Deep Learning

CSE 634 **Data Mining**
Professor Anita Wasilewska

References:

- Intro to DL: https://www.nature.com/articles/nature14539
- CNN: http://hi.cs.stonybrook.edu/cse-527
- RNN: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf
- LSTM: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- BPTT: https://deeplearning4j.org/lstm.html
- GANs: https://deeplearning4j.org/deepautoencoder
- Autoencoders: https://deeplearning4j.org/generative-adversarial-network
- DeepMind Blog: https://deepmind.com/blog/alphago-zero-learning-scratch/
- Mastering the game of Go without human knowledge: https://www.nature.com/articles/nature24270.pdf

Overview

- Introduction - What is it all about?
- DL vs ML
- Importance and Applications

Presented by **Komal Gyanani**:

- CNNs

Presented by **Swetha Tatavarthy**:

- RNNs
- LSTMs

Presented by **Nirvik Ghosh**:

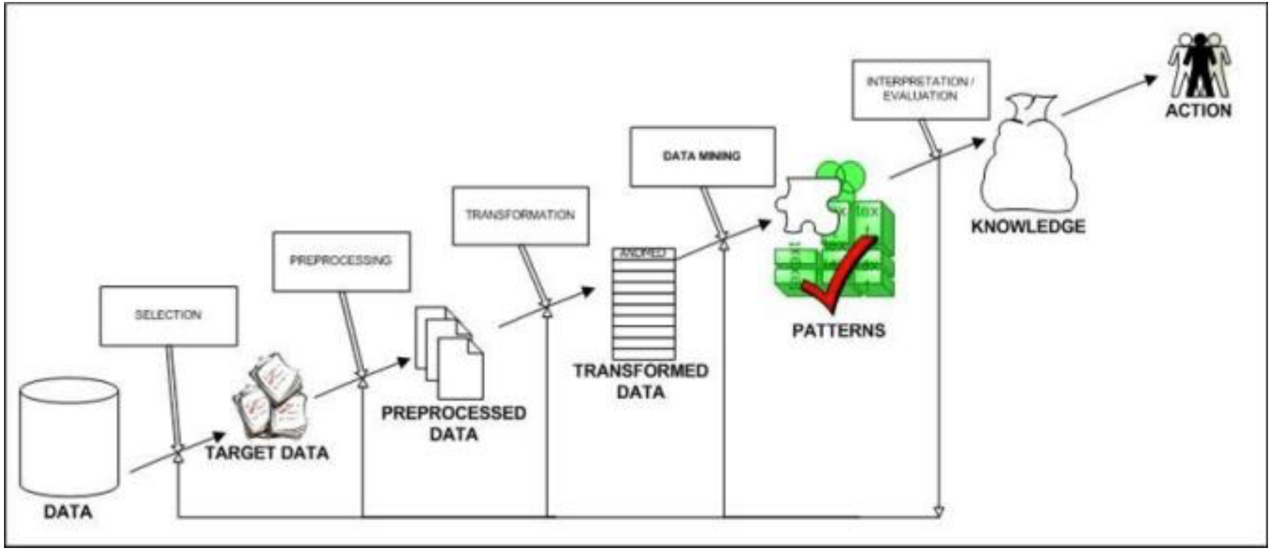
- Autoencoders
- GANs

Presented by **Sweta Kumari**:

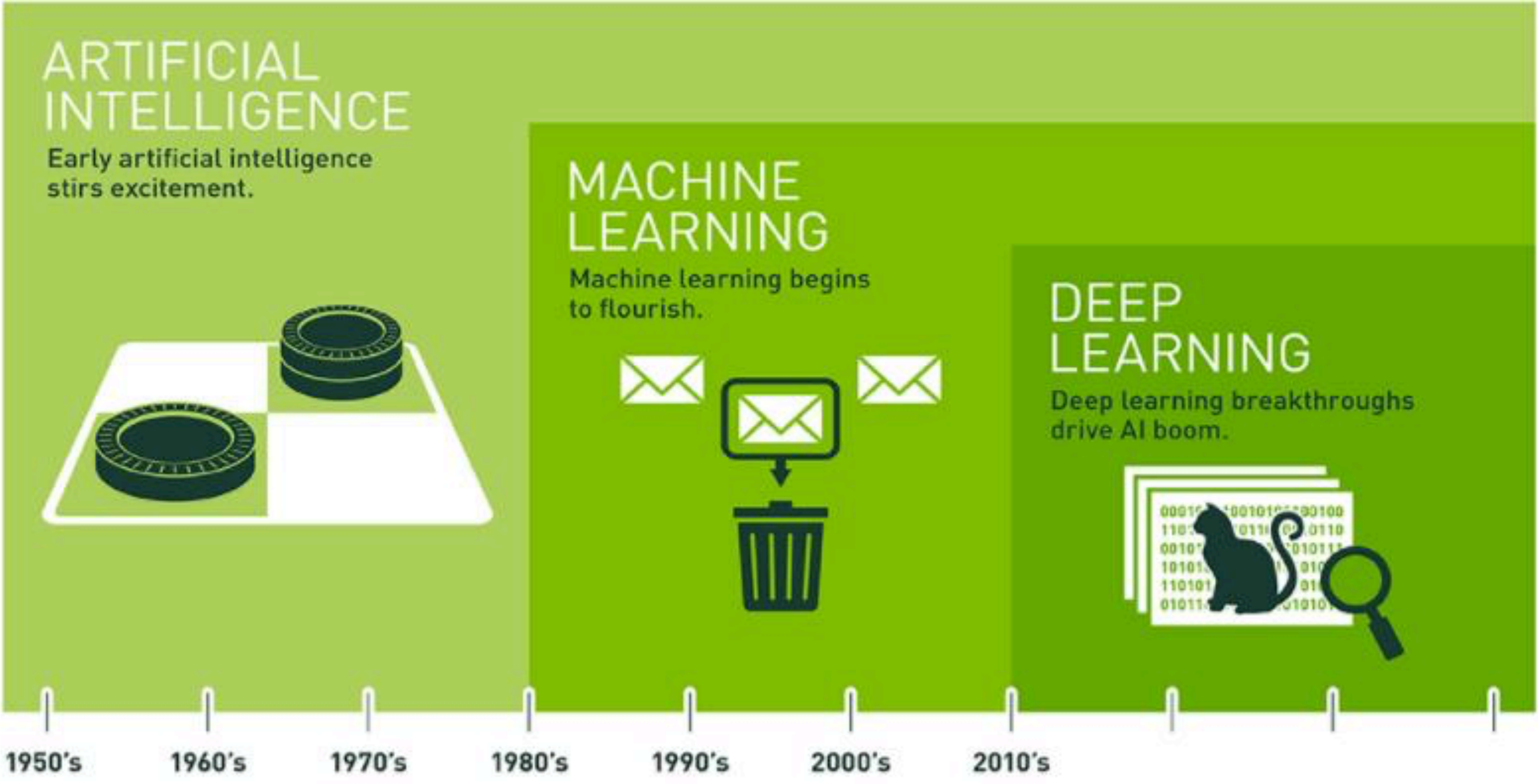
- Mastering the game of GO without human knowledge

What *is* Deep Learning?

- Is it Machine Learning but with a twist? How is it relevant to Data Mining?



http://dataminingwarehousing.blogspot.com/2008/10/data-mining-steps-of-data-mining.html

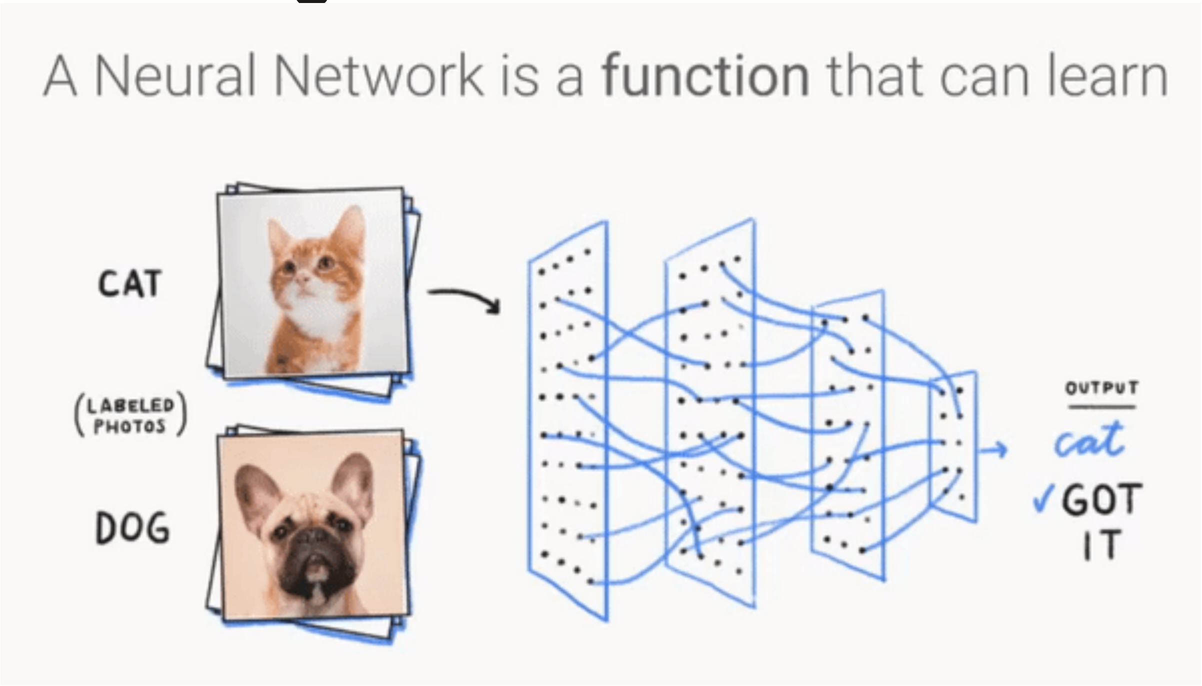


https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai

Deep Learning vs. Machine Learning

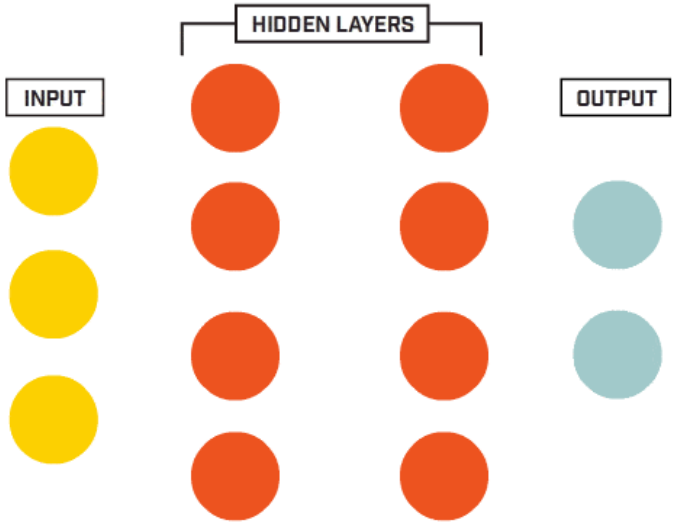
	Machine Learning	Deep Learning
What is it?	Gives computers the ability to learn without being explicitly programmed	Is ML in a way, but a more human-like approach to problem-solving
How do the algorithms work?	Can require manual intervention to check whenever a prediction goes awry	Capable of determining on their own if the prediction are accurate or not
Type of data	Thousands of data points	Millions of data points
Type of output	Usually a numerical value	Can be a score, an element, text, sound etc.

Deep Learning Basics



https://blogs.systweak.com/2018

Deep Learning Basics - contd.



https://www.ptgrey.com/Content/Images/uploaded/MarketingImages/Deep-learning-ai-machine-matrix2

Why is it important?

- Manually designed features are often **over-specified**, **incomplete** and take a **long time to design** and validate
- Learned Features are **easy to adapt**, **fast** to learn
- Deep learning provides a very **flexible**, **universal**, learnable framework

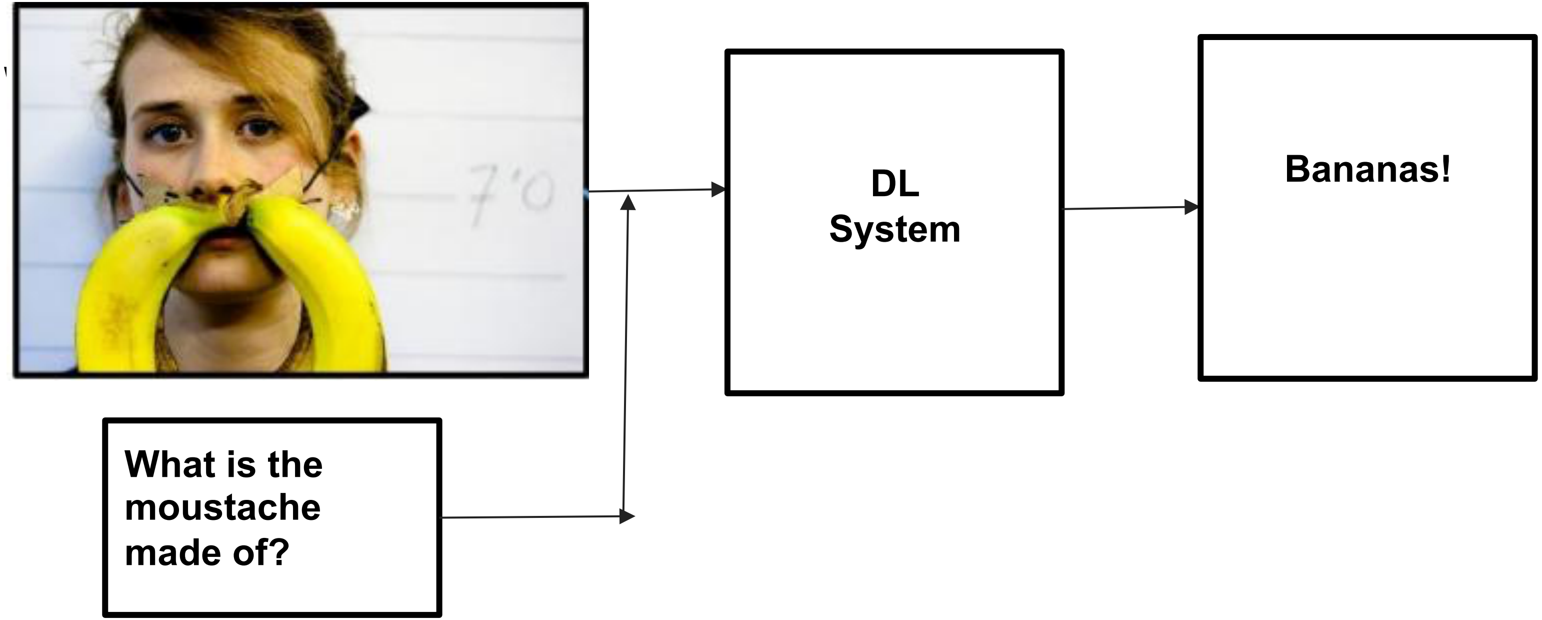
In ~2010, DL started outperforming other ML techniques - first in speech and vision, then NLP

Some DL Applications

Image Captioning

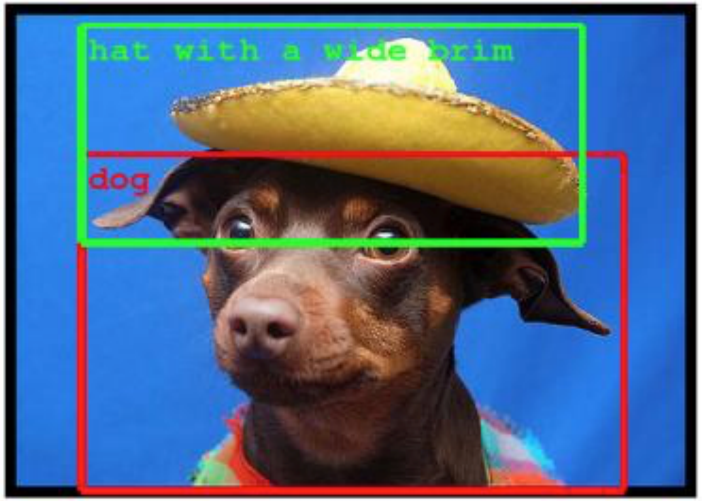
- A black cat has its head in a white toilet
- A black cat is balancing on the rim of a toilet bowl
- A black cat with it’s head inside of a toilet bowl

Visual Question Answering

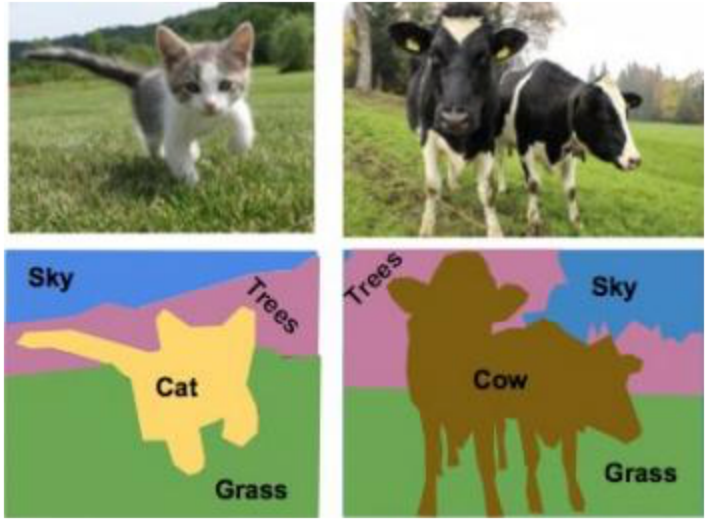


https://avisingh599.github.io/deeplearning/visual-qa/

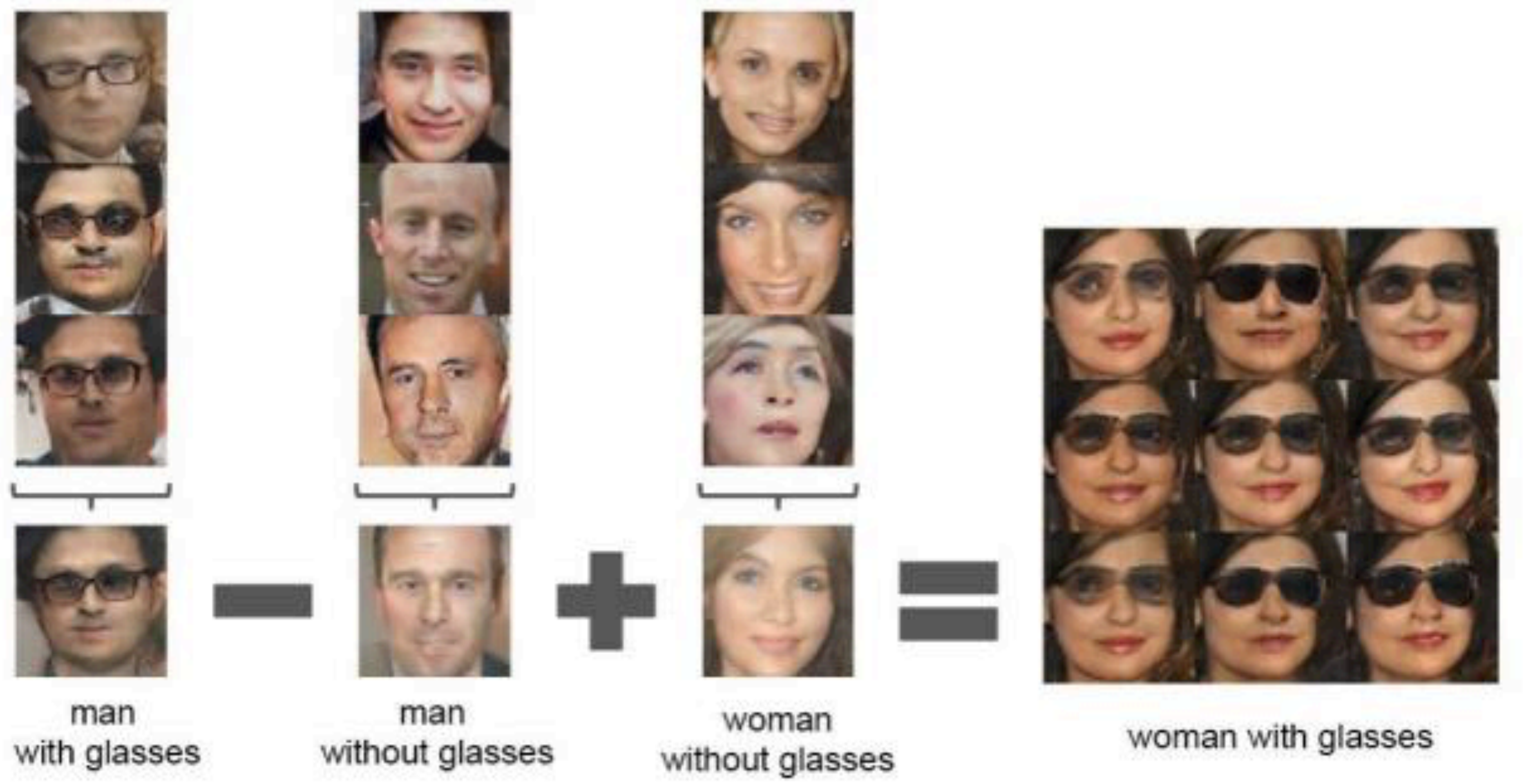
Object Detection



Semantic Segmentation



https://machinelearningmastery.com



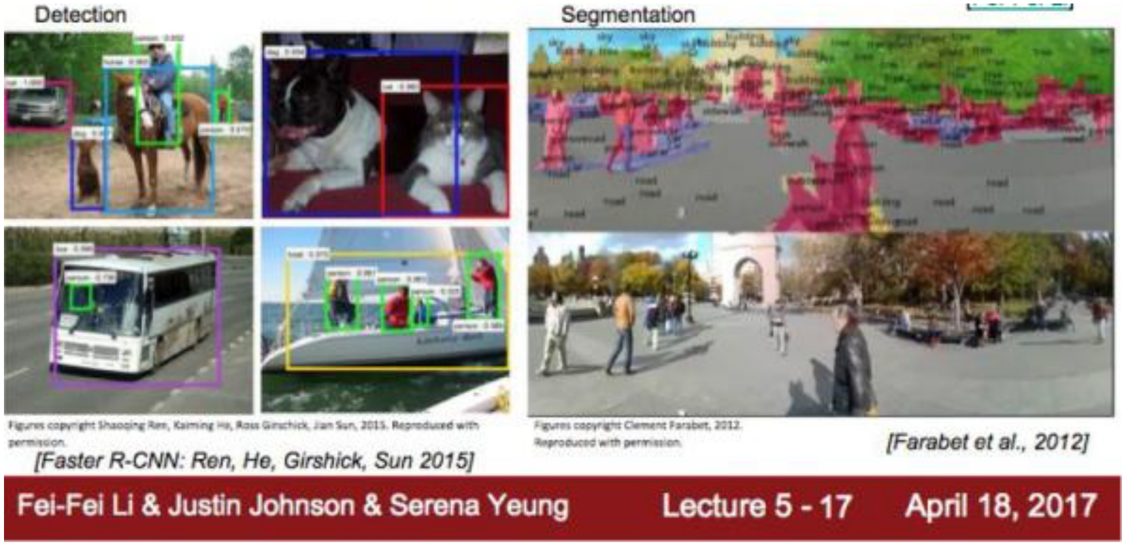
http://hi.cs.stonybrook.edu/cse-527

Who uses DL?

https://en.wikipedia.org/wiki/Category:Software_logos

Convolutional Neural Network (CNNs)

CNNs are everywhere!



http://hi.cs.stonybrook.edu/cse-527

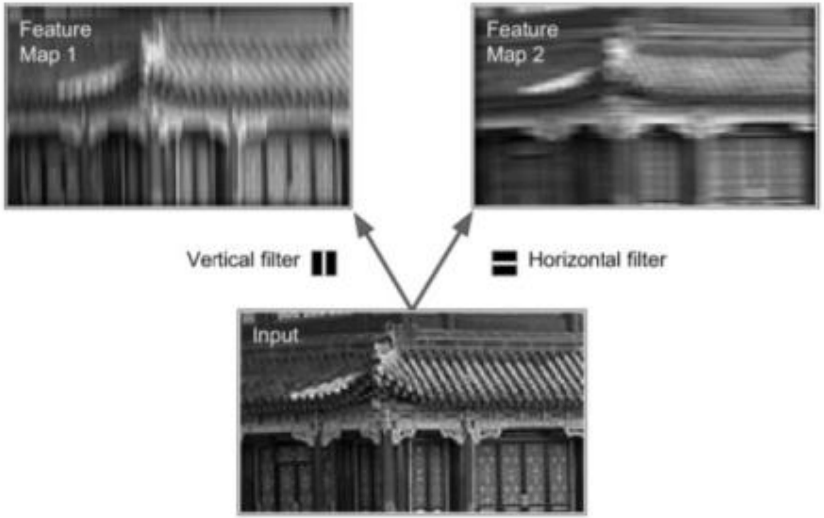
Convolutional Layers

Similar to human vision

Not fully connected, neurons at each layer are connected a handful of neurons in the layer below.

Each neuron learns a set of weights - convolution kernel.

Backprop adjusts weights to enable the neuron to learn the best convolution to minimize value of loss function.

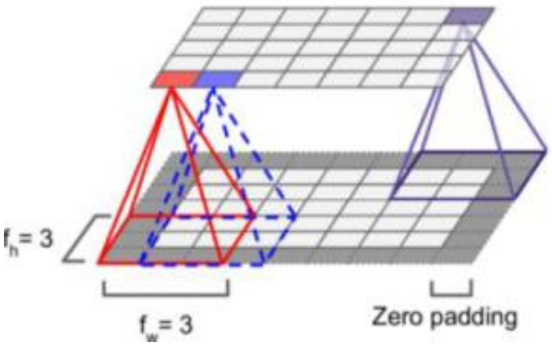


http://hi.cs.stonybrook.edu/cse-527

Padding and Stride

Border pixels missing out, because no padding.

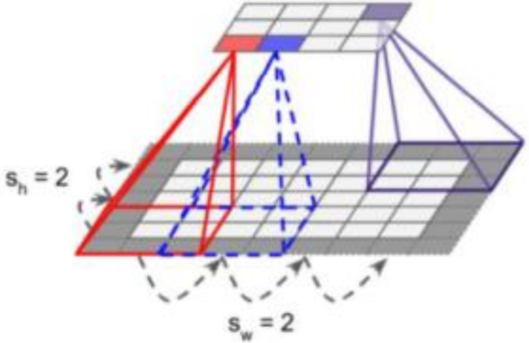
Padding provides control of the output volume spatial size. In particular, sometimes it is desirable to exactly preserve the spatial size of the input volume.



http://hi.cs.stonybrook.edu/cse-527

Stride controls how depth columns around the spatial dimensions (width and height) are allocated.

Used to reduce size of output of convolutional layers.

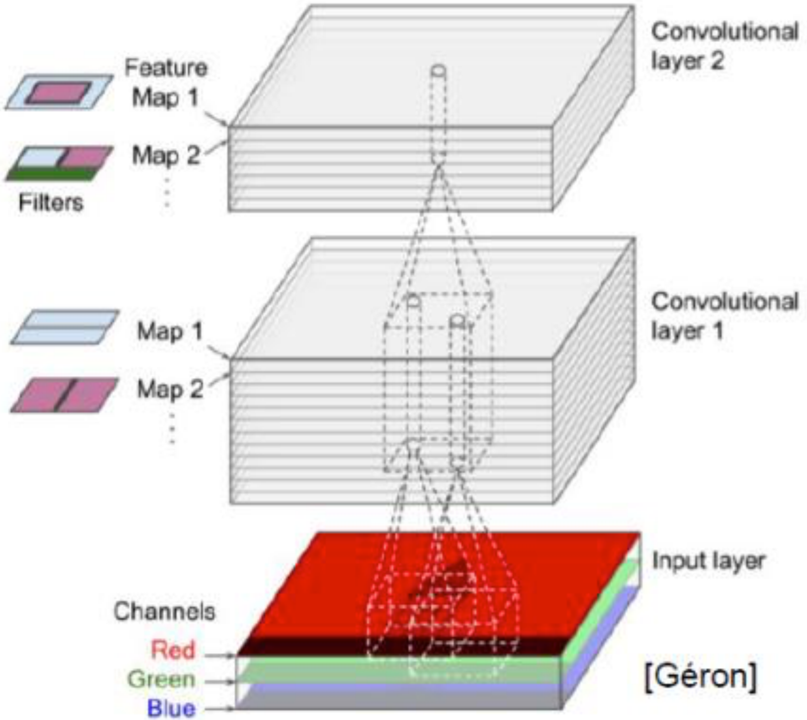


Feature Maps

Each neuron contributes a single feature extraction in form of a convolved image output.

All of these outputs are stacked together to form a convolutional layer.

Problem: This can blow up in dimensions. Oops!

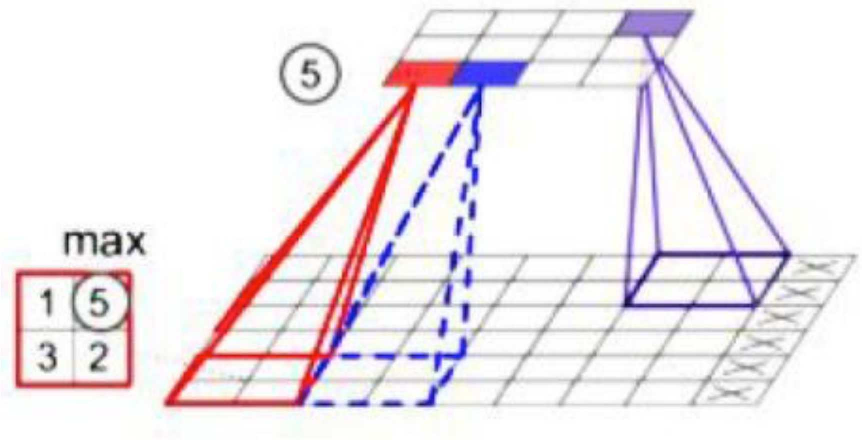


http://hi.cs.stonybrook.edu/cse-527

Pooling layer

Pooling layer periodically inserted between successive Conv layers in a ConvNet.

It progressively reduces the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence also controls overfitting.

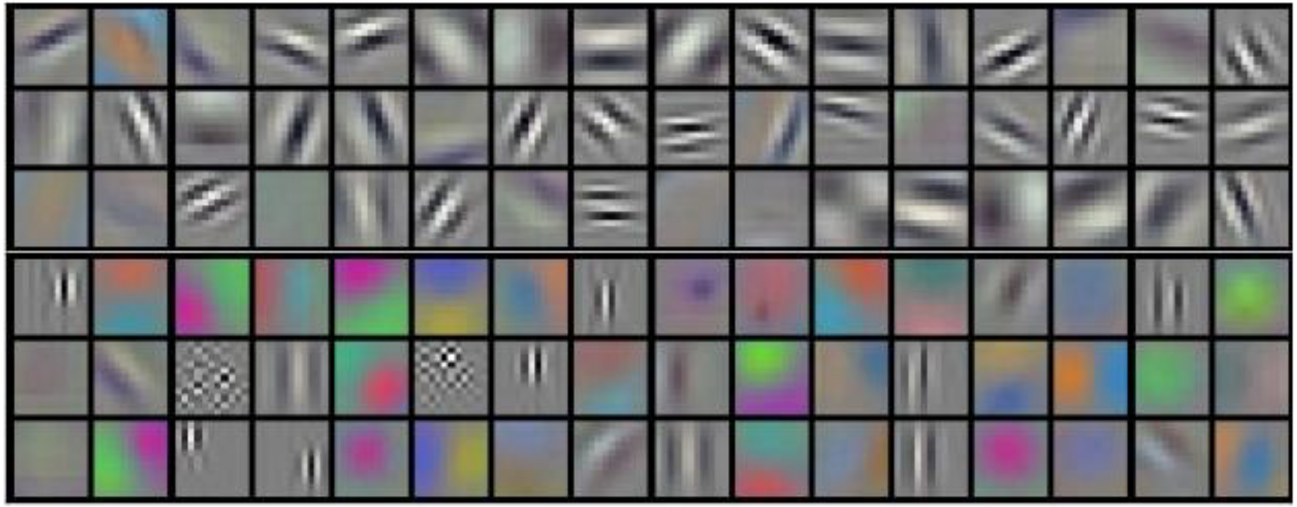


max-pool layer

http://hi.cs.stonybrook.edu/cse-527

Inside a CNN!

[CNN Visualization]



http://hi.cs.stonybrook.edu/cse-527
http://cs231n.github.io/convolutional-networks/

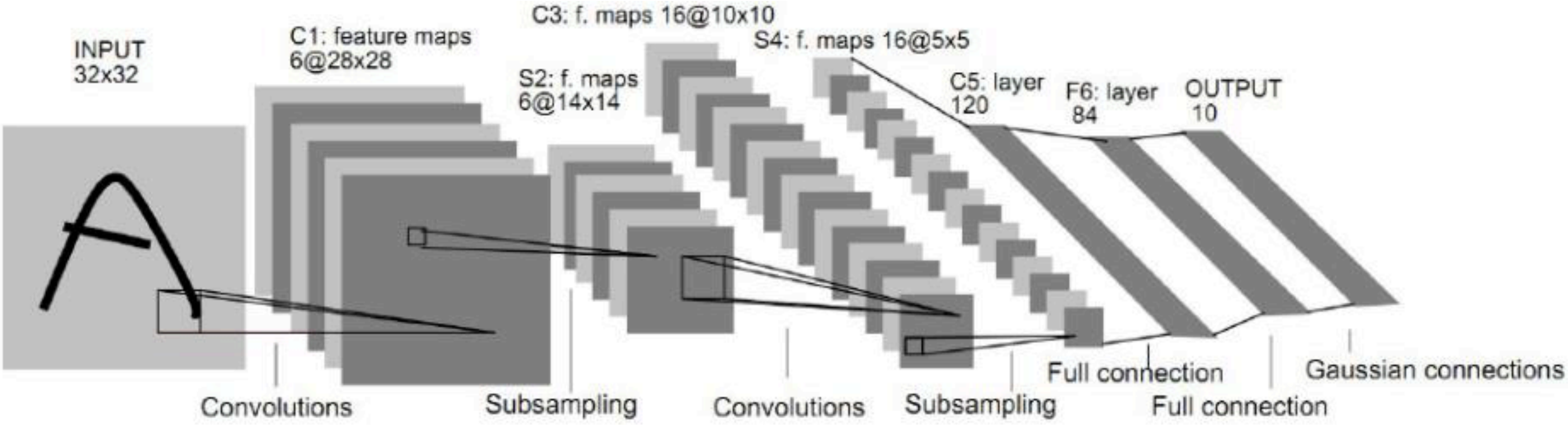
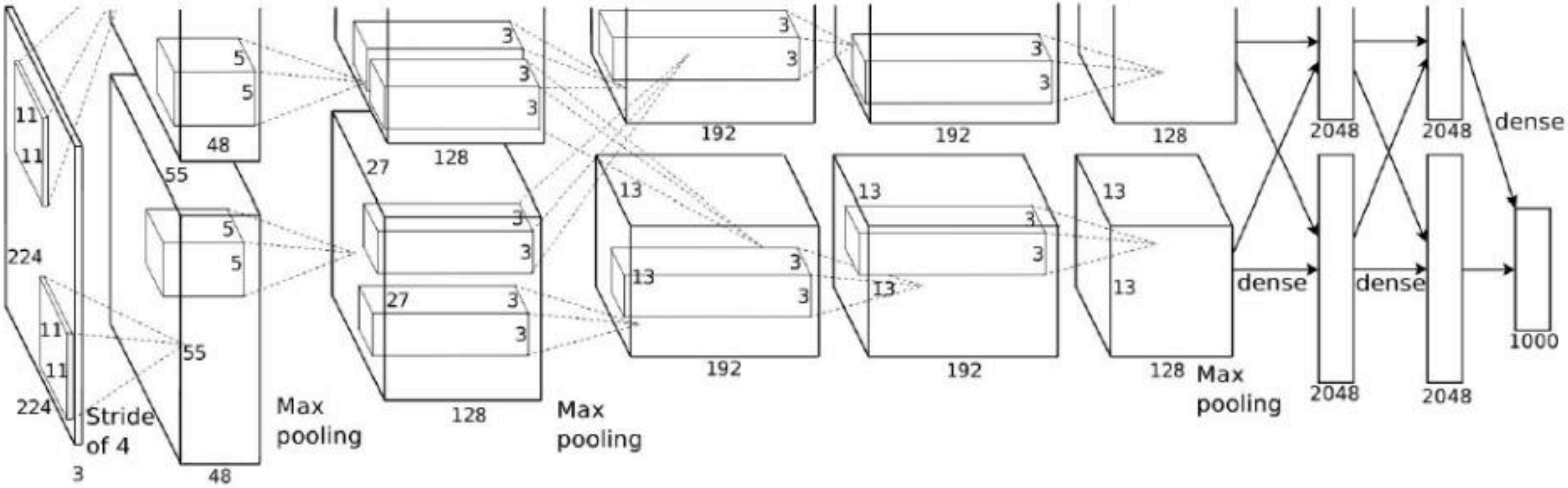


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

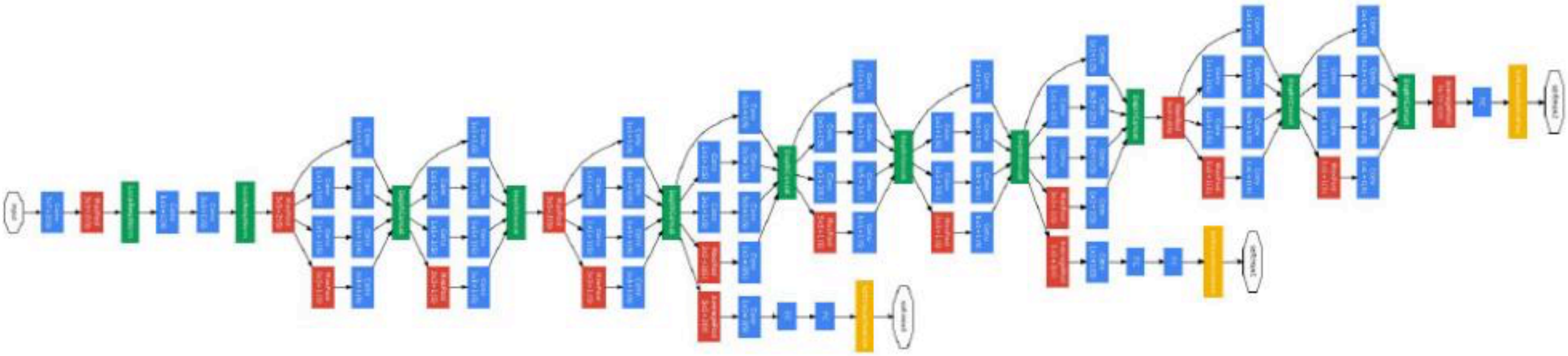
"LeNet", LeCun 1998. Achieved 98% accuracy on MNIST (handwritten digit classification)

http://hi.cs.stonybrook.edu/cse-527



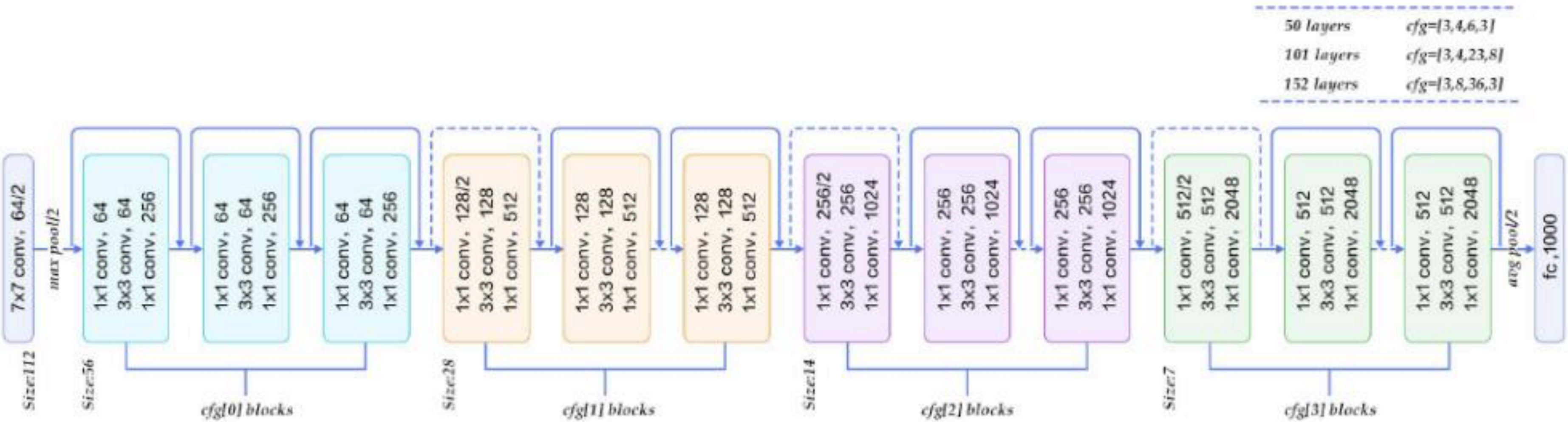
"AlexNet", Krizhevsky + Hinton 2010. Achieved ~15% top-5 error rate on ImageNet Large Scale Visual Recognition challenge (ILSVRC). Beat the 2nd runner up by more than 10%!

http://hi.cs.stonybrook.edu/cse-527



“GoogLeNet”, Szegedy et al 2015. Has 22 layers. Won the ILSVRC 2014, achieved 6.67% top-5 accuracy.

http://hi.cs.stonybrook.edu/cse-527

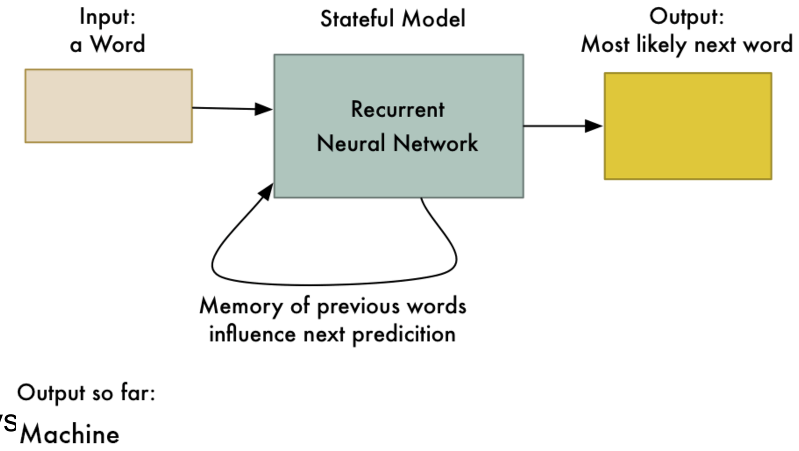


"ResNet" (Residual Network), He et al. 2016. Achieved **3.57%** accuracy on the ILSVRC 2015 with 152 layers.

http://hi.cs.stonybrook.edu/cse-527

Recurrent Neural Networks (RNNs)

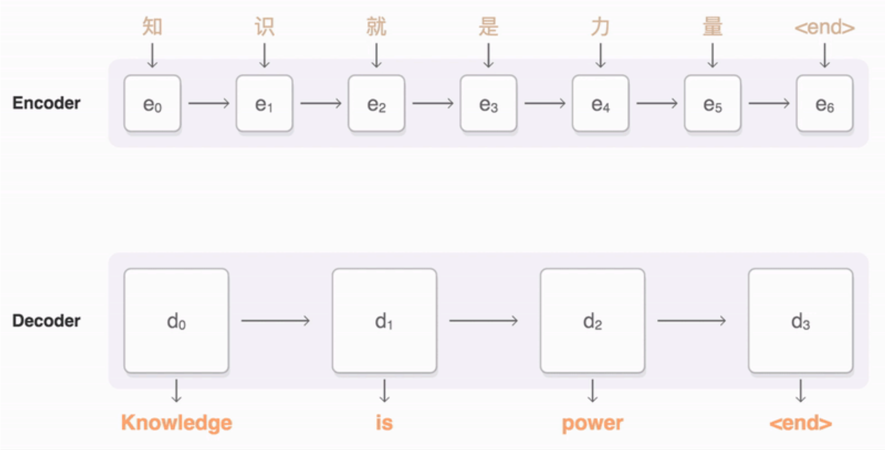
- Traditional neural network assume that all inputs (and outputs) are independent of each other.
- If you want to predict the next word in a sentence you better know which words came before it.
- Need for Neural networks dealing with long term sequences.
- Single input, sequence of output (Eg. Image captioning)
  Sequence input, single output (Eg. Sentiment analys
  Sequence to sequence (Eg. Video captioning)
- The idea behind RNNs is to make use of sequential information.

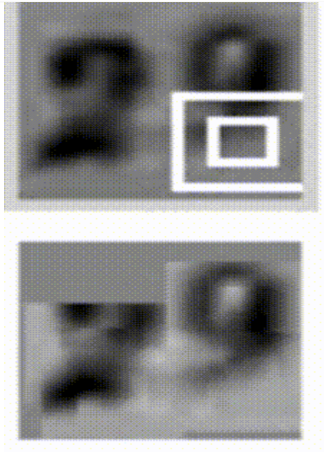


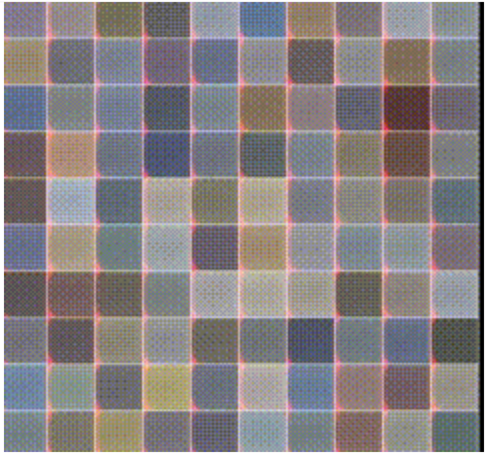
https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa

Main application areas

- Natural language processing
- POS tagging
- Syntactic Parsing
- Information Extraction
  - Entity Extraction
  - Relation Extraction
- Semantic Parsing
- Summarization
- Question Answering
- Dialog
- Image/Video captioning





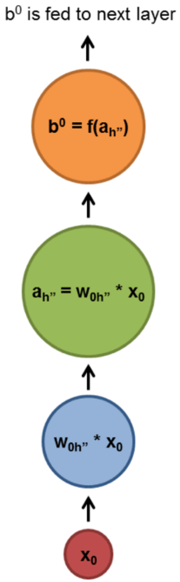


https://recast.ai/blog/ml-spotlight-rnn/ http://karpathy.github.io/2015/05/21/rnn-effectiveness/

RNN architecture

$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t\right)$$

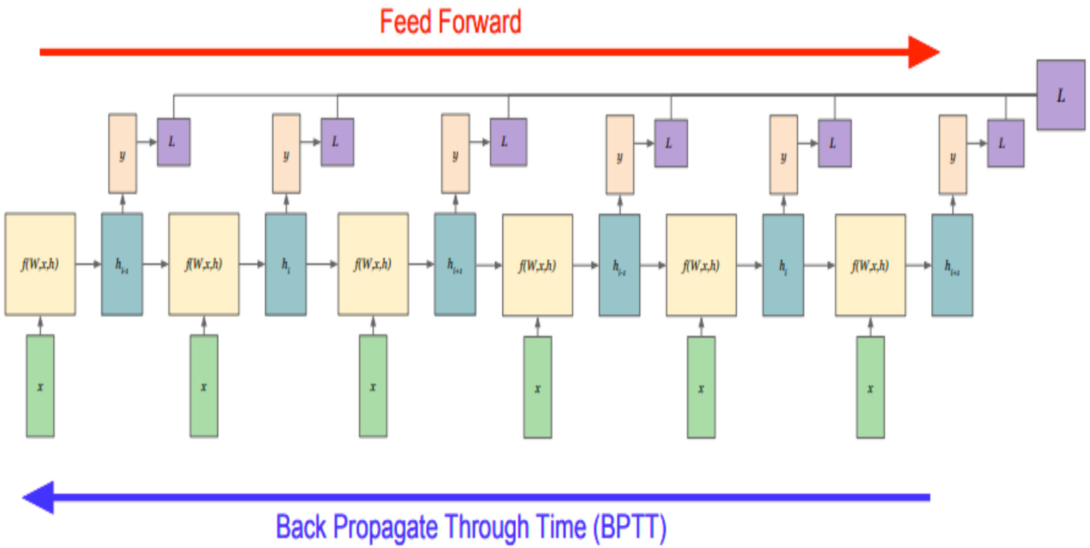
$$y_t = W_{hy}h_t$$



1. A single time step of the input xt is supplied to the network.
2. We then calculate ht its current state using a combination of the current input and the previous state.
3. The current ht becomes ht-1 for the next time step.
4. We can go as many time steps as the problem demands and combine the information from all the previous states.
5. Once all the time steps are completed the final current state is used to calculate the output yt.

https://deeplearning4j.org/lstm.html#code

Training RNNs- BackPropagation Through Time

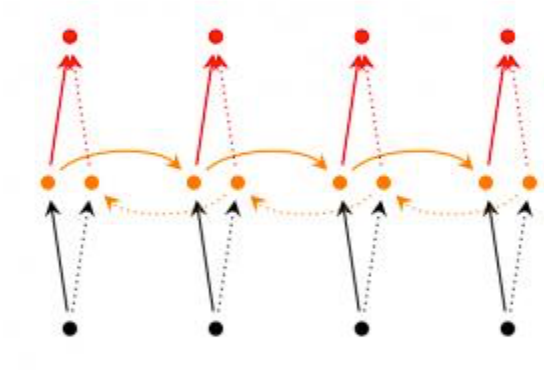


- Feed one character at a Time.
- Sample the output distribution (softmax).

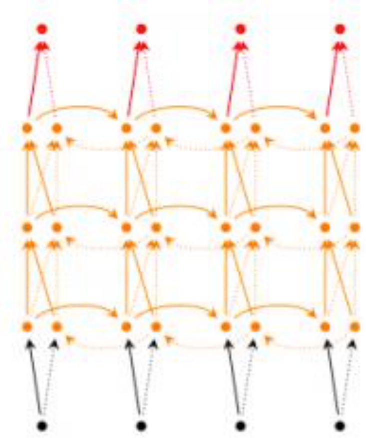
https://deeplearning4j.org/lstm.html#code

RNN extensions & architectures

Bidirectional RNNs



Deep (Bidirectional) RNNs



$$\overrightarrow{h}_t = f(\overrightarrow{W} x_t + \overrightarrow{V}\,\overrightarrow{h}_{t-1} + \overrightarrow{b})$$

$$\overleftarrow{h}_t = f(\overleftarrow{W} x_t + \overleftarrow{V}\,\overleftarrow{h}_{t+1} + \overleftarrow{b})$$

$$y_t = g(U_{\rightarrow}\overrightarrow{h}_t + U_{\leftarrow}\overleftarrow{h}_t + c)$$

http://docs.huihoo.com/deep-learning/deeplearningsummerschool/2015/Deep-NLP-Recurrent-Neural-Networks.pdf

Long Short Term Memory Cells (LSTMs)

“The man who ate my pizza has purple hair”

- In this case, the description purple hair is for the man and not the pizz
  This is a long term dependency.

- Regular RNNs might have difficulty in learning long range
  Dependencies.

- Vanishing/Exploding gradients.
  - Exploding gradients can be clipped.
  - What about vanishing gradients ??

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Core idea of LSTM networks

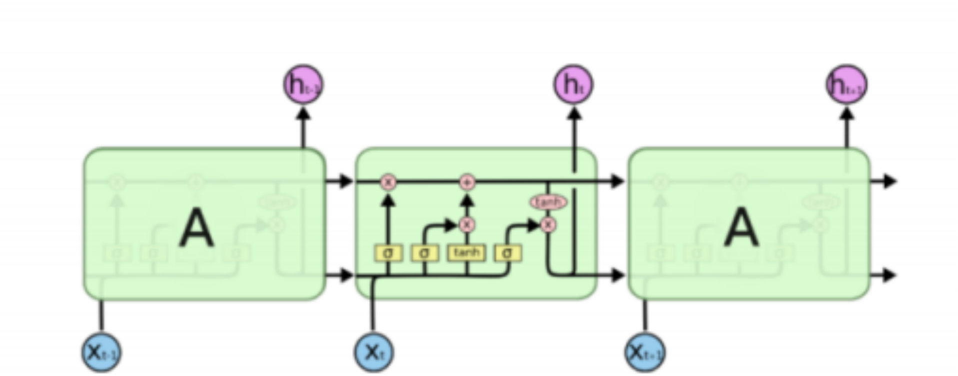
Control how various bits of information are combined.

1. Memory of what has been seen so far? **ce state.**

2. How much should previous state contribute to the current state? **Forget gate**

3. How much should current input contribute to the current state? **Input gate**

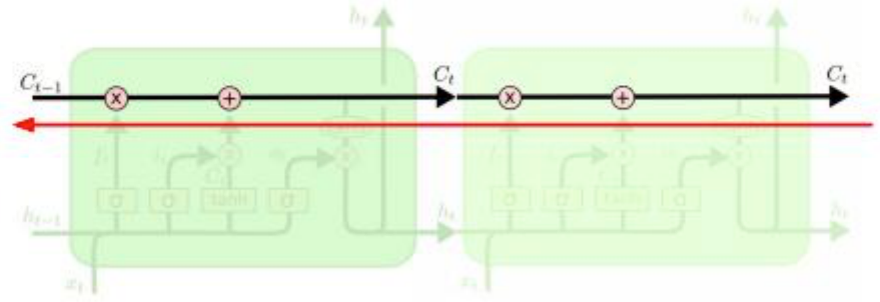
4. Should the current state contribute to future state representations? **Output gate**



The repeating module in an LSTM contains four interacting layers.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Gradient “highway” to prevent them from dying.
- Cell state C has no multiplication by W!
- But a series of “gates” that control what stays and what disappears (forgotten) from C.



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) are deep neural net architectures comprised of two nets, pitting one against the other (thus the “adversarial”). Known widely for their zero-sum game.

GANs have been used to produce samples of photorealistic images for the purposes of visualizing

Facebook’s AI research director Yann LeCun adversarial training as “the most interesting idea in the last 10 years in ML.”

https://deeplearning4j.org/generative-adversarial-network

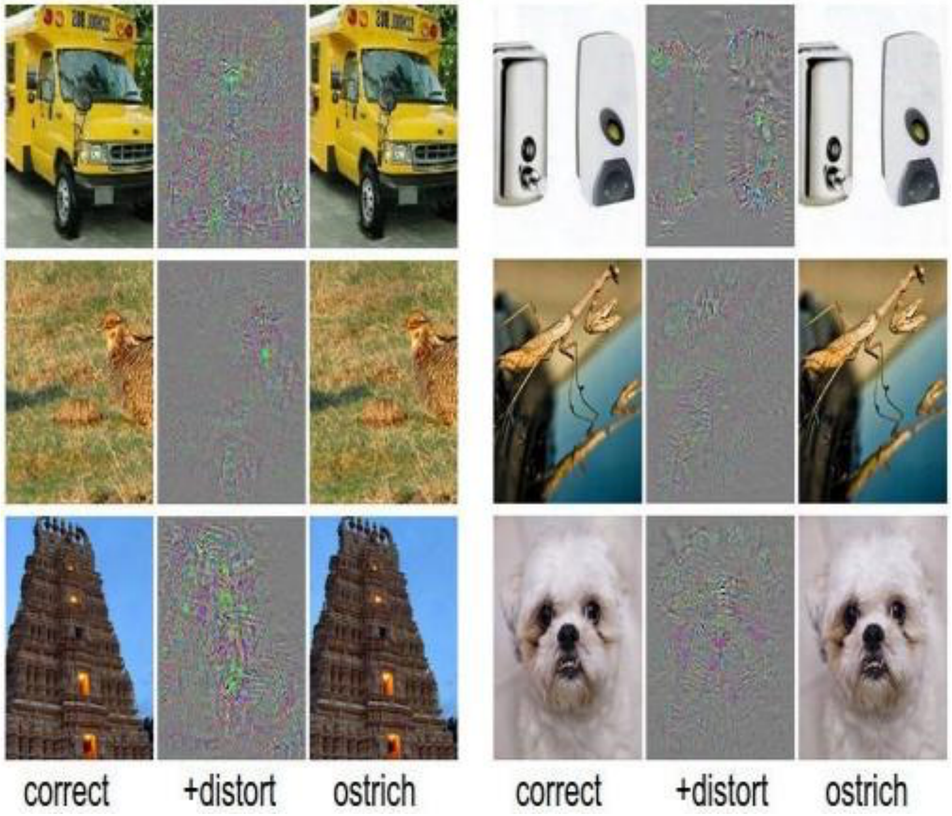
Why GAN ?

Input: Correct labeled image

Add a little bit of noise

Output: Completely wrong label

Purpose: Take in noise vectors and produce images that closely resembles input data in order to fool the discriminator into classifying a fake image as a real image.



https://medium.com/deep-dimension/gans-a-modern-perspective-83ed64b42f5c

GAN Goals

Train two separate networks with competitive goals:

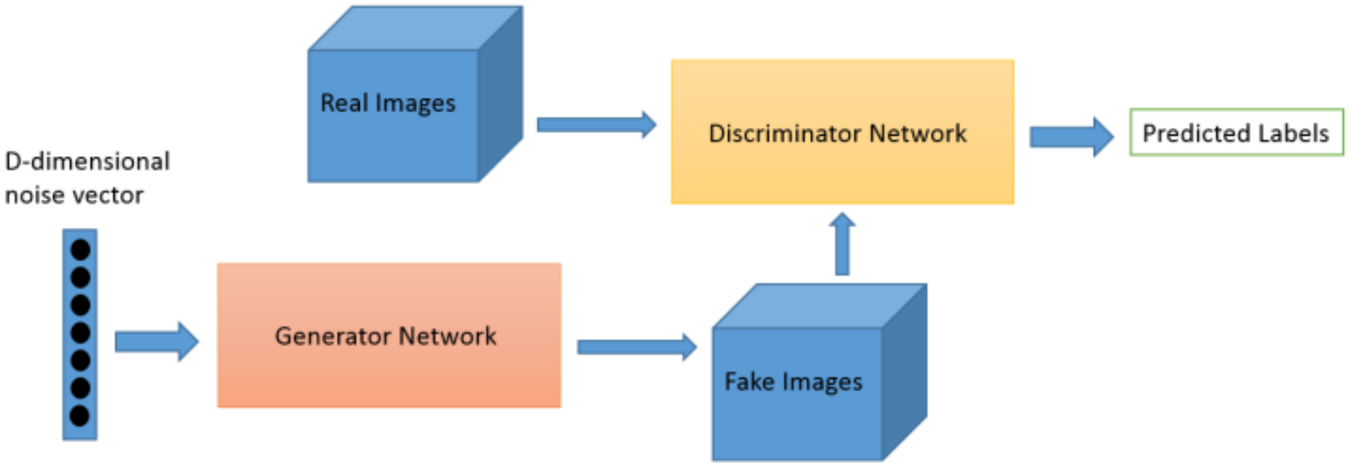
One network produces answers (generative)

Another network distinguishes between the real and the generated answers (adversarial)

The concept is to train these networks competitively, so that after some time, neither network can make further progress against the other.

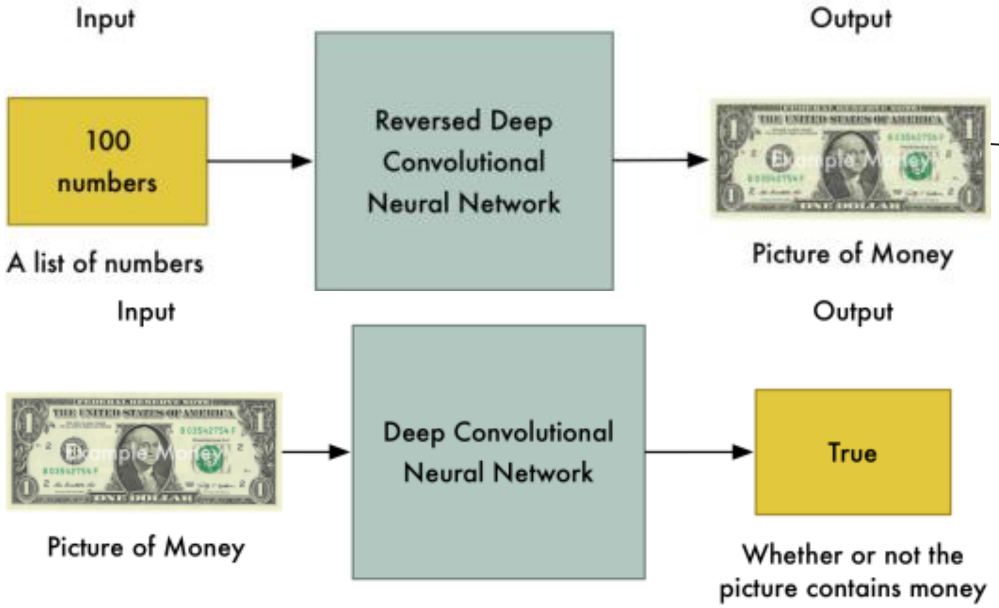
GAN in a nutshell

GANs are neural networks composed up of two networks competing with each other. The two networks namely generator—to generate data set and discriminator—to validate the data set.



https://deeplearning4j.org/generative-adversarial-network

GAN - Counterfeit notes game

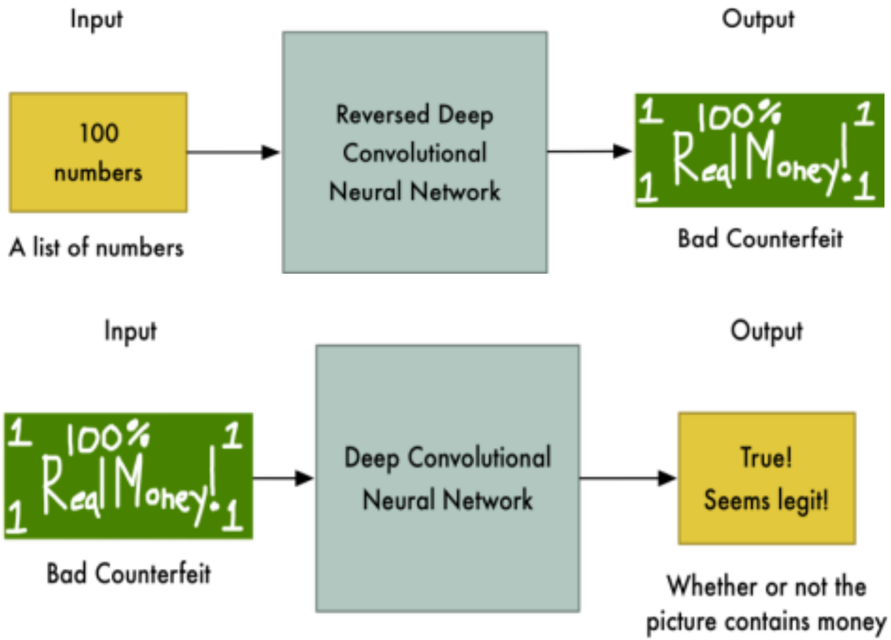


Generator

Discriminator

https://deeplearning4j.org/generative-adversarial-network

Game - contd.

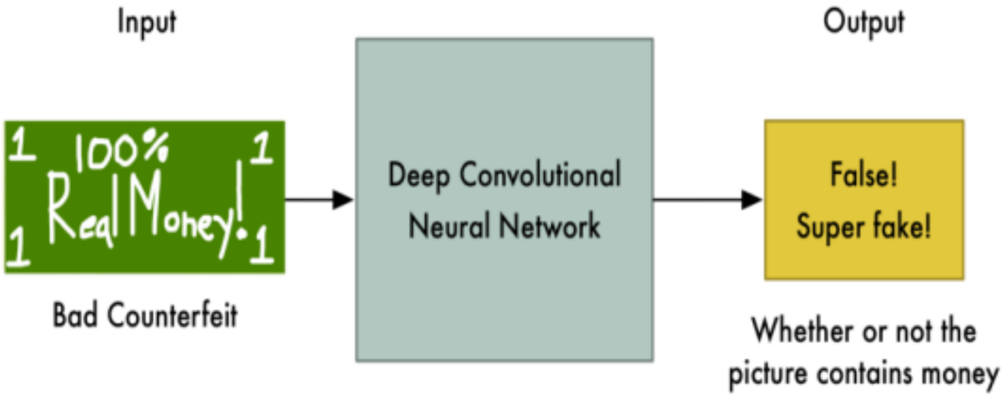


Generator tries to create a fake note

Discriminator is not smart enough yet.
Tell it to detect a face on the note

https://deeplearning4j.org/generative-adversarial-network

GAN (Last round)



Discriminator will reject all such counterfeit notes as there is no face.

Generator can then add an image and fool the Discriminator. However, the discriminator learns more unique features about the actual notes and classifies against the generated note.

Keep going on until cannot distinguish

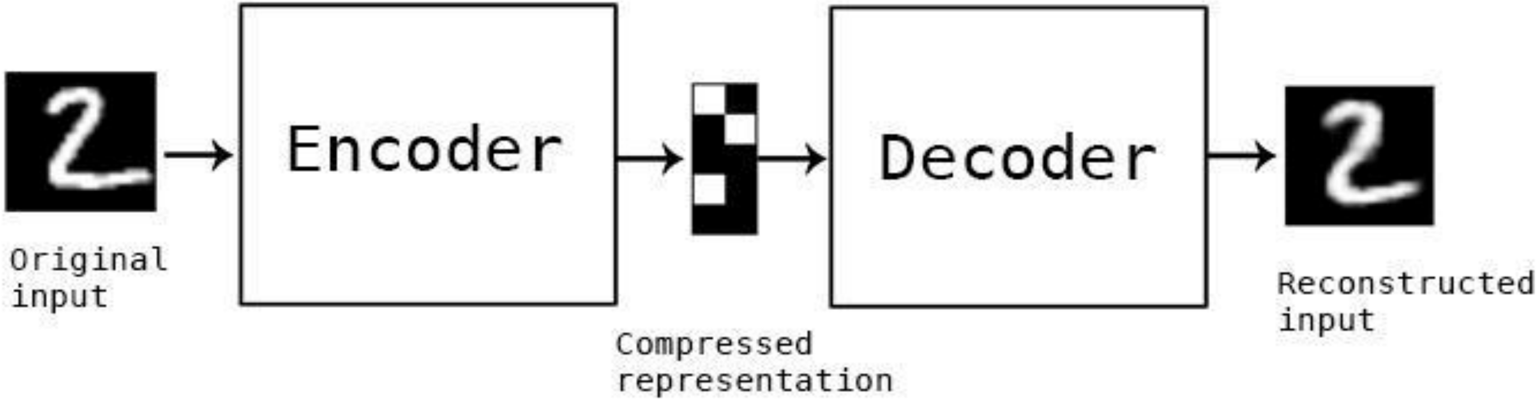
https://deeplearning4j.org/generative-adversarial-network

Autoencoders

- Encode input data as vectors
- Create hidden and compressed representation of raw data
- Extremely useful for dimensionality reduction
- Comes with decoder which reconstructs the input

https://deeplearning4j.org/deepautoencoder

Autoencoders



https://deeplearning4j.org/deepautoencoder

ARTICLE

doi:10.1038/nature24270

Mastering the game of Go without human knowledge

David Silver[1]*, Julian Schrittwieser[1]*, Karen Simonyan[1]*, Ioannis Antonoglou[1], Aja Huang[1], Arthur Guez[1], Thomas Hubert[1], Lucas Baker[1], Matthew Lai[1], Adrian Bolton[1], Yutian Chen[1], Timothy Lillicrap[1], Fan Hui[1], Laurent Sifre[1], George van den Driessche[1], Thore Graepel[1] & Demis Hassabis[1]

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo's own move selections and also the winner of AlphaGo's games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

AlphaGo Zero

- Year: **October 19, 2017**
- Journal: **Nature, Vol. 550**
- Accessible at: https://www.nature.com/articles/nature24270.pdf
- Majorly developed at *DeepMind Inc.*

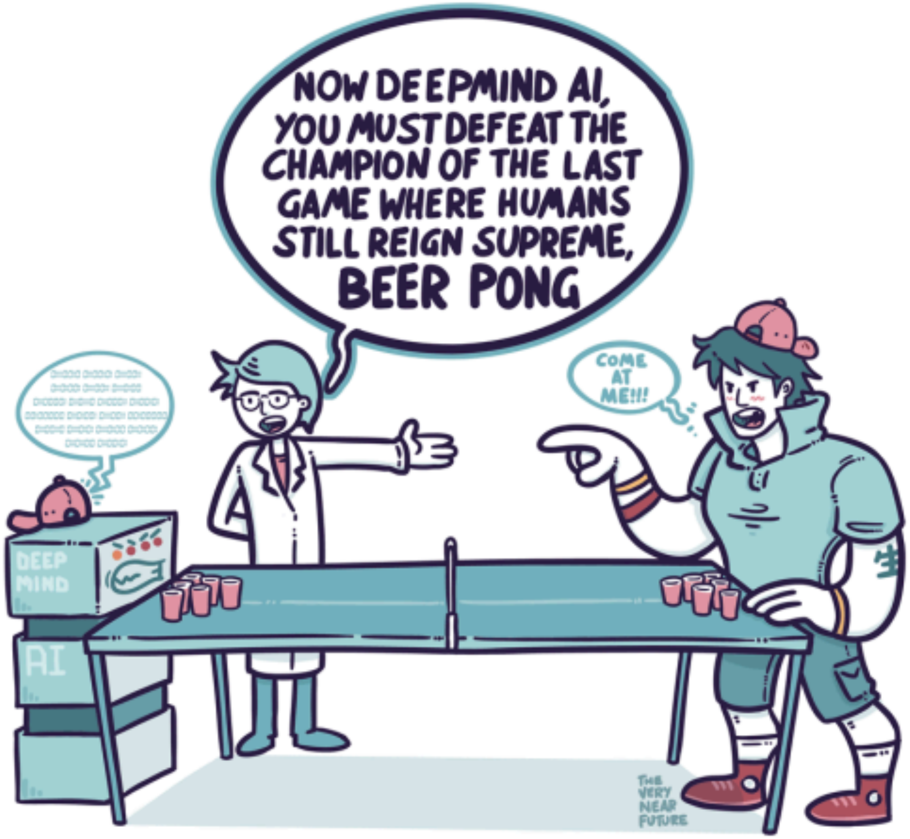
AlphaGo

- Ambition of AI research: To achieve superhuman performance in the most challenging domains with no human input.

- GO : Most difficult board game

- Why complex : sheer number of possible moves and the difficulty of evaluating the strength of each possible board position.

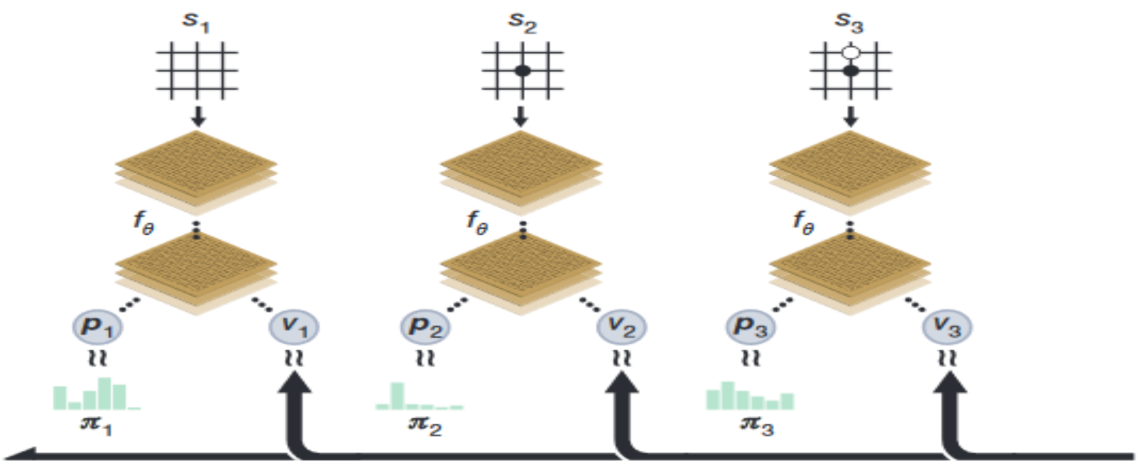
- AlphaGo - Defeated 18-time world champion *Lee Sedol* - by 100 games to 0 - March 2016



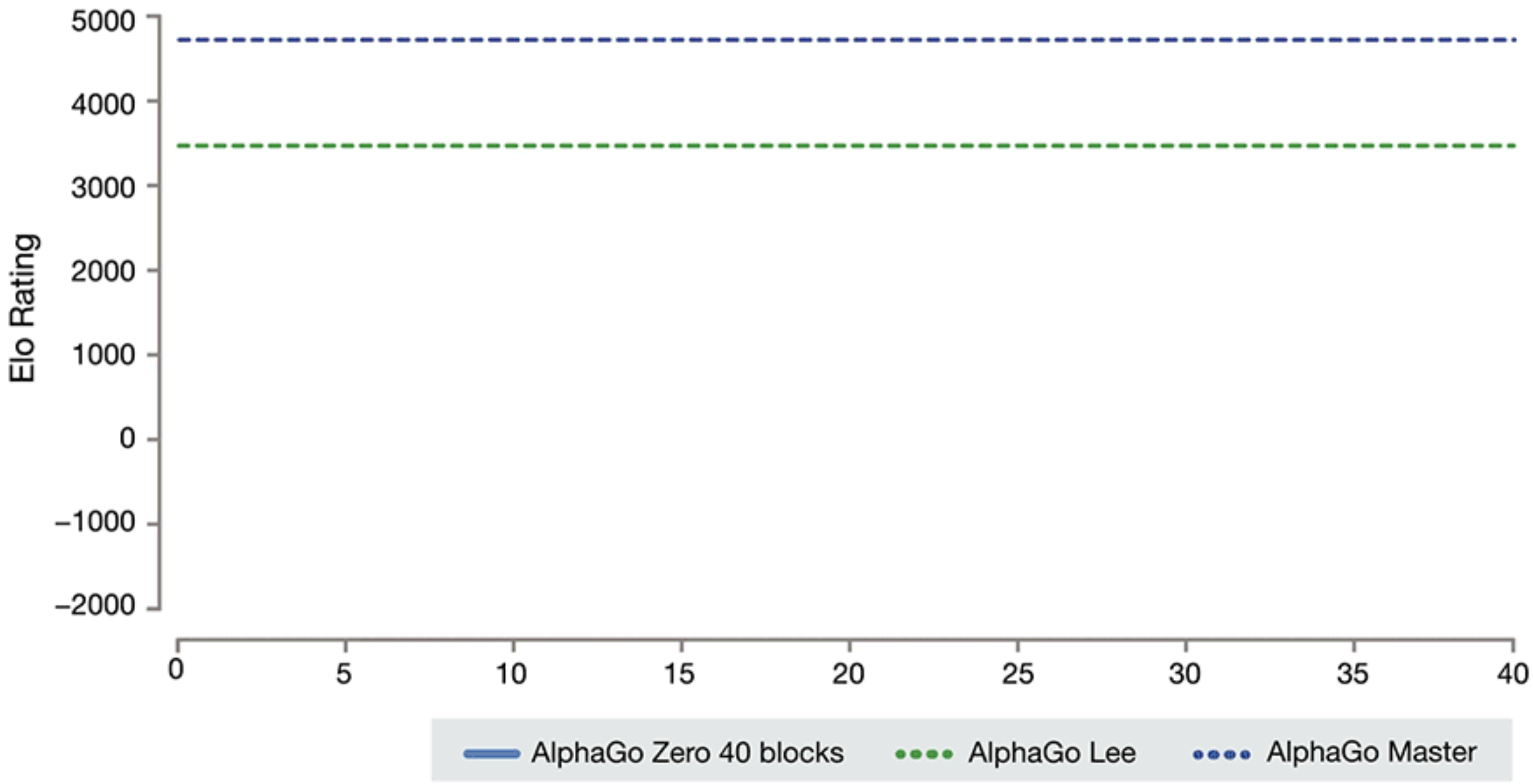
https://deepmind.com/blog/alphago-zero-learning-scratch/

What's the secret recipe of AlphaGo

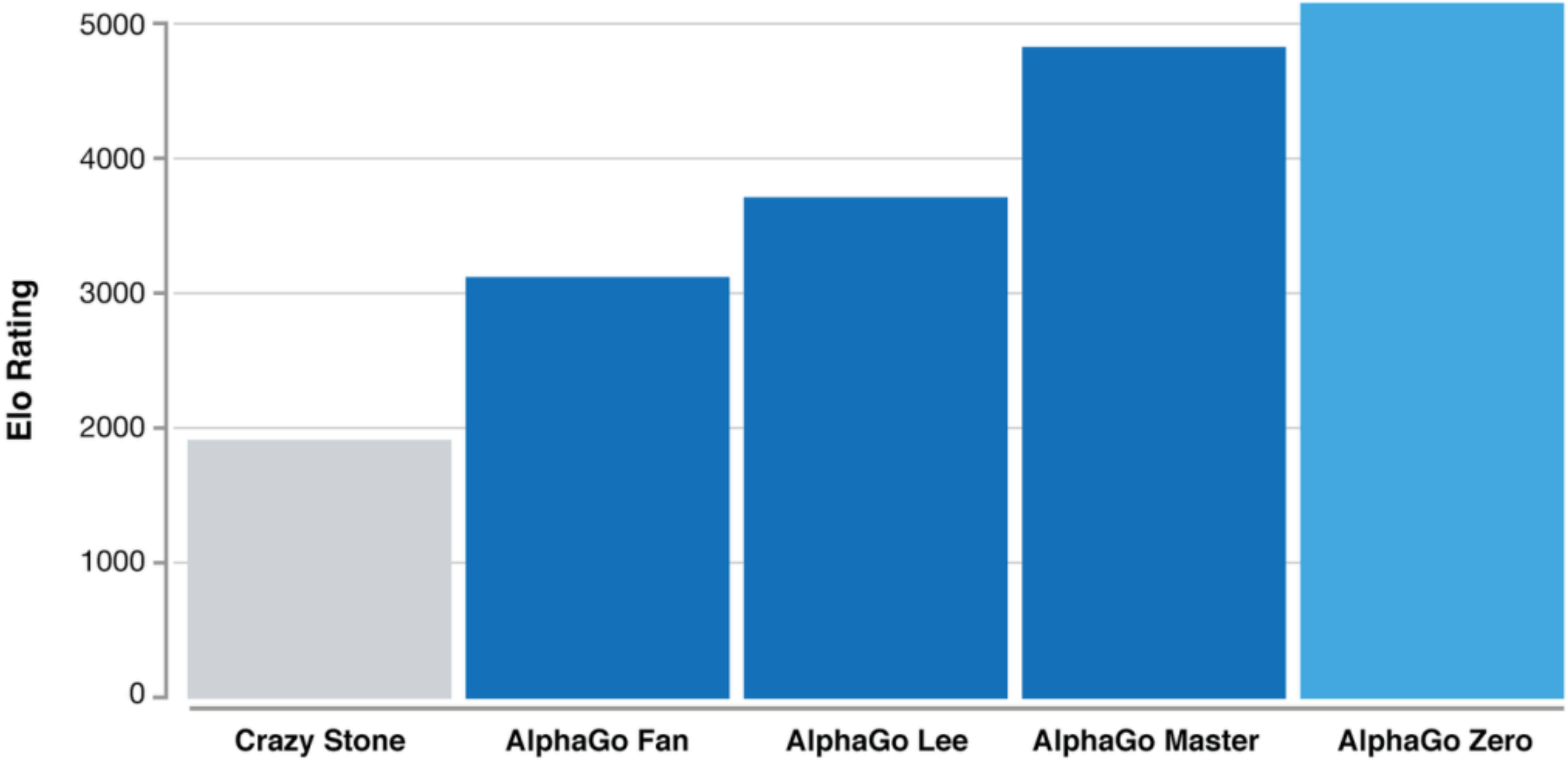
- AlphaGo Zero learnt to play the game of Go simply by playing games against itself, starting from completely random play.
- What AlphaGo does : **Advanced tree search** with **Deep Neural Networks** and **Reinforcement learning**



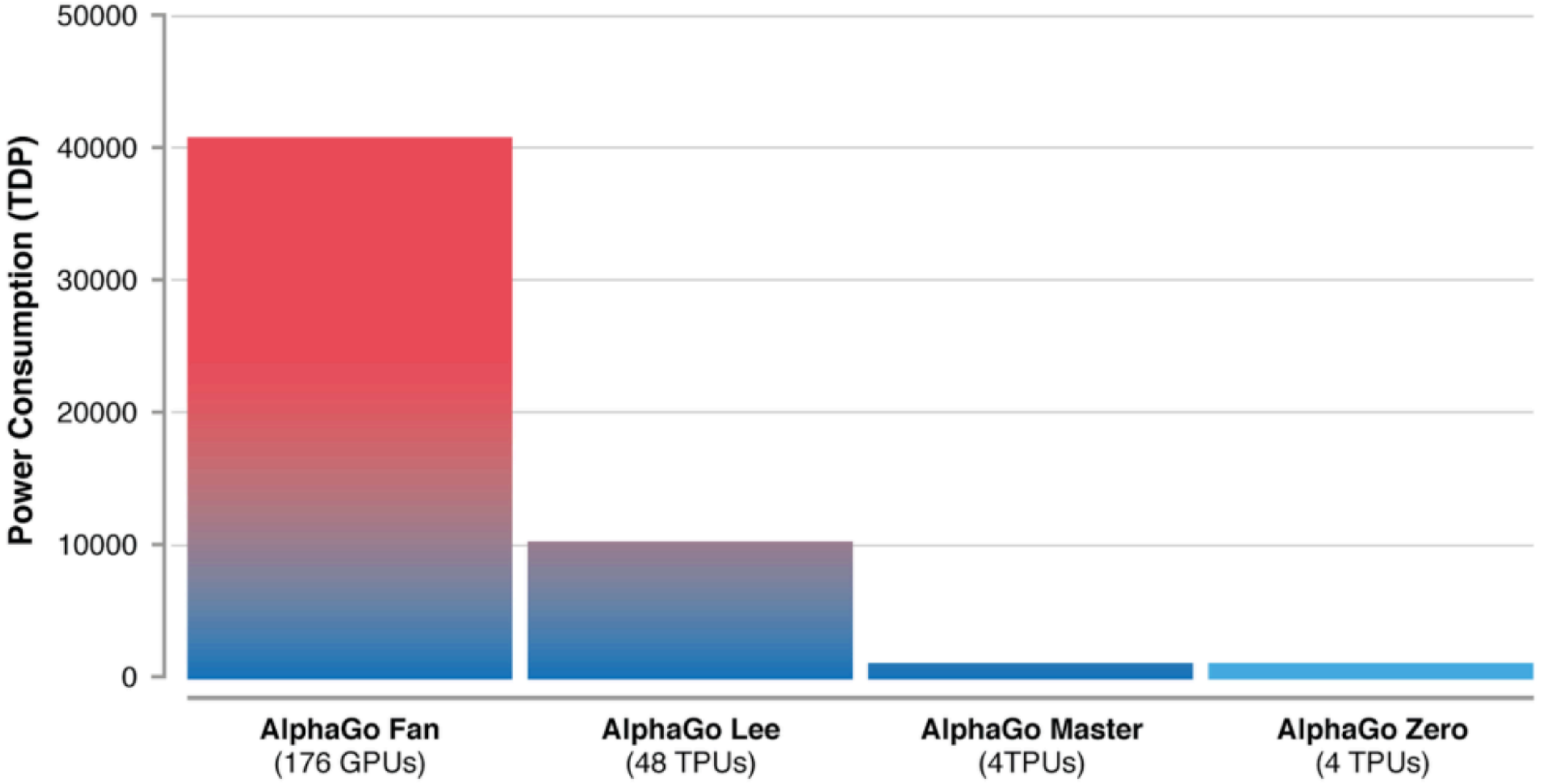
- Deep neural network fθ with parameters θ.
- **Input :** The raw board representations of the position and its history
- **Output :** Move probabilities, **p** and a value, (p, v) =fθ(s).
- **v** : Probability of the current player winning from position **s**



https://deepmind.com/blog/alphago-zero-learning-scratch/



https://deepmind.com/blog/alphago-zero-learning-scratch/

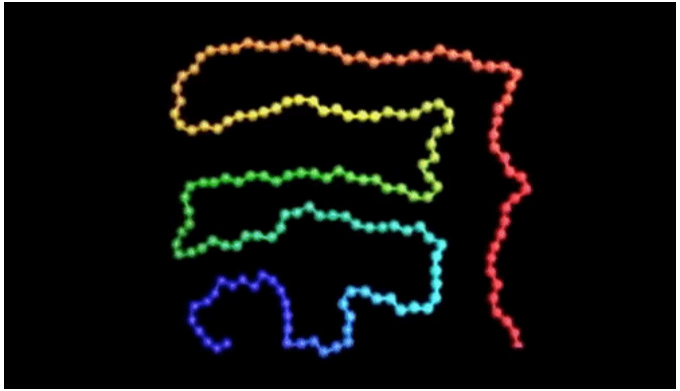


https://deepmind.com/blog/alphago-zero-learning-scratch/

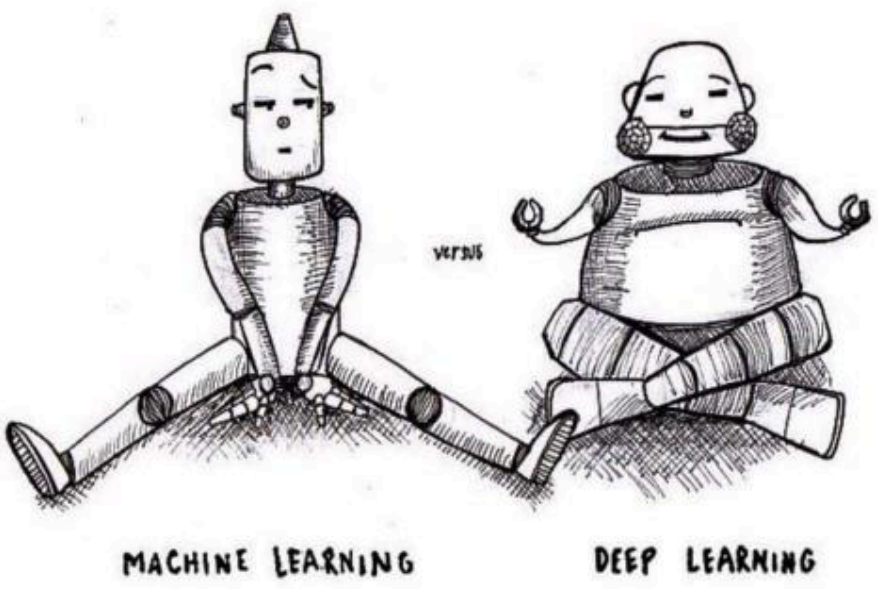
Where to from here?

Similar techniques can be applied to other structured problems

- protein folding
- reducing energy consumption
- searching for revolutionary new materials



Download AlphaGo games:
http://www.alphago-games.com/


versus
MACHINE LEARNING
DEEP LEARNING