What is Deep Learning

The Big Picture – From History to Todays Implementations

Daniel Schalk

October 22, 2018

History of Deep Learning

Deep Learning Timeline - 1

1943

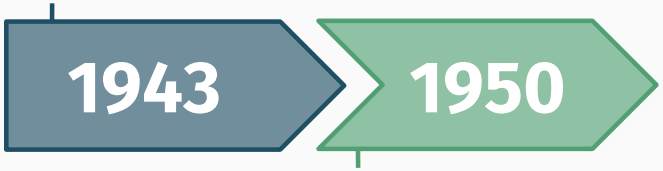
Warren S. McCulloch and **Walter Pitts** published tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together.

1943

Deep Learning Timeline - 1

1943

Warren S. McCulloch and **Walter Pitts** published tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together.



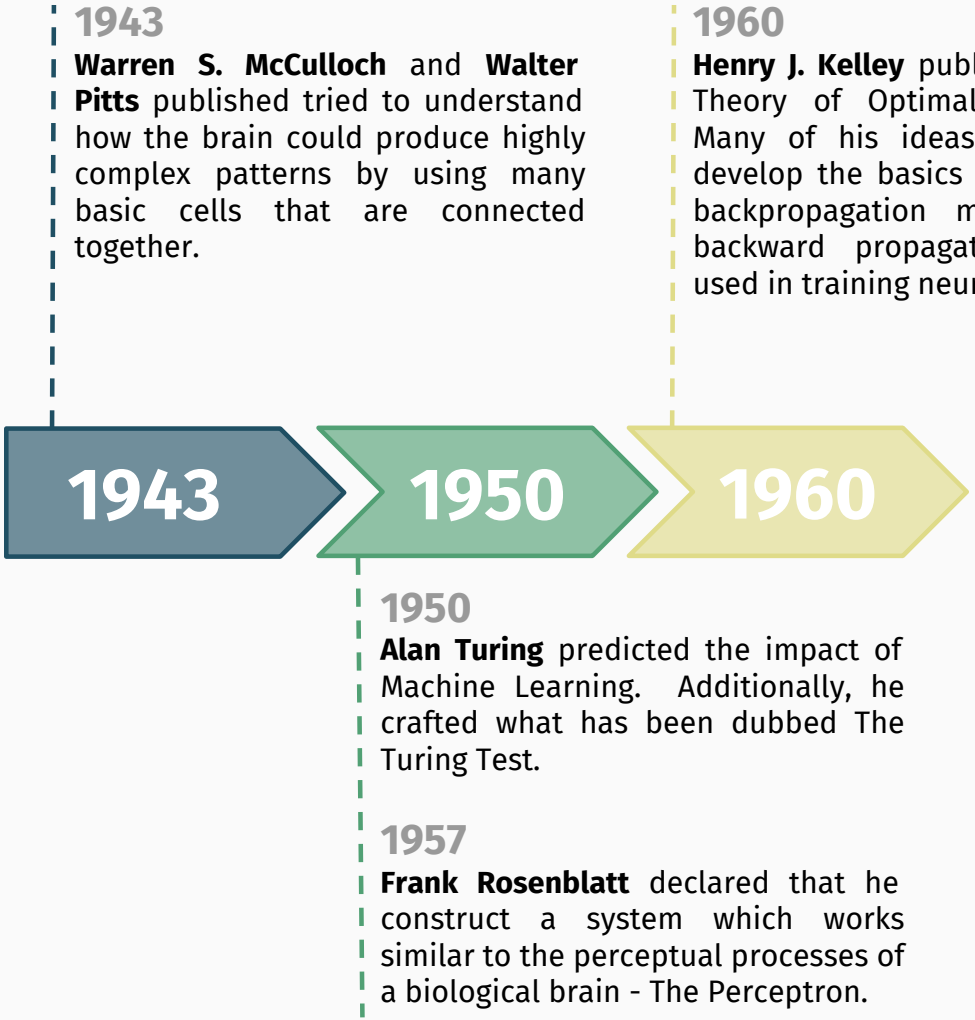
1950

Alan Turing predicted the impact of Machine Learning. Additionally, he crafted what has been dubbed The Turing Test.

1957

Frank Rosenblatt declared that he construct a system which works similar to the perceptual processes of a biological brain - The Perceptron.

Deep Learning Timeline - 1



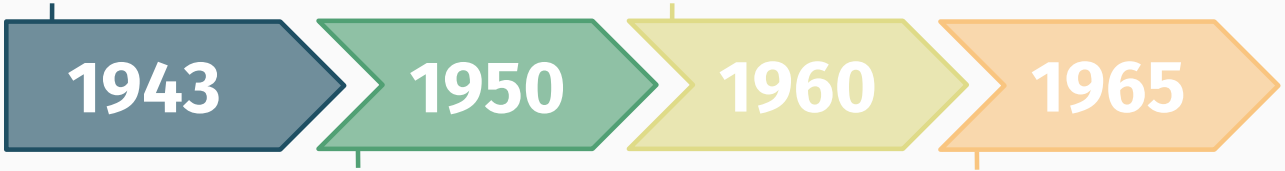
Deep Learning Timeline - 1

1943

Warren S. McCulloch and **Walter Pitts** published tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together.

1960

Henry J. Kelley published "Gradient Theory of Optimal Flight Paths". Many of his ideas were used to develop the basics of a continuous backpropagation model (aka the backward propagation of errors) used in training neural networks.



1950

Alan Turing predicted the impact of Machine Learning. Additionally, he crafted what has been dubbed The Turing Test.

1957

Frank Rosenblatt declared that he construct a system which works similar to the perceptual processes of a biological brain - The Perceptron.

1965

Alexey Ivakhnenko and **V. G. Lapa** created the first *working* deep learning networks, applying what had been only theories and ideas up to that point.

Ivakhnenko developed a learning algorithm using deep feedforward multilayer perceptrons. For that reason alone, many consider Ivakhnenko the father of modern deep learning.

Deep Learning Timeline - 1

1943
Warren S. McCulloch and **Walter Pitts** published tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together.

1950
Alan Turing predicted the impact of Machine Learning. Additionally, he crafted what has been dubbed The Turing Test.

1957
Frank Rosenblatt declared that he construct a system which works similar to the perceptual processes of a biological brain - The Perceptron.

1960
Henry J. Kelley published "Gradient Theory of Optimal Flight Paths". Many of his ideas were used to develop the basics of a continuous backpropagation model (aka the backward propagation of errors) used in training neural networks.

1965
Alexey Ivakhnenko and **V. G. Lapa** created the first *working* deep learning networks, applying what had been only theories and ideas up to that point.

Ivakhnenko developed a learning algorithm using deep feedforward multilayer perceptrons. For that reason alone, many consider Ivakhnenko the father of modern deep learning.

1970 - First AI Winter
AI was subject to critiques and financial setbacks. AI researchers had failed to appreciate the difficulty of the problems they faced.

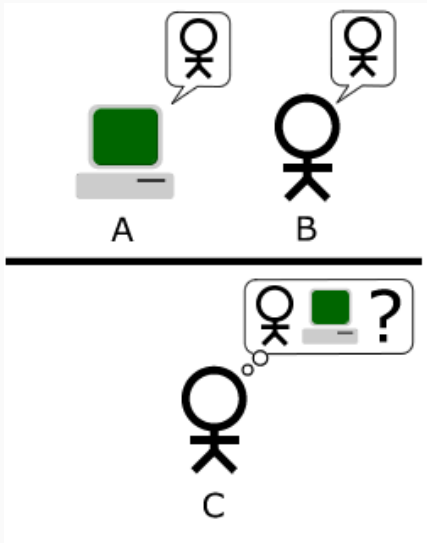
AI was claimed to only be suitable for solving "toy" versions.

Deep Learning Timeline - 2

Fascination Deep Learning

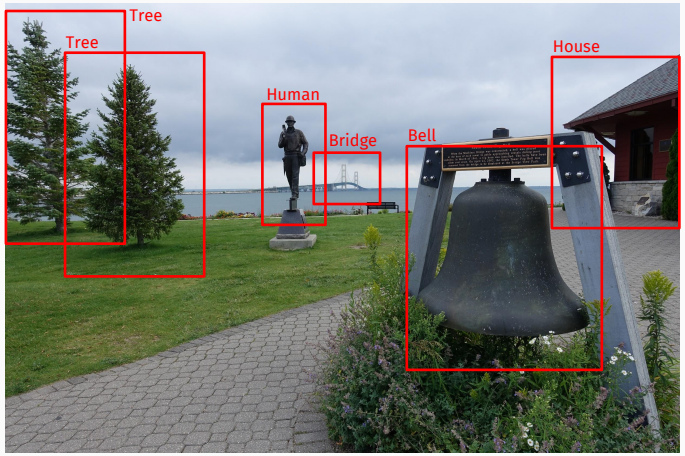
Imitating Humans - 1

- Turing test:
  - Developed by Alan Turing in 1950
  - Test of a machine's ability to exhibit intelligent behavior
  - Player C, the interrogator, is given the task of trying to determine which player, A or B, is a computer and which is a human

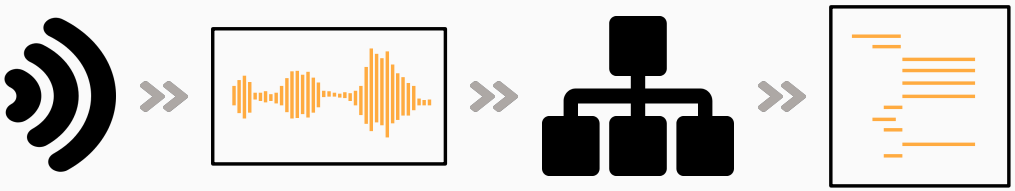


Imitating Humans - 2

- Image Recognition (Seeing):



- Speech Recognition and Text Mining (Hearing and understanding text):



Imitating Humans - 3

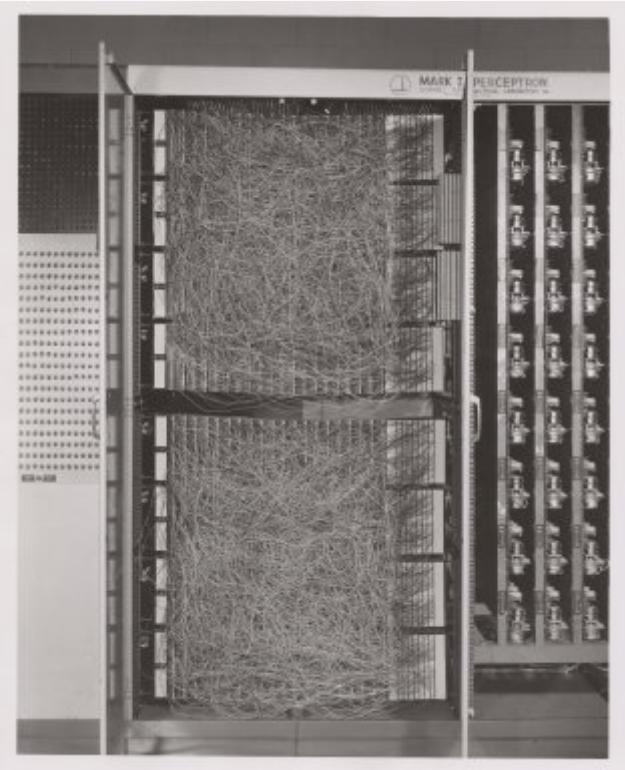
- And know we try to learn them being creative:
  - Music and Text Generation
  - Neural Style Transfer:

  - ...

Why Deep Learning is so Powerful?

The Perceptron

- The perceptron was invented by Frank Rosenblatt 1957.

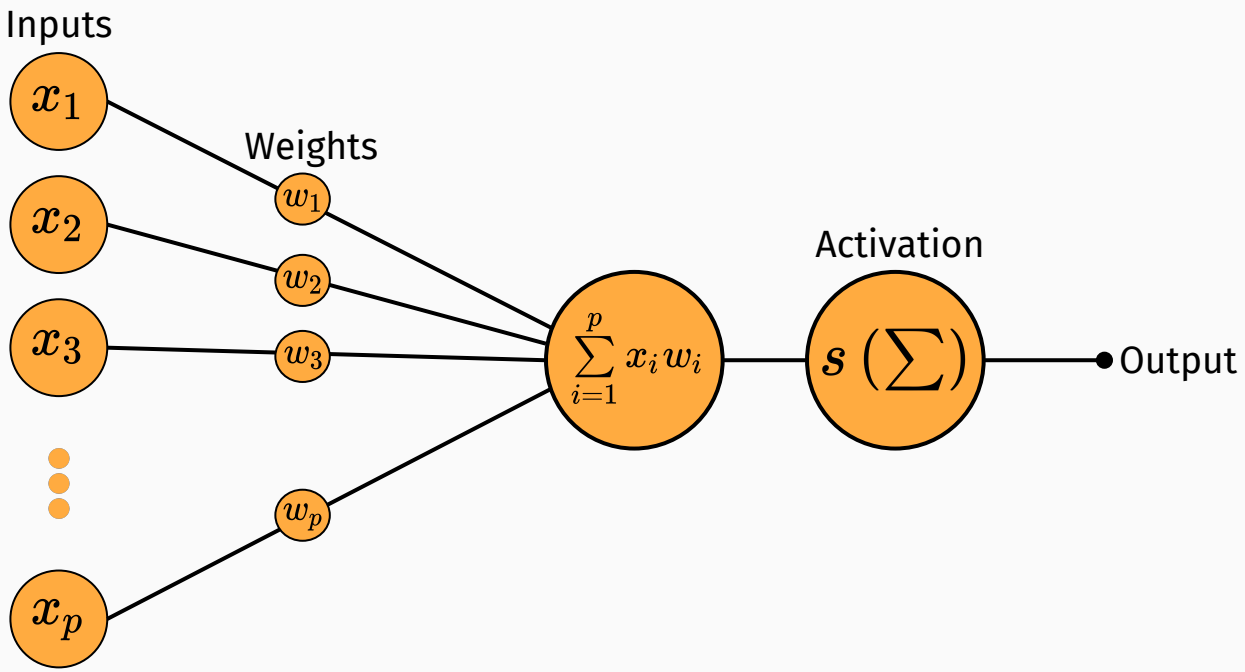


The Mark I Perceptron

- It is the basic computational unit for neural networks.

Singlelayer Perceptron

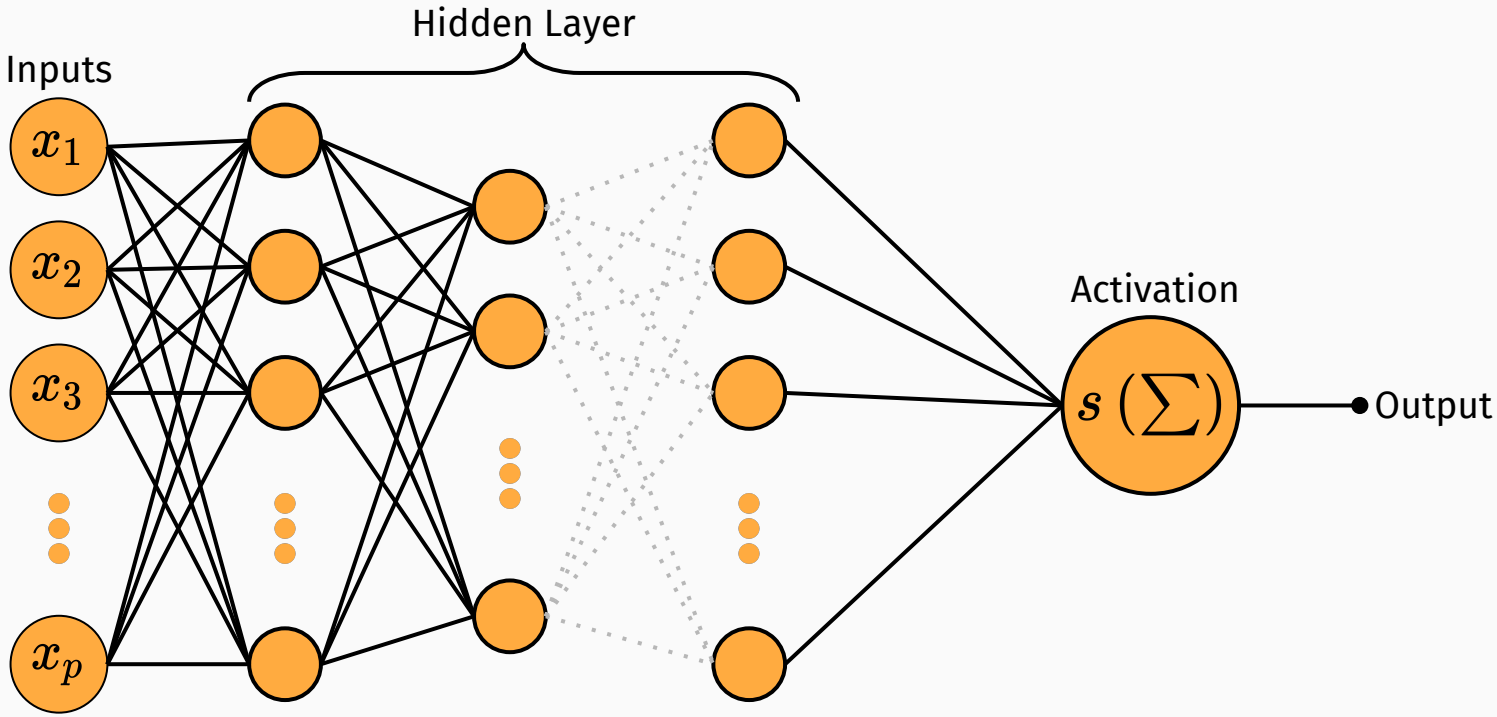
- Weighted sum of input values transformed by an activation function s
- If s is the sigmoid function $(1 + \exp \sum)^{-1}$, then the perceptron does exactly the same as the logistic regression



Multilayer Perceptron

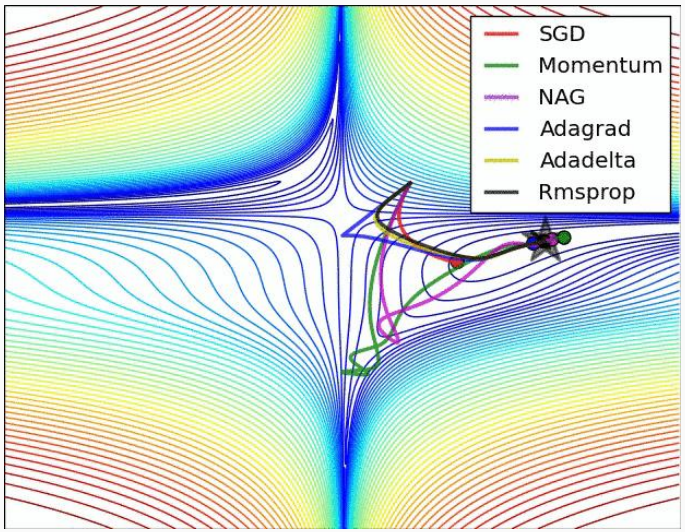
- Stacking of multiple perceptrons
- Corresponds to stacking GLM models
- Number of parameter grows very fast

  $\rightarrow$ Optimizing becomes more difficult



Optimizer

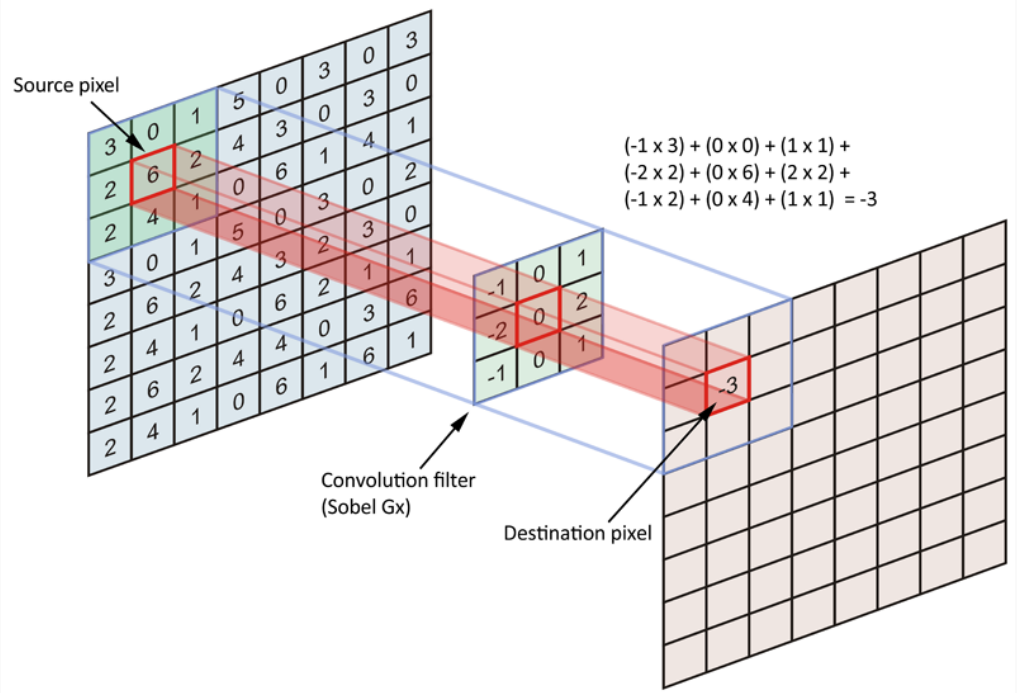
- Having that much parameter/weights to find, standard optimizer like Gradient Descent may fail
- Therefore, much effort was spend to get faster optimizer like momentum, adagrad, etc.:



Source: Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

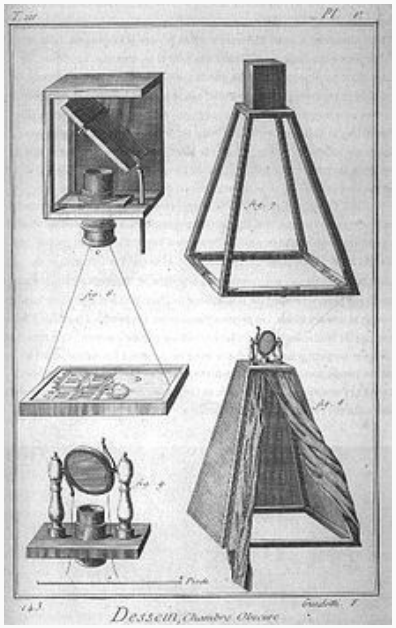
Convolution

- Generating of new, hopefully meaningful, features of the input (commonly images)



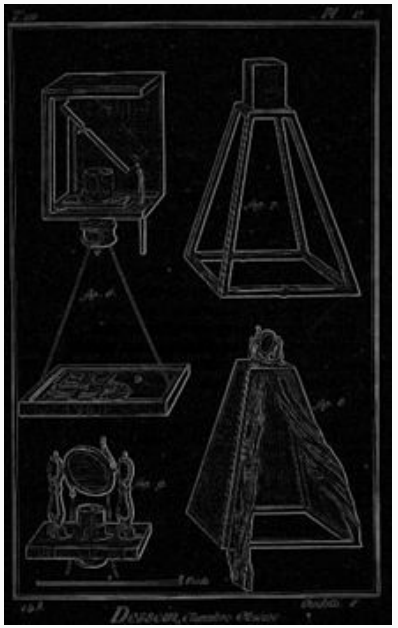
Convolution

Input Image



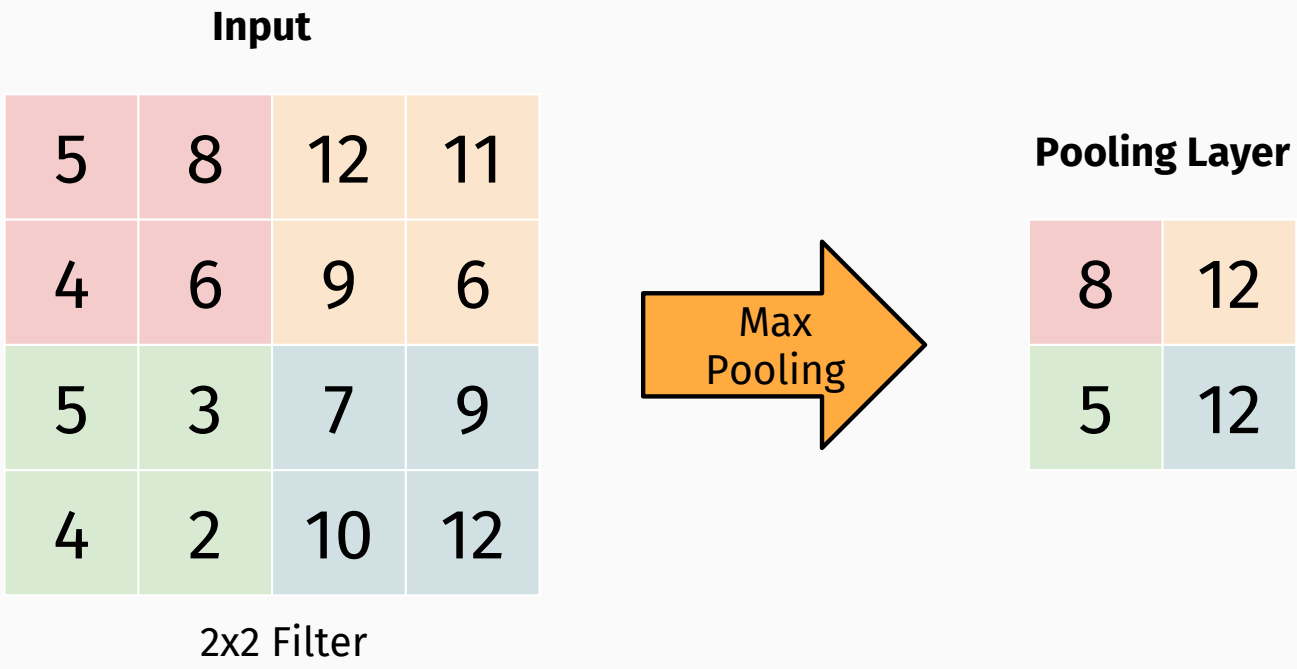
G_x

Generated Image/Feature



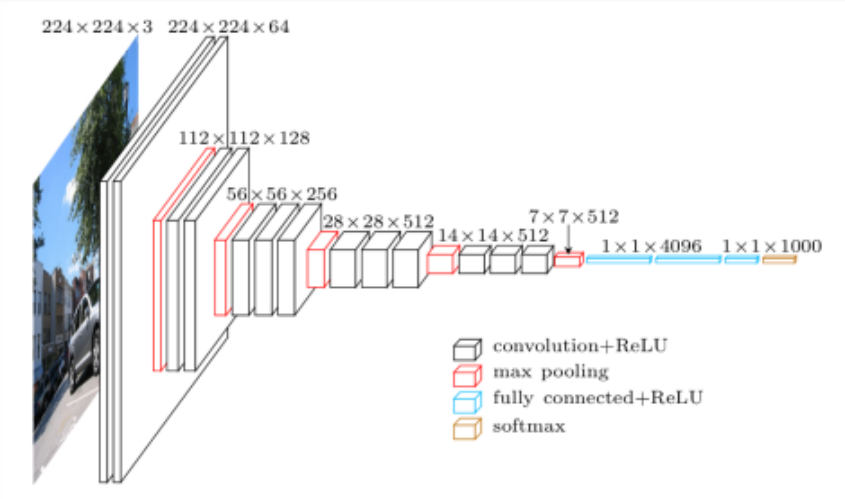
Pooling

- Down-sampling of images
- Reduces overfitting, memory, and therefore speeds up the fitting process



Lets Get Deep

- The secret of Deep Learning is the chaining of hidden layer such as convolution layers, pooling layers, and so on
- This deep structure allows the network to create powerful features and explore complex structures within the data
- VGG16 architecture:



Source: https://www.cs.toronto.edu/~frossard/post/vgg16/

Pre Trained Models

Model	Size	Parameters	Depth
Xception	88 MB	22,910,480	126
VGG16	528 MB	138,357,544	23
VGG19	549 MB	143,667,240	26
ResNet50	99 MB	25,636,712	168
InceptionV3	92 MB	23,851,784	159
InceptionResNetV2	215 MB	55,873,736	572
MobileNet	16 MB	4,253,864	88
MobileNetV2	14 MB	3,538,984	88
DenseNet121	33 MB	8,062,504	121
DenseNet169	57 MB	14,307,880	169
DenseNet201	80 MB	20,242,984	201
NASNetMobile	23 MB	5,326,716	-
NASNetLarge	343 MB	88,949,818	-

Source: Keras Documentation

Challenges in Deep Learning

Structure Search

$\rightarrow$ Transfer learning.

Expensive Training

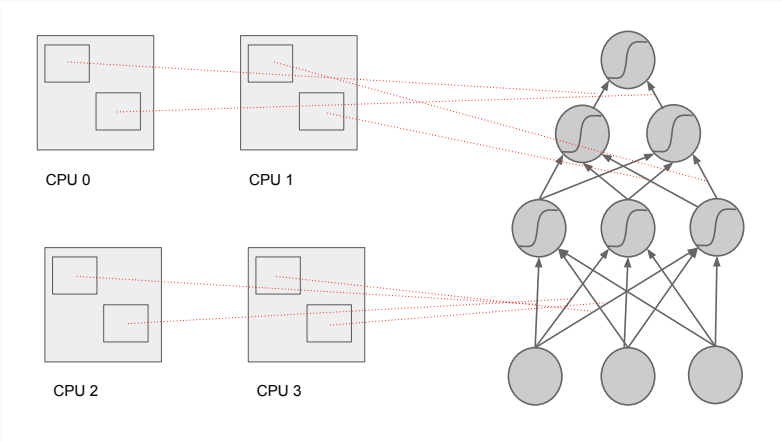
- Very very much parameter

$\rightarrow$ Use server or GPUs.

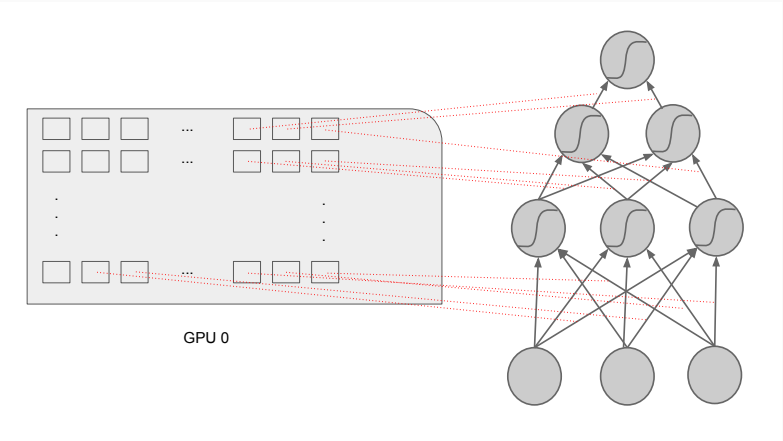
About Implementations

Hardware

- Deep Neural Networks require special hardware to be trained efficiently.
- The training is done using **G**raphics **P**rocessing **U**nits and a special programming language called **CUDA**.
- Training on standard CPUs takes a very long long time and gets infeasible for anything but toy examples.



Each CPU can do $2 - 8$ parallel computations.

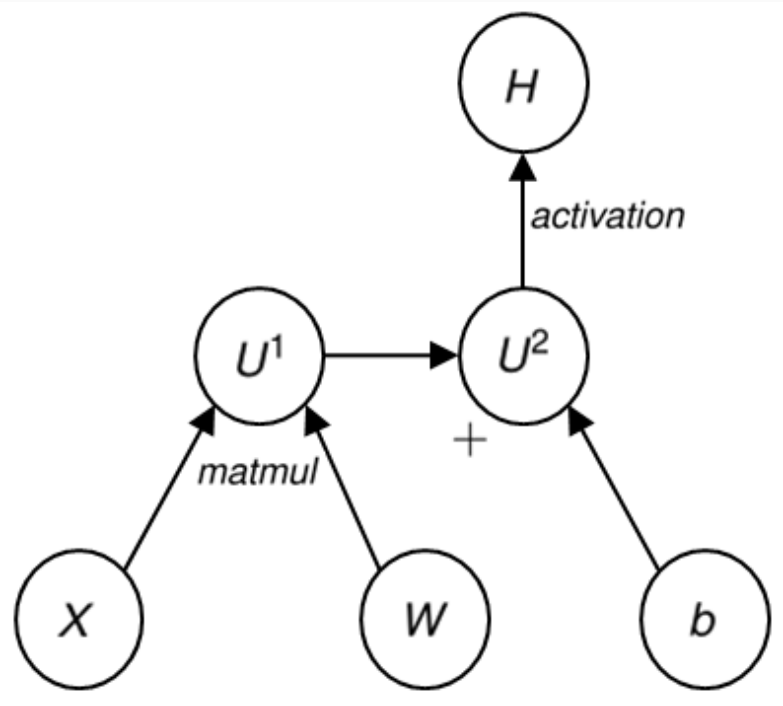


A single GPU can do thousands of simple parallel computations.

Software

CUDA is a very *low level* programming language and thus writing code for deeplearning requires a lot of work. Software projects, like TensorFlow and abstract CUDA provide additional functionality.

The basic concept of calculations in deep neural networks is a *computational graph*, which describes the dependency structure of the network.



Computational graph for $f(XW + b)$.

TensorFlow

- Open-source framework developed by google.
- Rather low-level and aimed to directly work with computational graphs.
- Mainly support for Python (R support only via *reticulate*).
- Widely used and well documented.

Keras

- Open-source high-level API for deeplearning.
- Can run on top of **TensorFlow**, CNTK or Theano.
- Mainly support for Python (R support only via *reticulate*).
- Widely used and well documented.

PyTorch

- Open-source framework developed by facebook.
- Reimplementation of Torch.
- Only support for Python.
- Widely used and well documented.

Fast.ai

- Open-source high-level API build on top of PyTorch.
- Still in alpha version.
- Only support for Python.
- Initially developed for the *Practical Deep Learning for Coders* online course.

mxnet

- Open-source framework in the Apache foundation.
- Scalable, allow easy training on multiple GPUs in parallel.
- Supports multiple languages (C++, Python, R, Julia, Matlab, JavaScript, Go, Scala, Pearl).
- Not as widely used as other frameworks.

Where to Start in the DL Jungle

Getting Started with Keras - Installation

Getting Started with Keras - Overview

- Instead of introducing theory fist, we want to get into the topic by applying it.
- We use examples from the book **Deep Learning with Python** which are prepared as **notebooks**.
- **But**: When using something new, e.g. a convolution layer or optimizer, try to understand what it does and why it might be beneficial!

Getting Started with Keras - First Neural Net

Explain API

Getting Started with Keras - First Neural Net

Some Code

Getting Started with Keras - Getting Deep

Explain API

Getting Started with Keras - Getting Deep

Some Code

Getting Started with Keras - Transfer Learning

Explain API

Getting Started with Keras - Transfer Learning

Some Code

Outlook

Getting More Complex

RNN, LSTM, GAN

NLP

Very very short intro how text mining connects to deep learning (gensim, word vectors, . . .)

Reinforcement Learning

This is what comes closest to AI as we are thinking of it. Just show examples