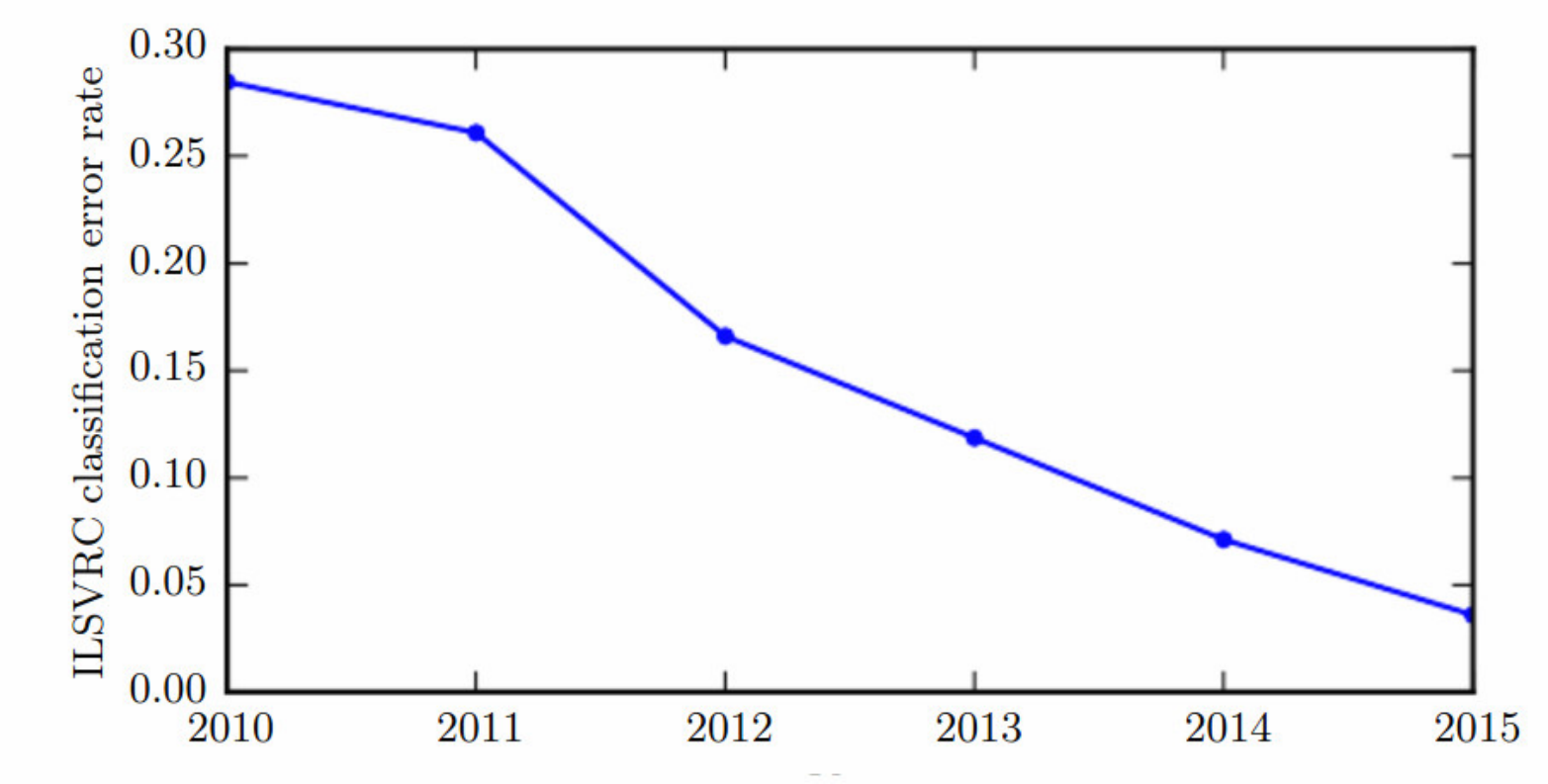


Figure: Since deep networks reached the scale necessary to compete in the ImageNetLarge Scale Visual Recognition Challenge, they have consistently won the competition every year, and yielded lower and lower error rates each time. Data from Russakovsky et al. (2014b) and He et al. (2015).

Current hardware

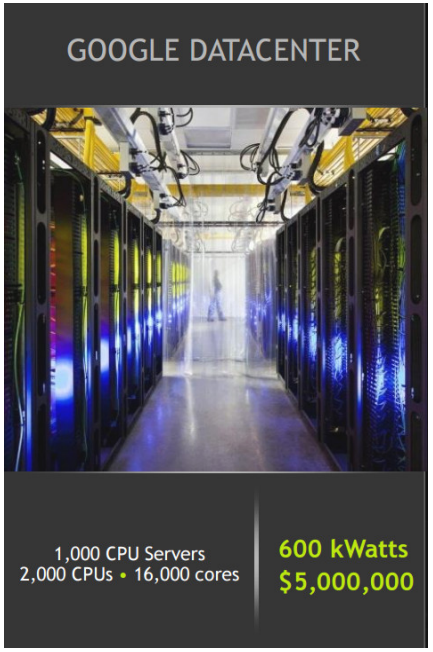
In 2012, Google trained a large network of 1.7 billion weights and 9 layers

The task was image recognition (10 million youtube video frames)

The hw comprised a 1000 computer network (16 000 cores), computation took three days.

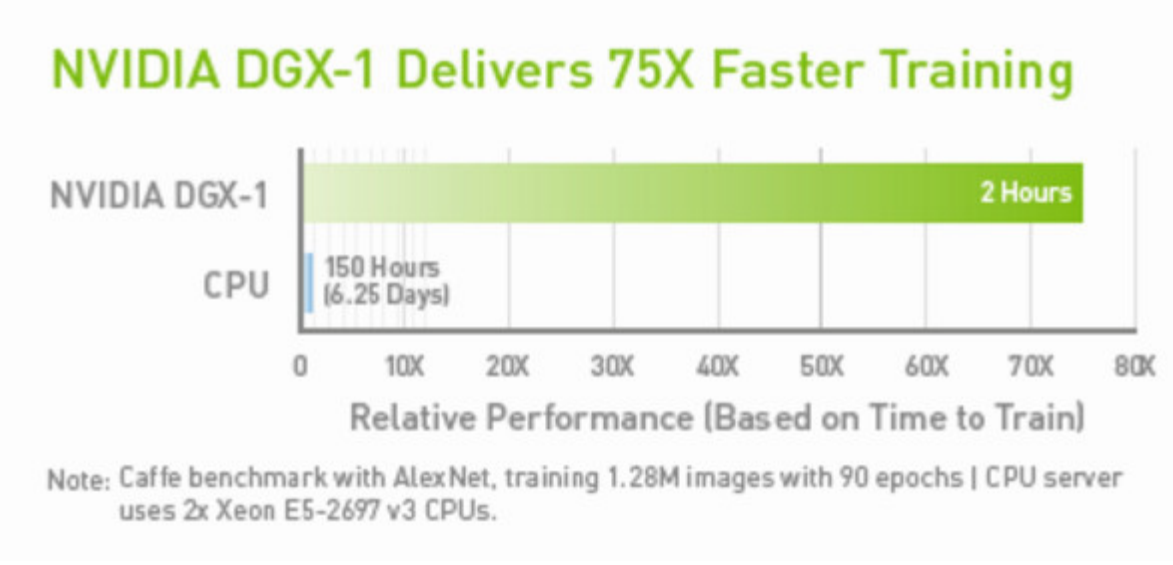
In 2014, similar task performed on Commodity Off-The-Shelf High Performance Computing (COTS HPC) technology: a cluster of GPU servers with Infiniband interconnects and MPI.

Able to train 1 billion parameter networks on just 3 machines in a couple of days.
Able to scale to 11 billion weights (approx. 6.5 times larger than the Google model) on 16 GPUs.



Current hardware – NVIDIA DGX-1 (example)

- 8x GPU (Tesla GP100)
- TFLOPS = 170
- GPU memory 16GB per GPU
- NVIDIA CUDA Cores: 28672
- System memory: 512 GB
- Network: Dual 10 GbE
- NVIDIA Deep Learning SDK



Current software

- **TensorFlow** (Google)
  - open source software library for numerical computation using data flow graphs
  - allows implementation of most current neural networks
  - allows computation on multiple devices (CPUs, GPUs, ...)
  - Python API
  - **Keras**: a library on top of TensorFlow that allows easy description of most modern neural networks
- **CNTK** (Microsoft)
  - functionality similar to TensorFlow
  - special input language called BrainScript
- **Theano**:
  - The "academic" grand-daddy of deep-learning frameworks, written in Python. Strongly inspired TensorFlow (some people developing Theano moved on to develop TensorFlow).
- There are others: Caffe, Torch (Facebook), Deeplearning4j, ...

Current software – Keras

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.optimizers import SGD

model = Sequential()
# Dense(64) is a fully-connected layer with 64 hidden units.
# in the first layer, you must specify the expected input data shape
# here, 20-dimensional vectors.
model.add(Dense(64, input_dim=20, init='uniform'))
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(64, init='uniform'))
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(10, init='uniform'))
model.add(Activation('softmax'))

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy',
              optimizer=sgd,
              metrics=['accuracy'])

model.fit(X_train, y_train,
          nb_epoch=20,
          batch_size=16)
score = model.evaluate(X_test, y_test, batch_size=16)
```

Other software implementations

Most "mathematical" software packages contain some support of neural networks:

- MATLAB
- R
- STATISTICA
- Weka
- ...

The implementations are typically not on par with the previously mentioned dedicated deep-learning libraries.

SyNAPSE (USA)

- Big research project, partially funded by DARPA
- Among the main subjects IBM a HRL, collaboraton with top US universities, e.g. Boston, Stanford
- The project started in 2008
- Invested tens of millions USD.

The goal

- Develop a neural network comparable with a real brain of a mammal
- The resulting hw chip should simulate 10 billion neurons, 100 trillion synaptic connections, consume 1 kilowatt ($\sim$ a small heater), size 2 dm^3
- Oriented towards development of a new parallel computer architecture rather than neuroscience.

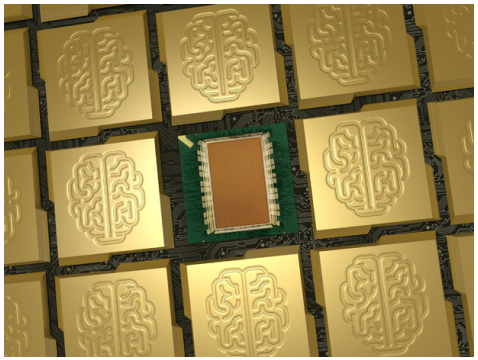
SyNAPSE (USA) – some results

A cat brain simulation (2009)

- A simulation of a network with 10^9 neurons, 10^{13} synapses
- Simulated on a supercomputer Dawn (Blue Gene/P), 147,450 CPU, 144 tB of memory
- 643 times slower than the real brain
- The network modelled according to the real-brain structure (hierarchical model of a visual cortex, 4 layers)
- The authors claim that they observed some behaviour similar to the behaviour of the real brain (signal propagation, α, γ waves)

... simulation was heavily criticised (see latter)
... in 2012 the number of neurons increased to 530 bn neurons a 100 tn synapses

SyNAPSE (USA) – TrueNorth



- A chip with 5.4 billion elements
- 4096 neurosynaptic cores connected by a network, implementing 1 million programmable "spiking" neurons, 256 million programmable synaptic connections
- global frequency 1-kHz
- low energy consumption, approx. 63mW

- Offline learning, implemented some known algorithms (convolutional networks, RBM etc.)
- Applied to simple image recognition tasks.

Human Brain Project, HBP (Europe)

- Funded by EU, budget 10^9 EUR for 10 years
- Successor of Blue Brain Project at EPFL Lausanne
- Blue Brain started in 2005, ended in 2012, Human Brain Project started in 2013

The original goal: Deeper understanding of human brain

- **networking in neuroscience**
- diagnosis of brain diseases
- thinking machine

The approach:

- study of brain tissue using current technology
- modelling of biological neurons
- simulation of the models (program NEURON)

HBP (Europe

Blue brain project (2008)

- Model of a part of the brain cortex of a rat (approx. 10,000 neurons), much more complex model of neurons than in SyNAPSE
- Simulated on a supercomputer of the type Blue Gene/P (provided by IBM on discount), 16,384 CPU, 56 teraflops, 16 terabytů paměti, 1 PB disk space
- Simulation 300x slower than the real brain

Human brain project (2015):

- Simplified model of the nervous system of a rat (approx. 200 000 neurons)

SyNAPSE vs HBP

2011: IBM Simulates 4.5 percent of the Human Brain, and All of the Cat Brain (Scientific American)

"... performed the first near real-time cortical simulation of the brain that exceeds the scale of a cat cortex" (IBM)

This announcement has been heavily criticised by Dr. Markram (head of HBP)

"This is a mega public relations stunt – a clear case of scientific deception of the public"

"Their so called "neurons" are the tiniest of points you can imagine, a microscopic dot"

"Neurons contain 10's of thousands of proteins that form a network with 10's of millions of interactions. These interactions are incredibly complex and will require solving millions of differential equations. They have none of that."

SyNAPSE vs HBP

"Eugene Izhikevik himself already in 2005 ran a simulation with 100 billion such points interacting just for the fun of it: (over 60 times larger than Modha's simulation)"

Why did they get the Gordon Bell Prize?
"They seem to have been very successful in influencing the committee with their claim, which technically is not peer-reviewed by the respective community and is neuroscientifically outrageous."

But is there any innovation here?
"The only innovation here is that IBM has built a large supercomputer."

But did Mohda not collaborate with neuroscientists?
"I would be very surprised if any neuroscientists that he may have had in his DARPA consortium realized he was going to make such an outrageous claim. I can't imagine that the San Fransisco neuroscientists knew he was going to make such a stupid claim. Modha himself is a software engineer with no knowledge of the brain."

... and in the meantime in Europe

In 2014, the European Commission received an open letter signed by more than 130 heads of laboratories demanding a substantial change in the management of the whole project.

Peter Dayan, director of the computational neuroscience unit at UCL:

“The main apparent goal of building the capacity to construct a larger-scale simulation of the human brain is radically premature.”

“We are left with a project that can’t but fail from a scientific perspective. It is a waste of money, it will suck out funds from valuable neuroscience research, and would leave the public, who fund this work, justifiably upset.”

... and in 2016

The European Commission and the Human Brain Project Coordinator, the École Polytechnique Fédérale de Lausanne (EPFL), have signed the first Specific Grant Agreement (SGA1), releasing EUR 89 million in funding retroactively from 1st April 2016 until the end of March 2018. The signature of SGA1 means that the HBP and the European Commision have agreed on the HBP Work Plan for this two year period.

The SGA1 work plan will move the Project closer to achieving its aim of establishing a cutting-edge, ICT-based scientific Research Infrastructure for brain research, cognitive neuroscience and brain-inspired computing.