INSTITUTE OF SCIENTIFIC INSTRUMENTS
The Czech Academy of Sciences

Introduction to Deep Neural Networks

M.Sc. Petr Nejedlý
nejedly@isibrno.cz

Deep Layers
a machine-learning forum

STRATEGIE AV21

brno ai

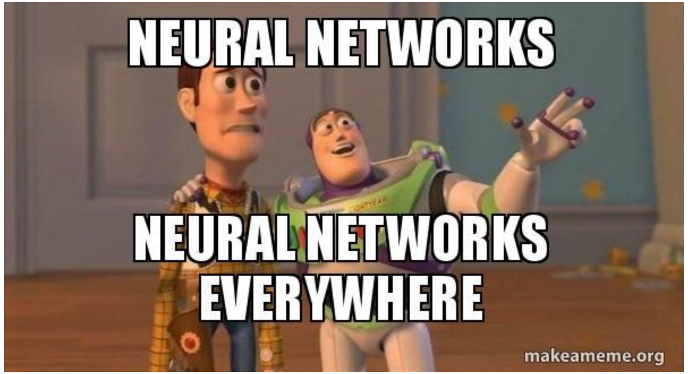
AI MT

Průlomové technologie budoucnosti

ISI CAS

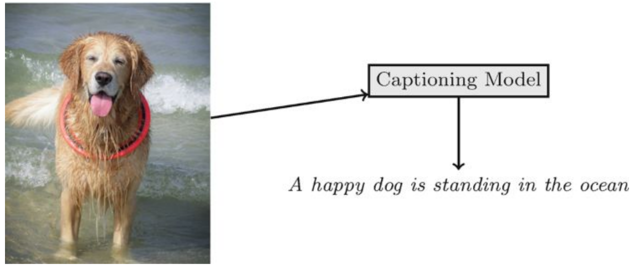
Outline

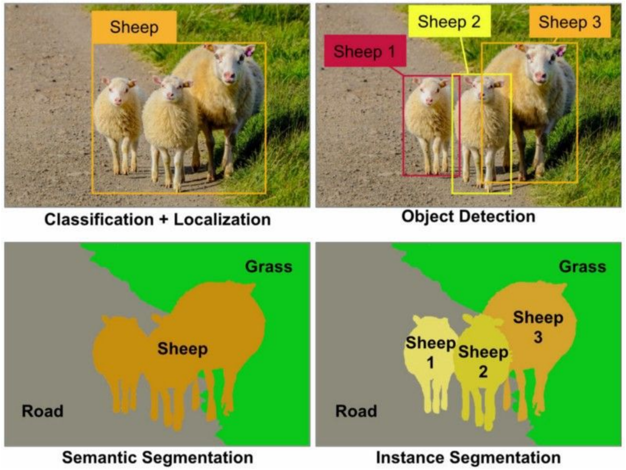
- Example of Deep Learning applications
- Machine Learning vs Deep Learning
- Single Perceptron/Neuron
- Basic Neural Network
- How to train?
- Convolutional Neural Networks
- Recurrent Neural Networks

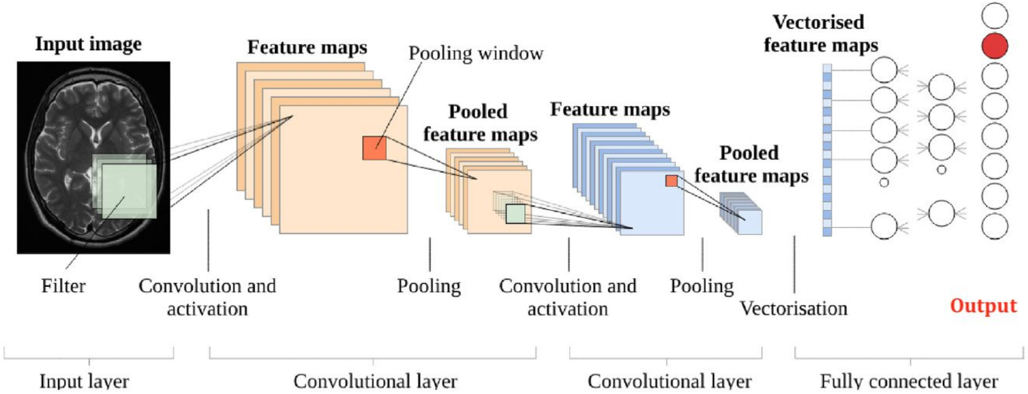


What can we do with neural networks?

Image
- classification
- localization
- captioning
- segmentation

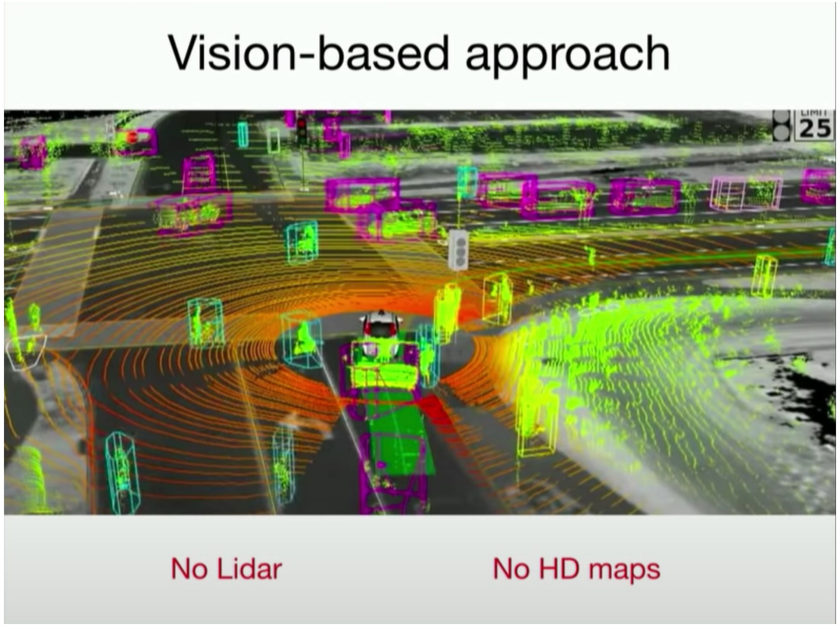






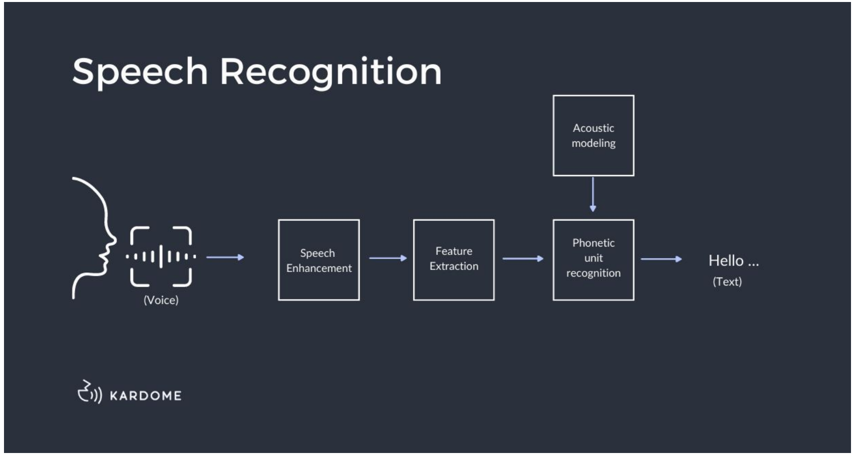
What we can do with neural networks?

- Self Driving Cars

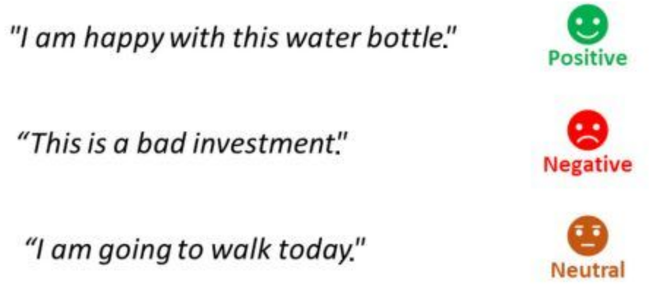


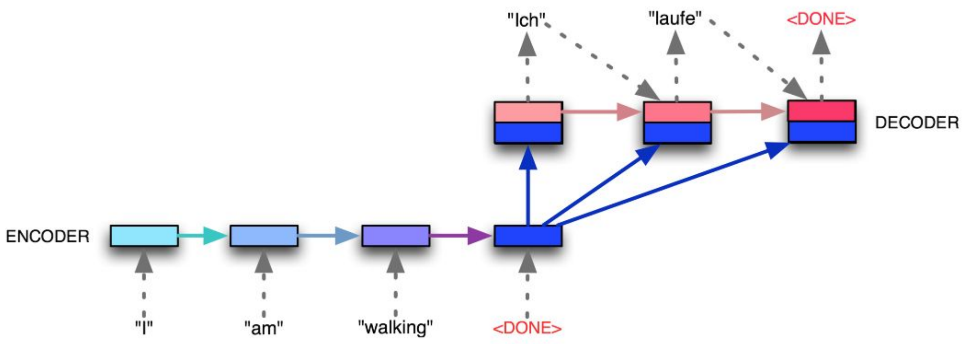
What we can do with neural networks?

- Speaker classification, Speech2Text translation, Voice assistant (Siri, Alexa)



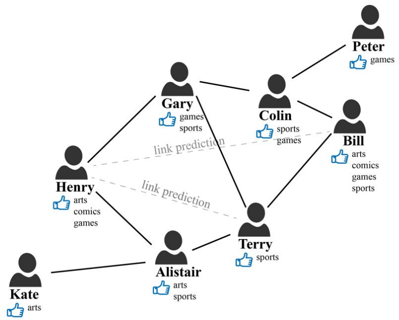
- Language translation (Google Translate)
- Sentiment analysis

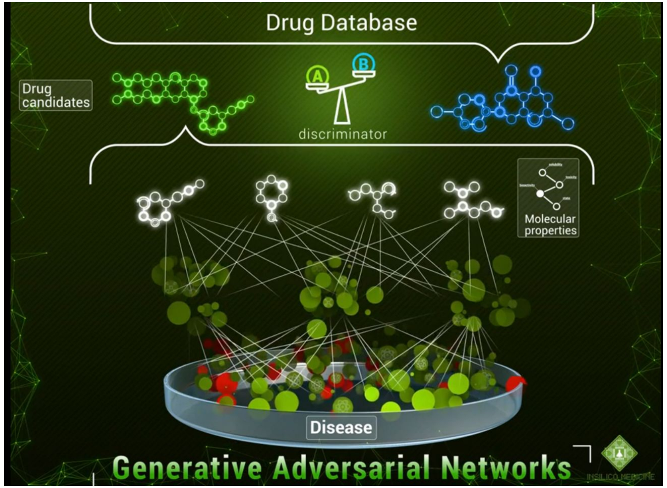




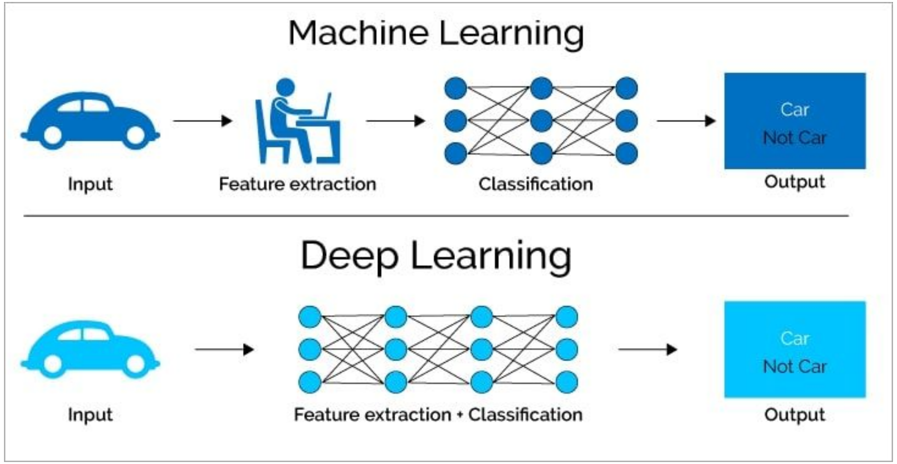
What we can do with neural networks?

- Drug discovery
- Social network analysis
- Movie recommender system (e.g. Netflix)
- And many others …



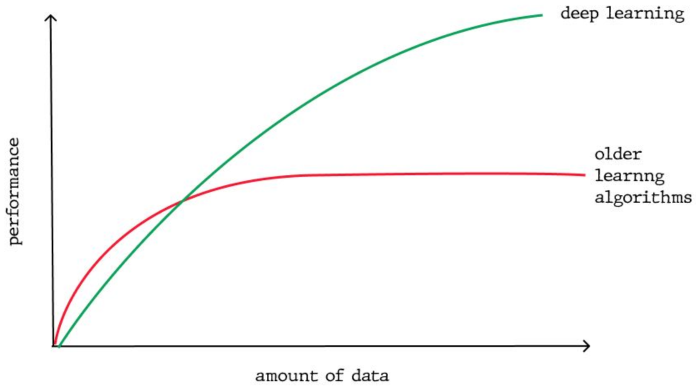


Machine Learning vs Deep Learning



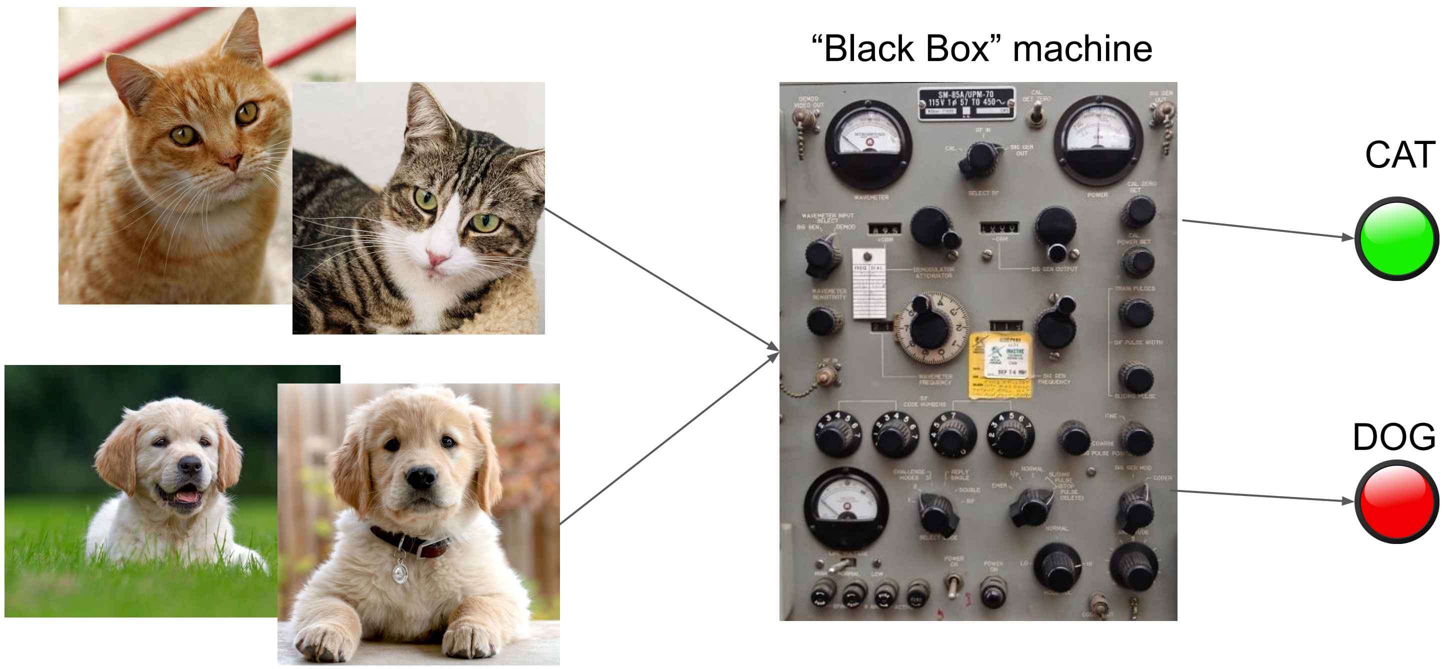
- requires manual feature engineering
+ small datasets posible

+ NO manual feature engineering
- requires large datasets !!!



Supervised learning

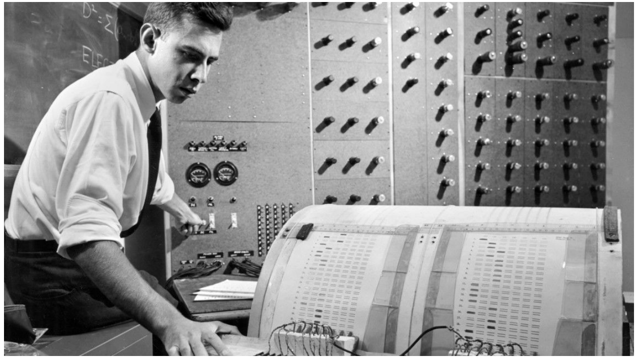
- Manual feature engineering is difficult! Fox example, how can you describe dog or cat algorithmically?
- Train a machine by showing examples instead of programming it
- When the output is wrong, tweak the parameters of the machine

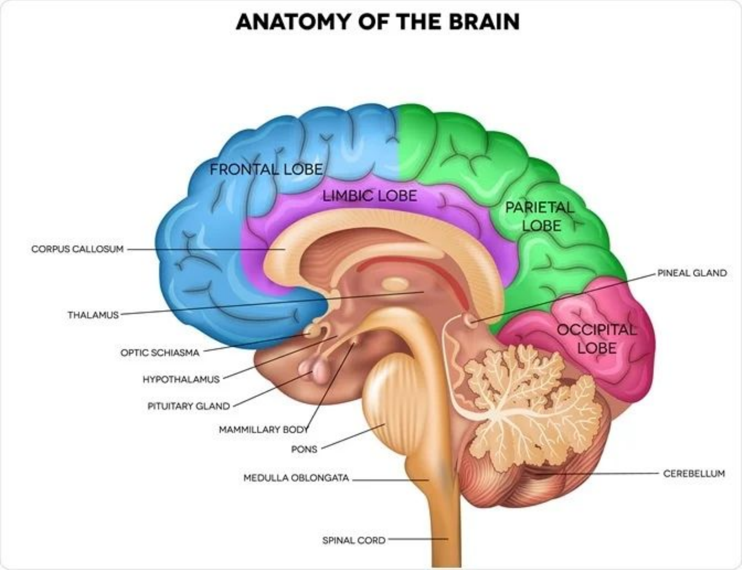


How should we design the “BlackBox” machine?

Can we design the “Black Box” as a human brain?

- First neural networks were originally designed to approximate human brain/neurons (Perceptron, Rosenblatt 1957)

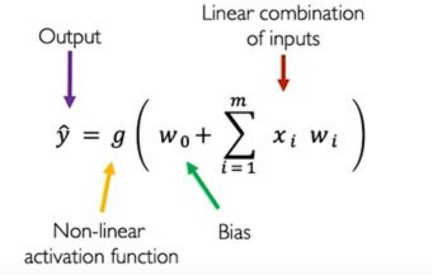


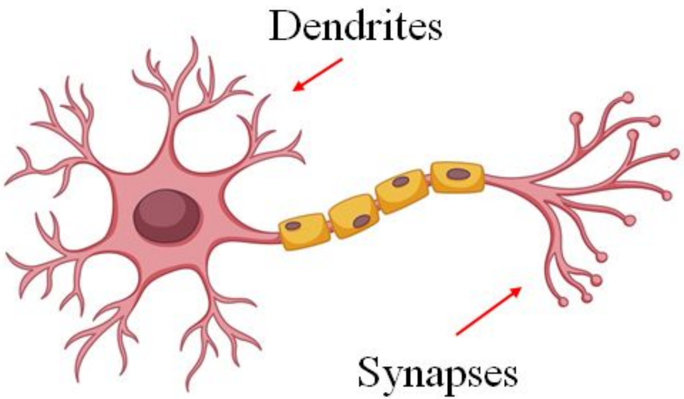


- More details in Neuroscience vs AI presentation

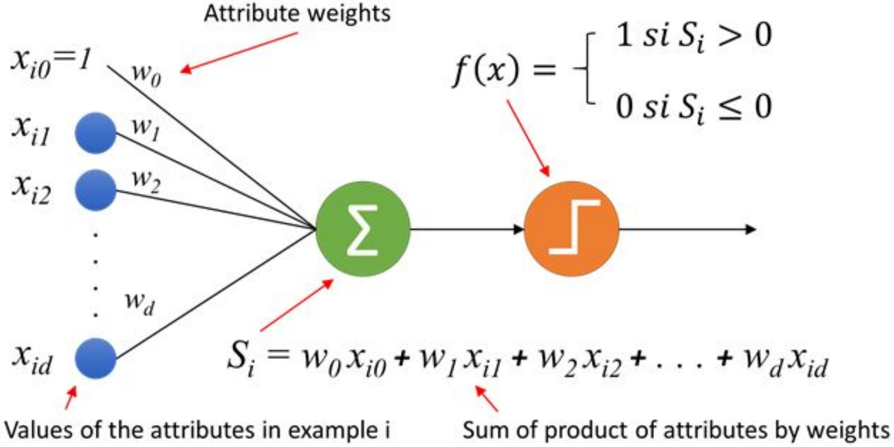
Basic Block of NN: Perceptron

Perceptron = simple mathematical model of biological neuron





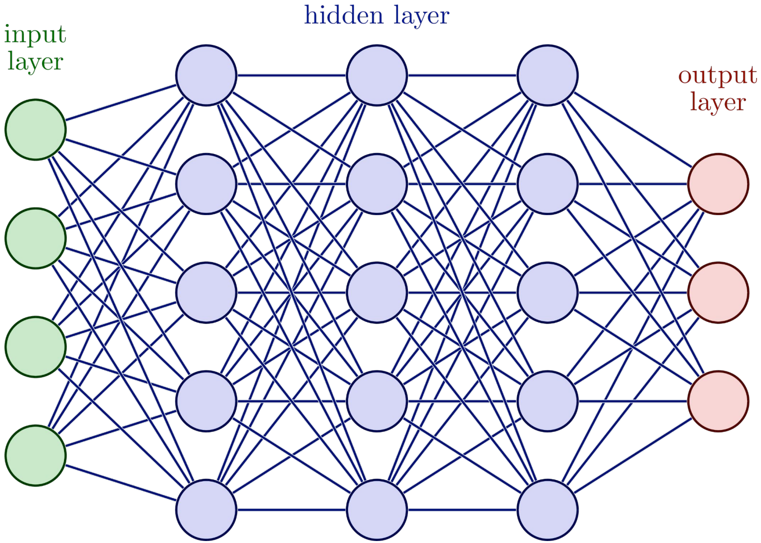
NEURON

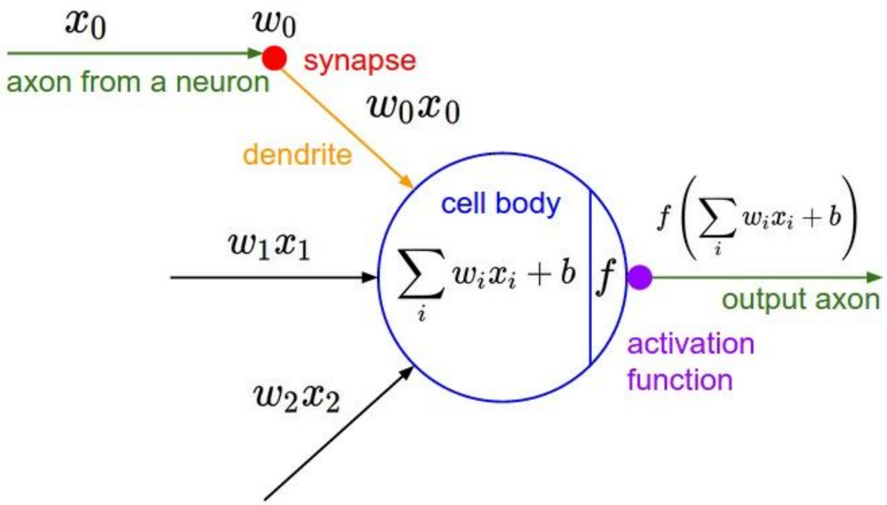


PERCEPTRON

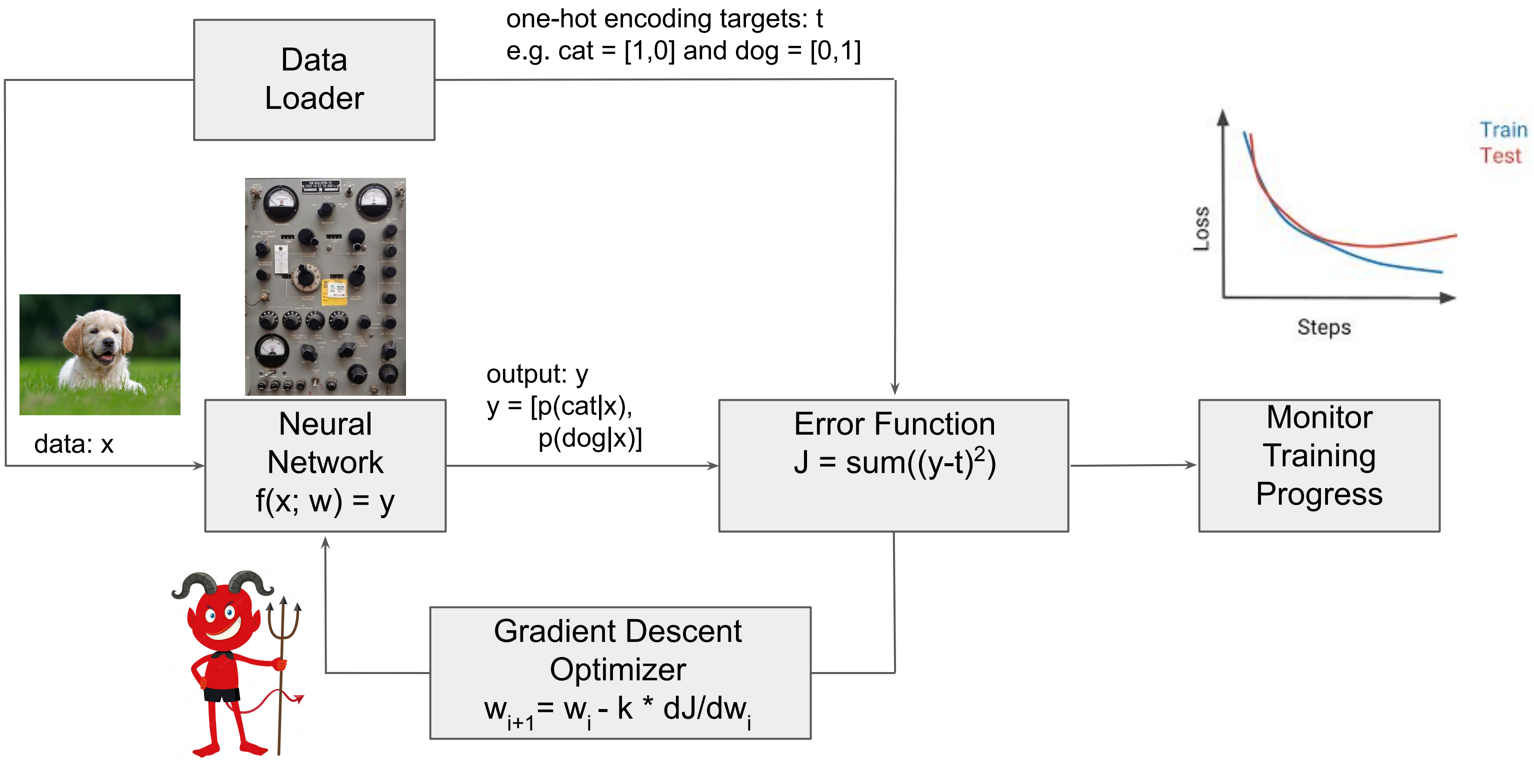
Neural Networks

- Neural network = Collection of neurons organized in multiple layers
- Mathematically speaking, neural network allows finding of approximation of any function y= f(x) that can map inputs (x) to outputs (y)
- In theory, large enough NN can be used to approximate any function => Universal Approximation Theorem!





Basic building blocks

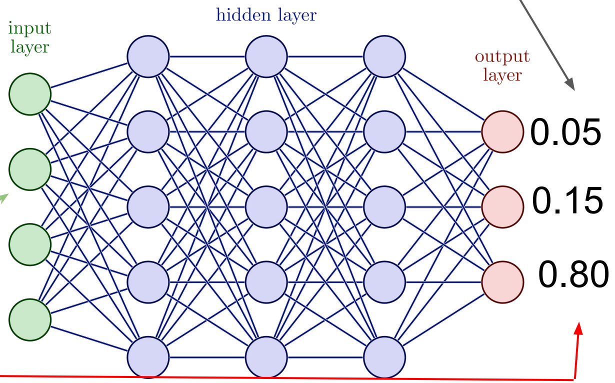


Example, shallow NN for Iris classification

Table with features

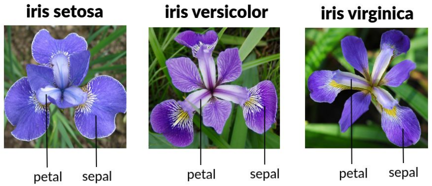
sepal		petal		class
length	width	length	width	
6.3	2.3	4.4	1.3	versicolor
6.2	3.4	5.4	2.3	virginica
5.2	3.4	1.4	0.2	setosa
6.9	3.1	5.4	2.1	virginica
5.7	4.4	1.5	0.4	setosa
5.4	3.7	1.5	0.2	setosa
5	3.3	1.4	0.2	setosa
6.4	2.8	5.6	2.1	virginica
6	3	4.8	1.8	virginica
5.5	2.5	4	1.3	versicolor

Output Probabilities



target = [0,0,1]

one-hot encoding:
setosa = [1,0,0]
versicolor = [0,1,0]
virginica = [0,0,1]



- Feed line by line to your NN model
-

Training

- NN are trained by Backpropagation and Gradient descent (difficult math needed)
- Basically, iteratively adapting “Black Box” to minimize objective error function

one-hot encoding targets: t
e.g. cat = [1,0] and dog = [0,1]

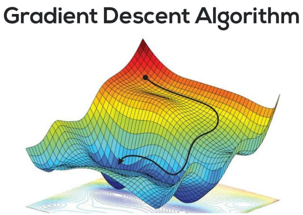
data: x

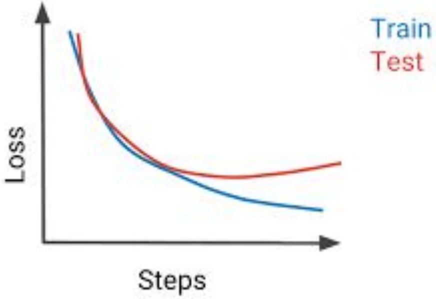
Neural Network
$f(x; w) = y$

output: y
y = [p(cat), p(dog)]

Error/Loss Function
$J = sum((y-t)^2)$

Monitor Training Progress





Gradient Descent Optimizer
$w_{i+1} = w_i - k * dJ/dw_i$

Neural network playground
https://playground.tensorflow.org/

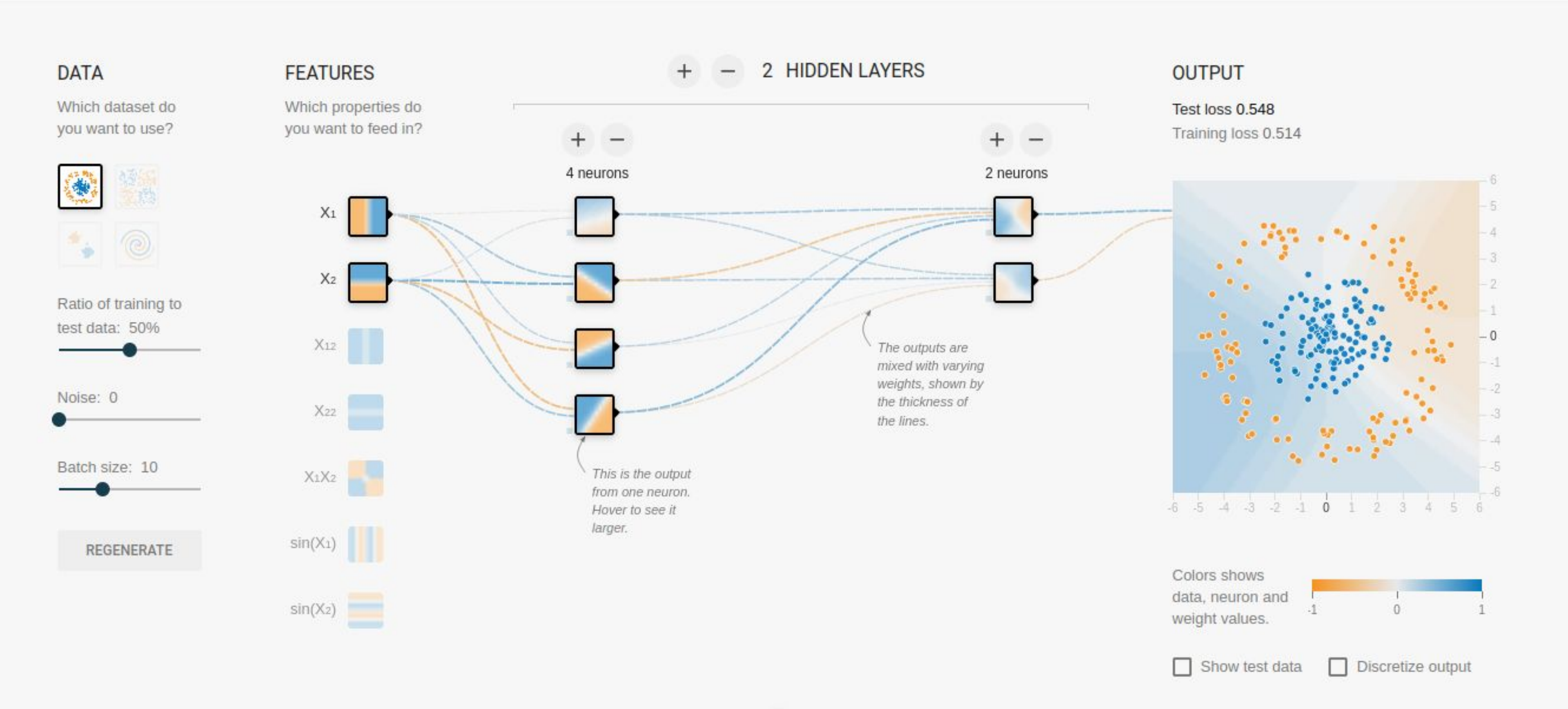
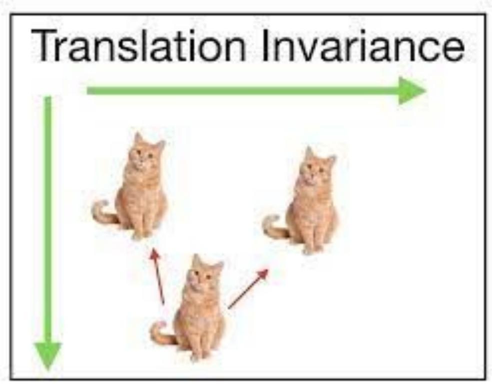
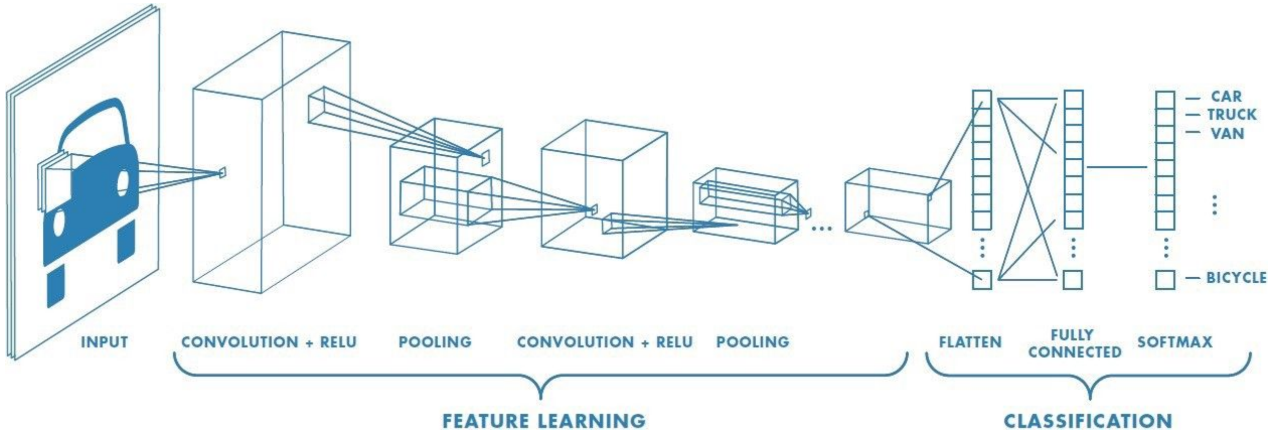


Image classification => Convolutional Neural Network

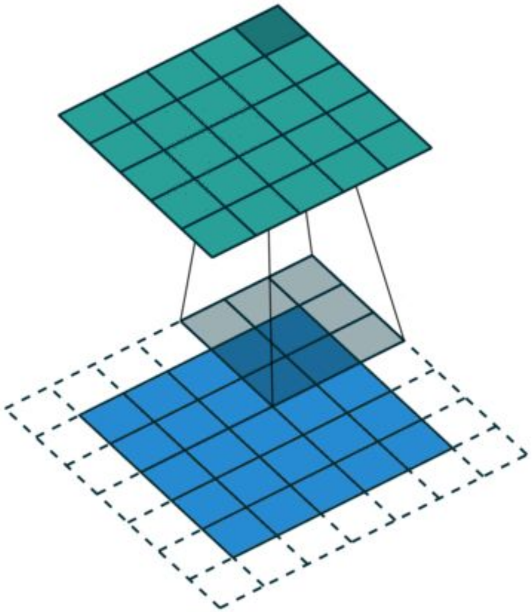
- NN must be invariant for translation, i.e. it shouldn't matter where the cat is in the image
- Convolutions are used to extract translation invariant features

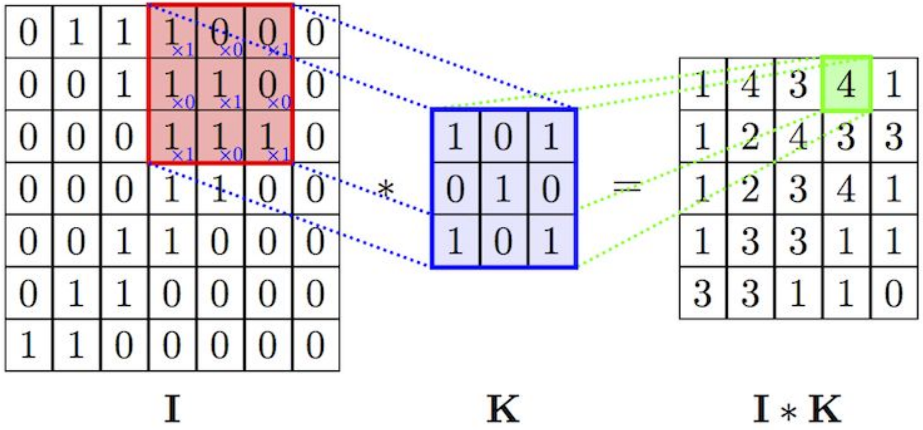


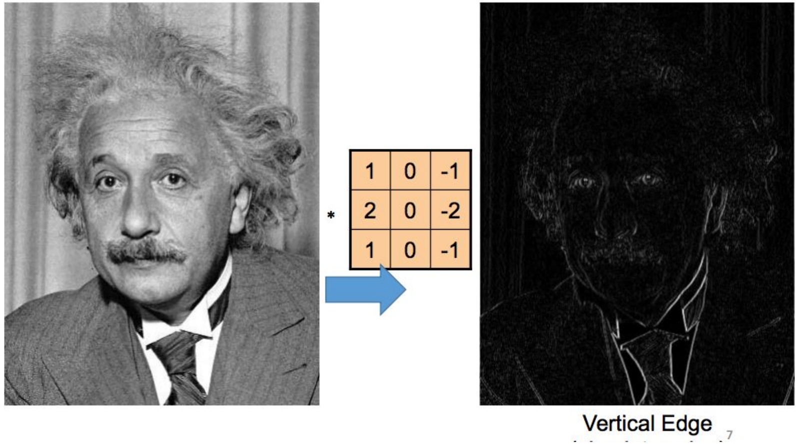


Convolution

- sliding the kernel “K” over the Image “I”
- Basically, multiply corresponding matrix cells and sum the results

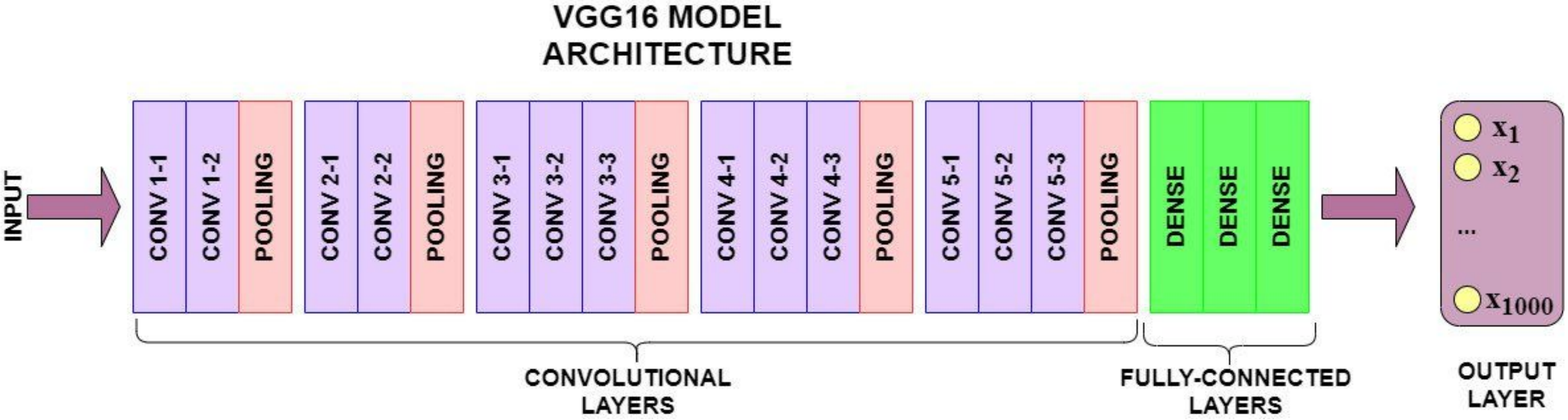


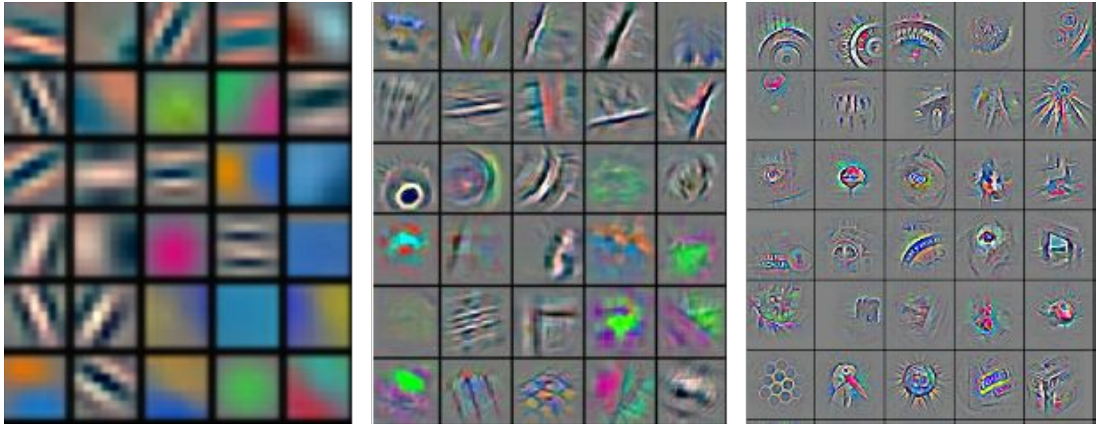




$$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-i, n-j]$$

Convolutional Neural Network Architecture

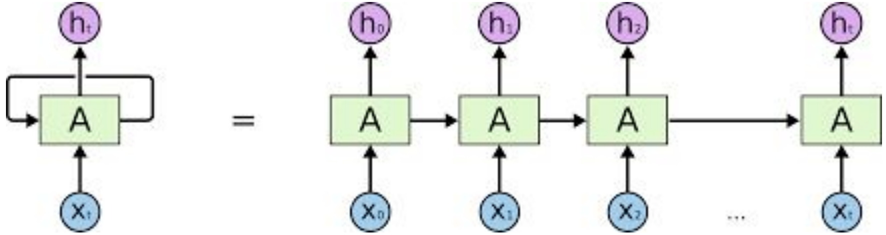


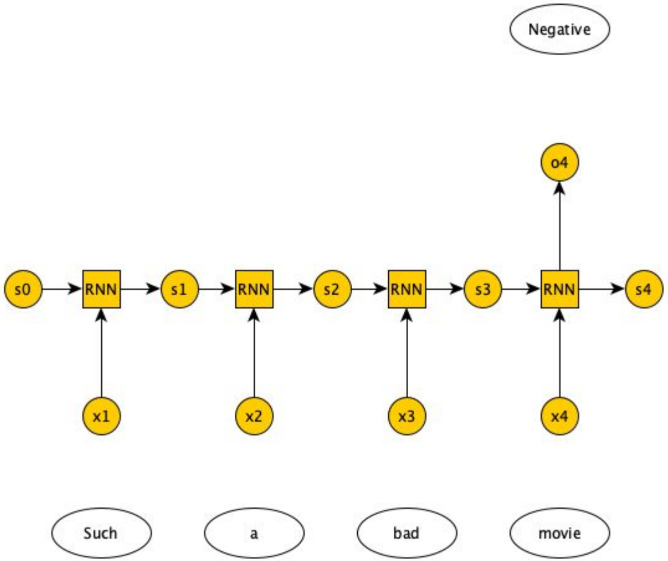


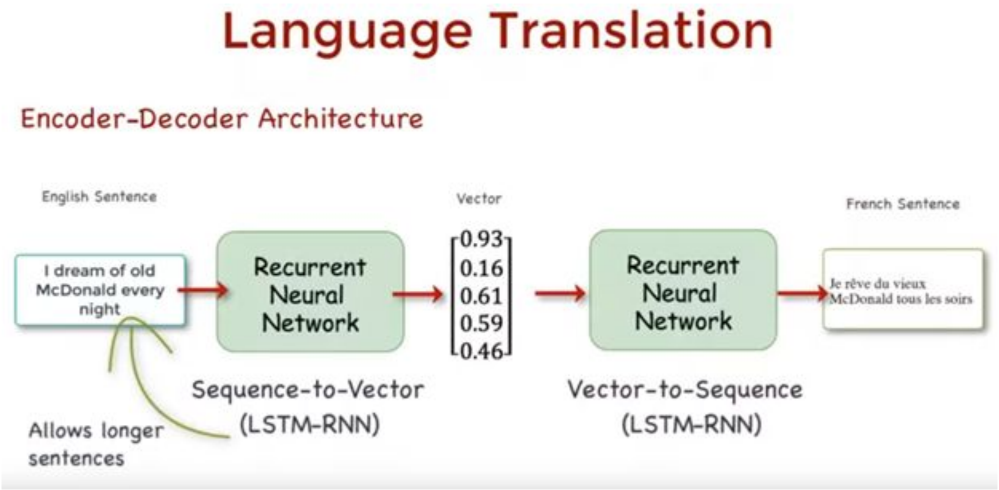
Convolutional kernels gets more complex with increasing number of convolutional layers

Recurrent Neural Networks

- Neurons have recurrent connections
- Works great for sequence
  e.g. sentiment analysis, machine language translation








THANK YOU FOR YOUR ATTENTION
ANY QUESTIONS?
makeameme.org