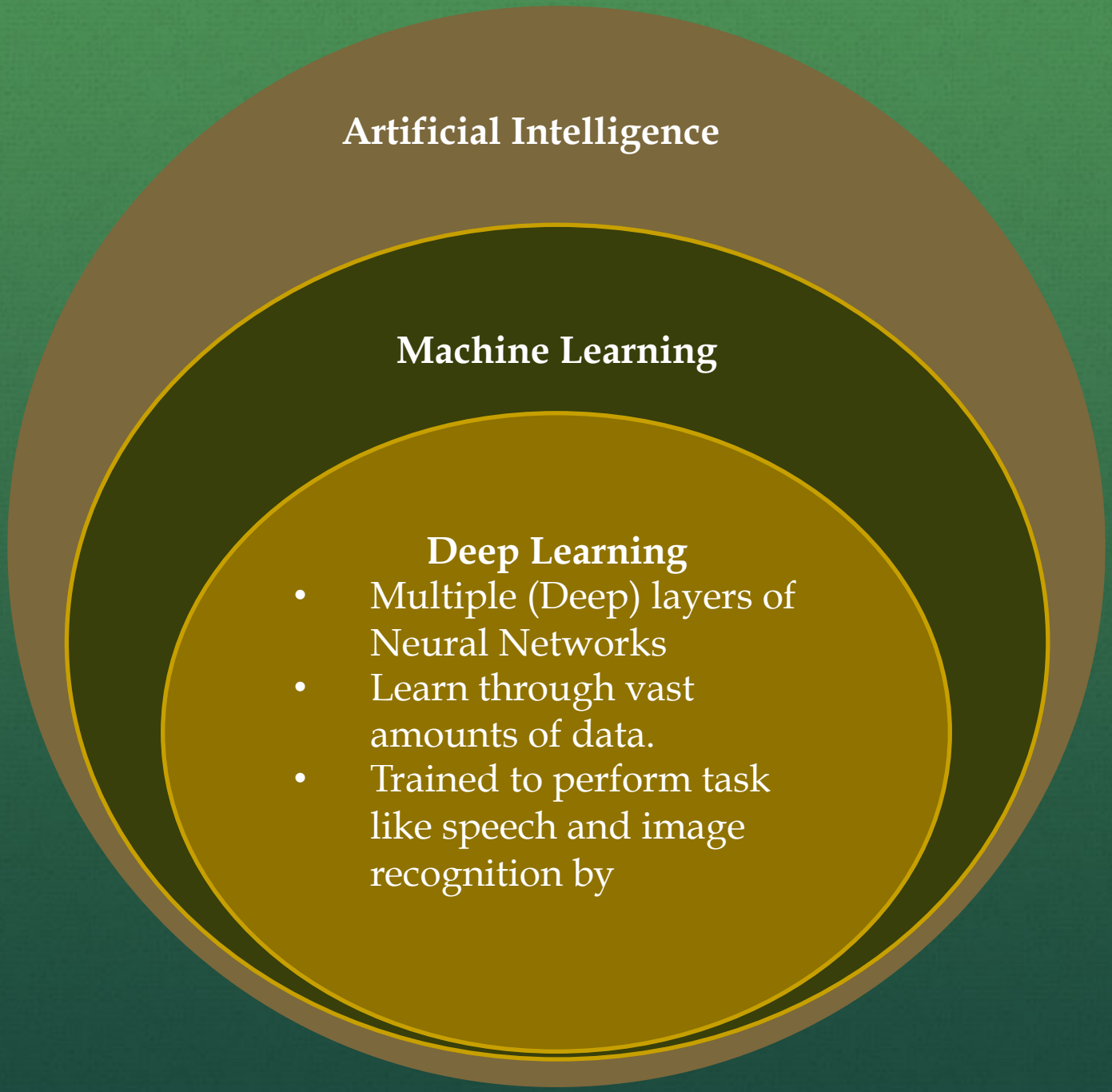
Convolutional Neural Networks

Dr Amita Kapoor, Nurture AI


Artificial Intelligence
Machine Learning
Deep Learning
• Multiple (Deep) layers of Neural Networks
• Learn through vast amounts of data.
• Trained to perform task like speech and image recognition by

Neural Networks: Biological Inspiration

To make the computers more robust and intelligent.
We take inspiration from the intelligent machine ever made

Human Brain

https://www.dreamstime.com/royalty-free-stock-images-human-brain-close-up-active-neurons-image18466049#

Features of the Brain

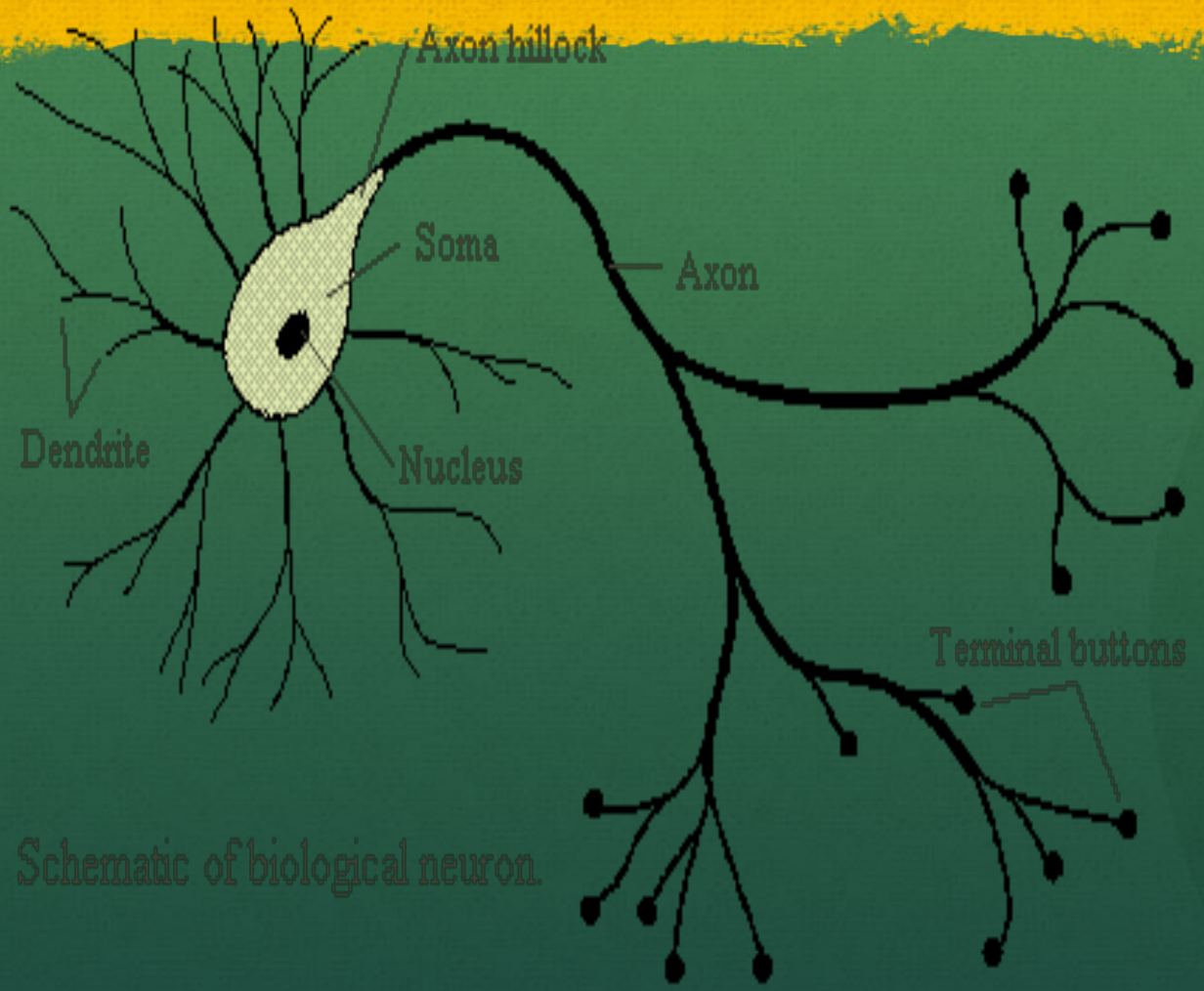
- Ten billion (10^{10}) neurons
- Neuron switching time >10^{-3}secs
- Face Recognition ~0.1secs
- On average, each neuron has several thousand connections
- Hundreds of operations per second
- High degree of parallel computation
- Distributed representations
- Compensated for problems by massive parallelism
- Graceful Degradation and Robust

How do we do it?

- The brain is a collection of about **10 billion** interconnected neurons.
- Each neuron is a cell that uses biochemical reactions to **receive**, **process** and **transmit** information.

https://giphy.com/gifs/uofcalifornia-brain-neuroscience-neurons-xT0BKr4MvHdohFTe6s/

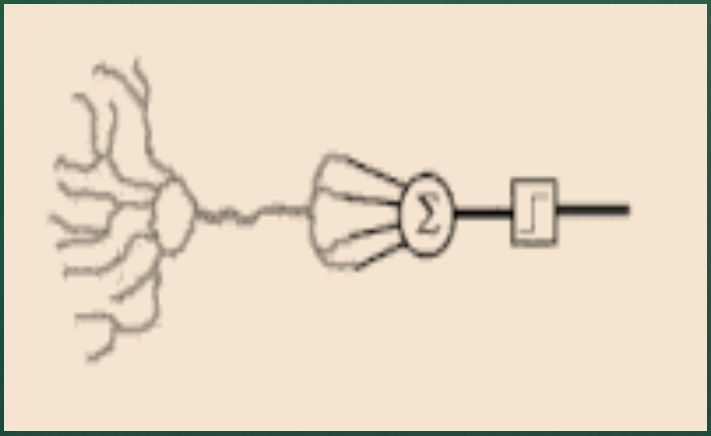
How do we do it?

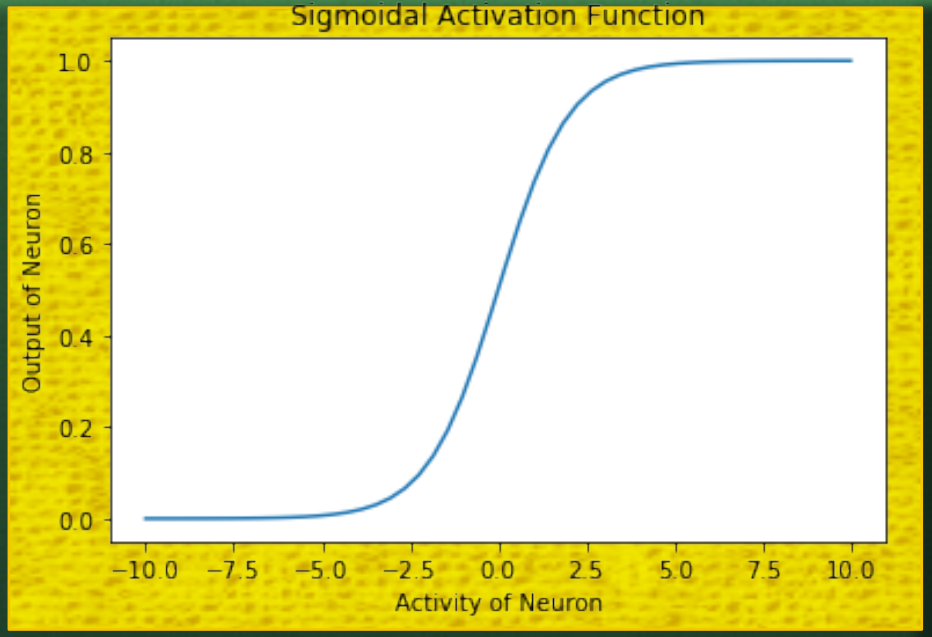


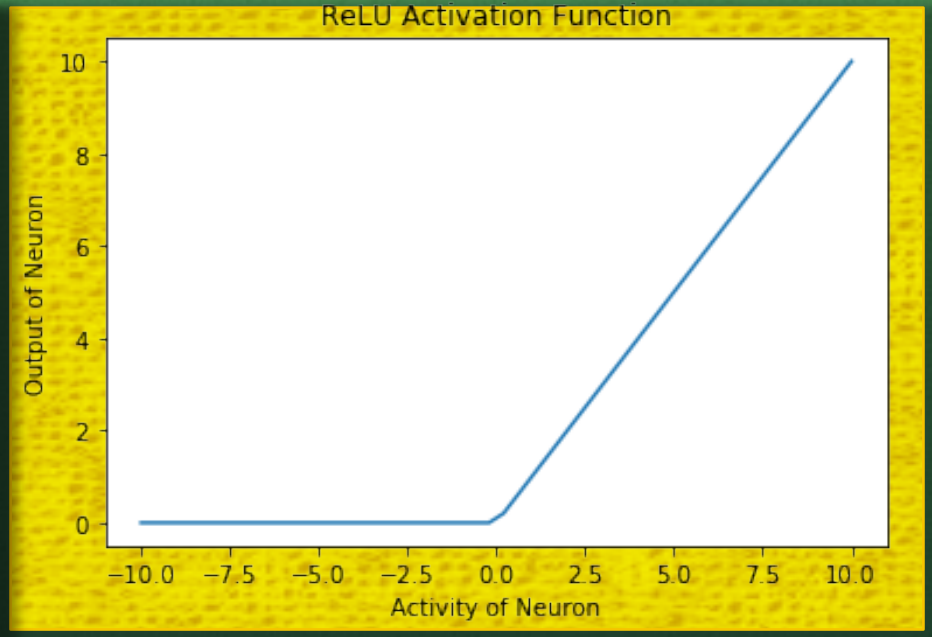
http://serendip.brynmawr.edu/exchange/brains/definition/def-neuron

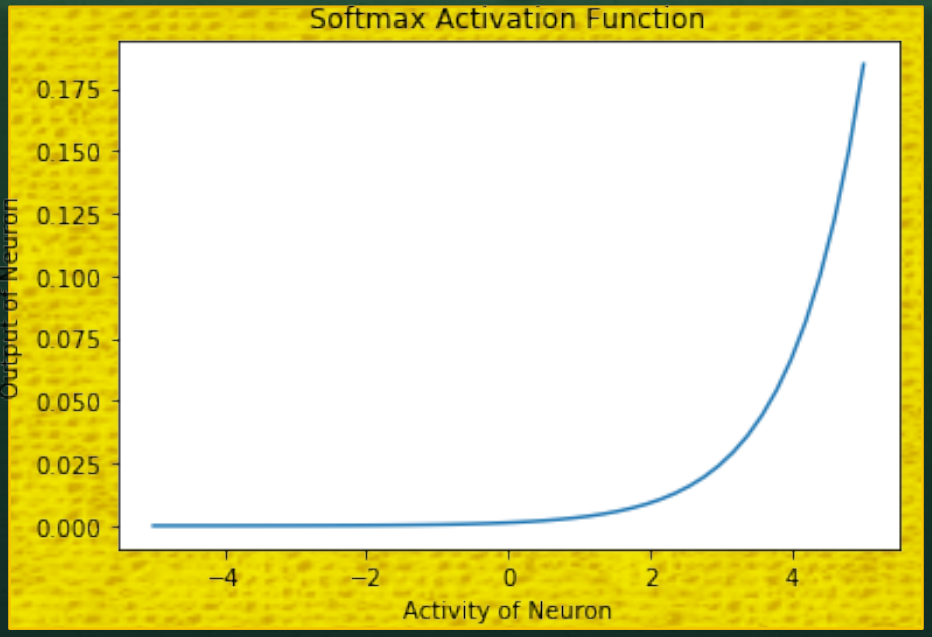
Artificial Neuron

- Inputs I
- Weights W
- Activity
- Activation function



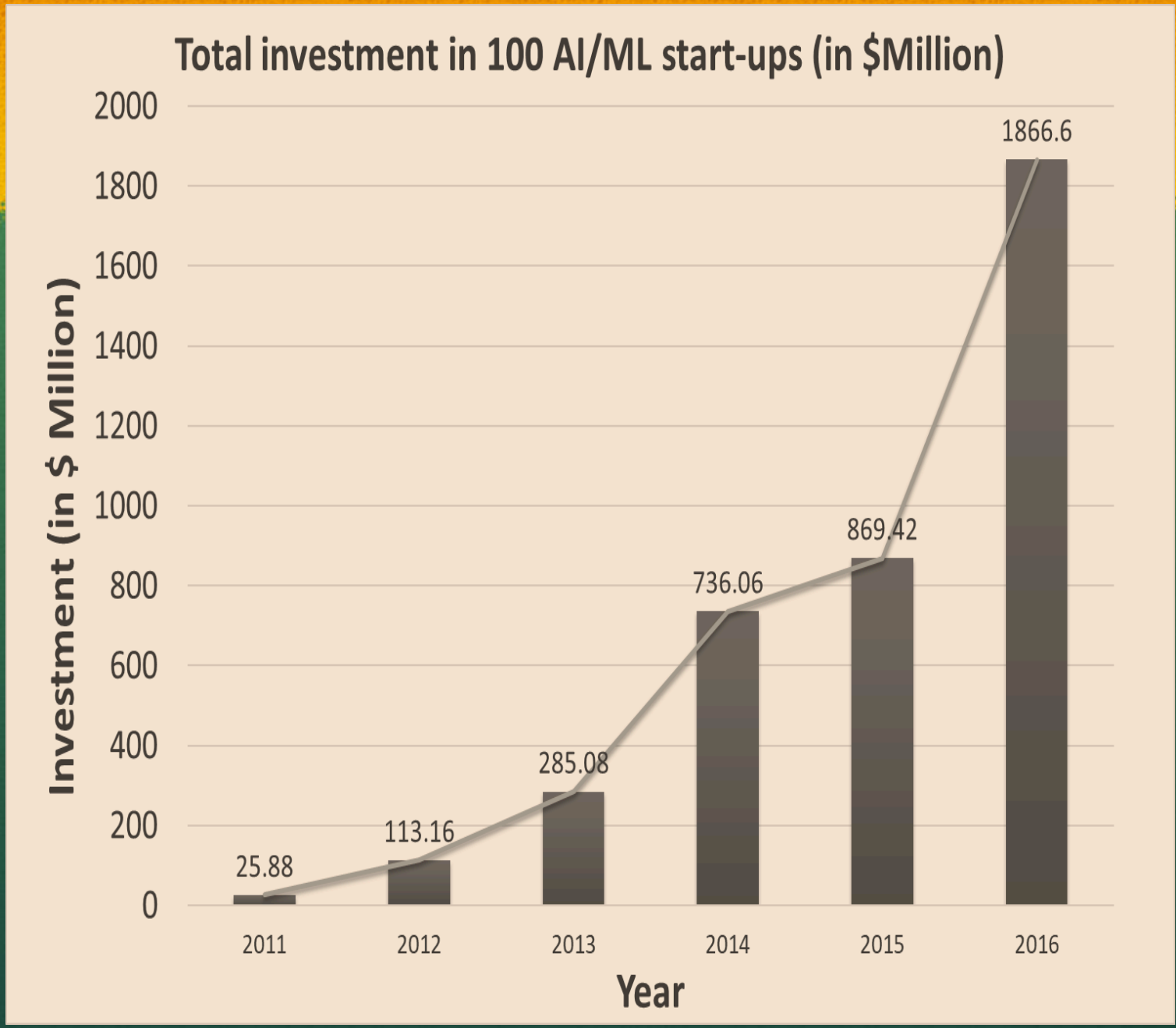

Sigmoidal Activation Function
Output of Neuron
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Activity of Neuron

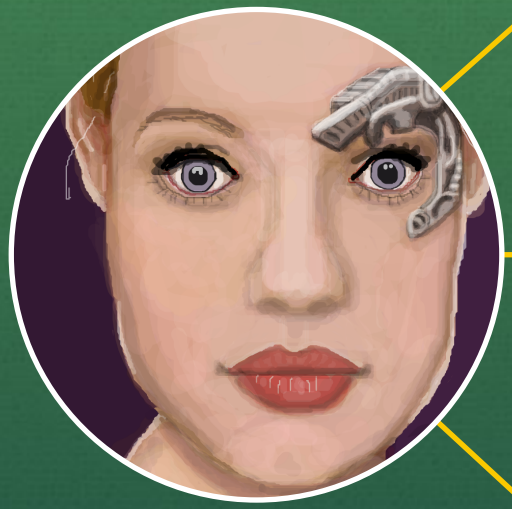

ReLU Activation Function
Output of Neuron
10
8
6
4
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Activity of Neuron


Softmax Activation Function
Output of Neuron
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−4
−2
0
2
4
Activity of Neuron

Why Deep Learning

- Deep Learning is achieving state-of-the art results across a range of difficult problem domains.
- It is about action
- Download Tensorflow, Keras and you can build your first neural network model in 5 minutes.
    - Only using four commands model (add, compile, fit, predict)


Total investment in 100 AI/ML start-ups (in $Million)
Investment (in $ Million)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
25.88
113.16
285.08
736.06
869.42
1866.6
2011
2012
2013
2014
2015
2016
Year



Supervised Learning: Classification/ Regression

- Multilayered Perceptrons (MLP)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Deep Belief Networks

Unsupervised Learning: Dimensionality Reduction/ Generating Data

- Self Organized Maps (SOM)
- Restricted Boltzmann Machines (RBM)
- Autoencoders
- Generative Adversial Networks (GANs)
- Deep Belief Networks

Reinforcement Learning: Robotics/ Inventory Management

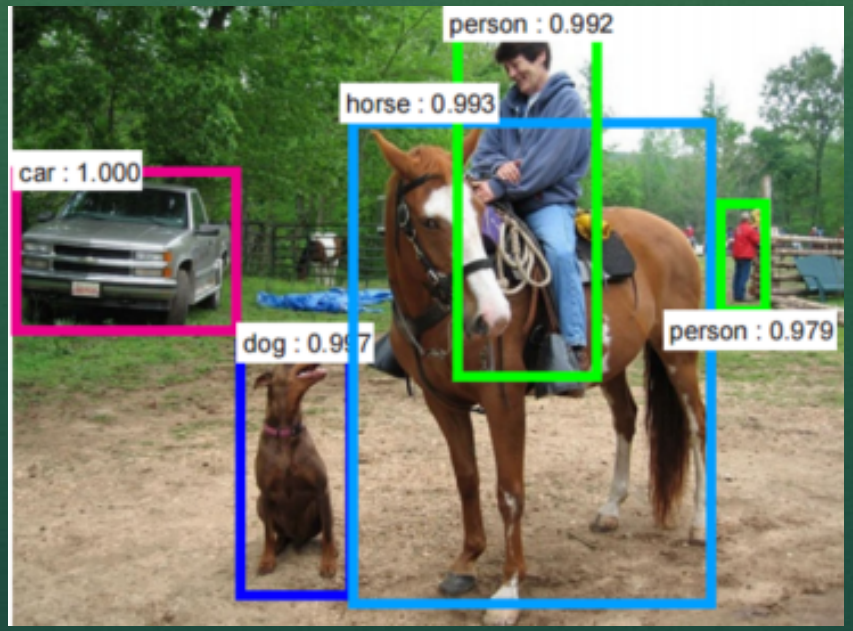
- Q- learning
- Monte Carlo Methods
- Temporal Difference Methods

Deep Learning: Success Stories

- Automatic Colorization of Black and White Images
- Automatically Adding Sounds To Silent Movies
- Automatic Machine Translation
- Object Classification and Detection in Photographs
- Automatic Handwriting Generation
- Automatic Text Generation
- Automatic Image Caption Generation
- Automatic Game Playing

CNN

- A type of Neural Networks.
- Effective in areas like image recognition and classification.



History

- Was proposed first by Yann LeCun in 1988 for the recognition of handwritten digits (MNIST).

Image Source: http://yann.lecun.com/

CNN-Architecture

- CNN consists of four main parts:
  - Convolution
  - Non Linearity
  - Pooling
  - Classification

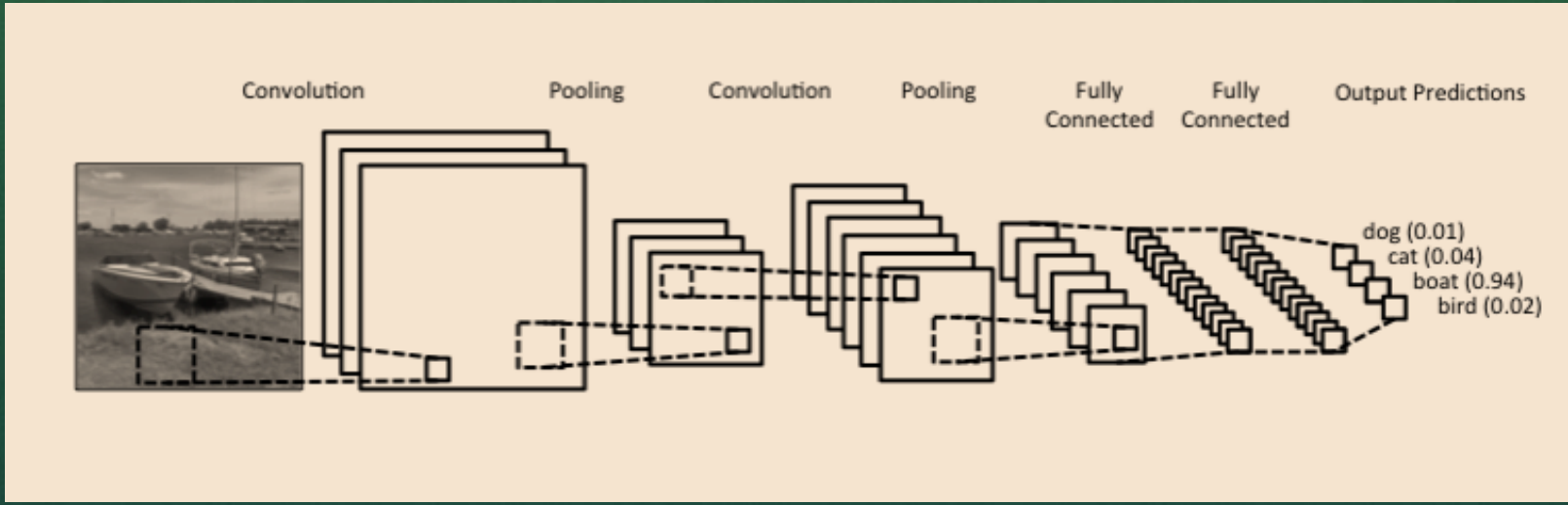
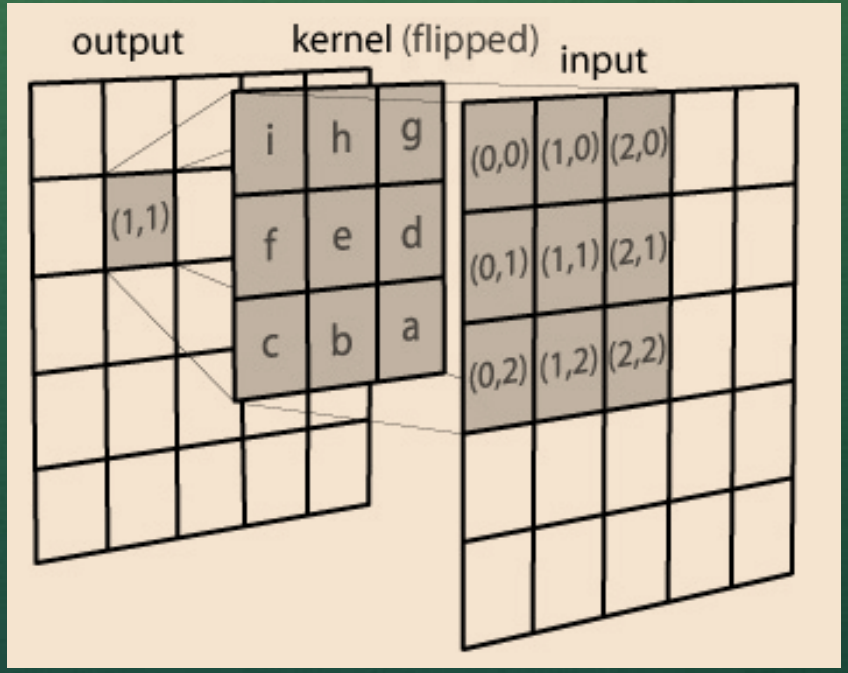


Image Source: https://www.clarifai.com/technology

Convolution and filters

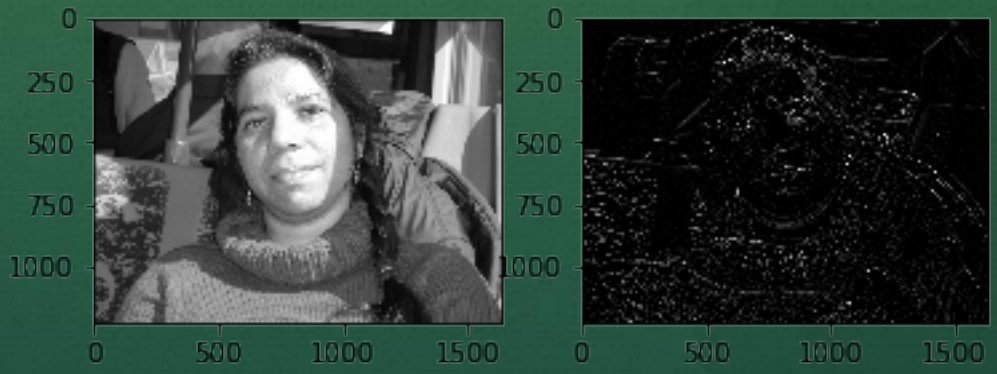
- Convolution is an old signal processing trick
- Process of adding each element of the image to its local neighbors, weighted by the kernel (filter).
- Traditionally it involves flipping both the rows and columns of the kernel and then multiplying locally similar entries and summing.

a	b	c
d	e	f
g	h	i



Convolution and filters

```
kernel = np.array([[ 1,  2,  1],
                   [ 0,  0,  0],
                   [-1, -2, -1]])
k2 = np.flip(np.fliplr(kernel),0)
filtered = cv2.filter2D(src=image, kernel=k2, ddepth=-1)
plt.subplot(121)
plt.imshow(image, cmap=gray)
plt.subplot(122)
plt.imshow(filtered, cmap=gray)
```



Convolution in CNN

- No flip is needed

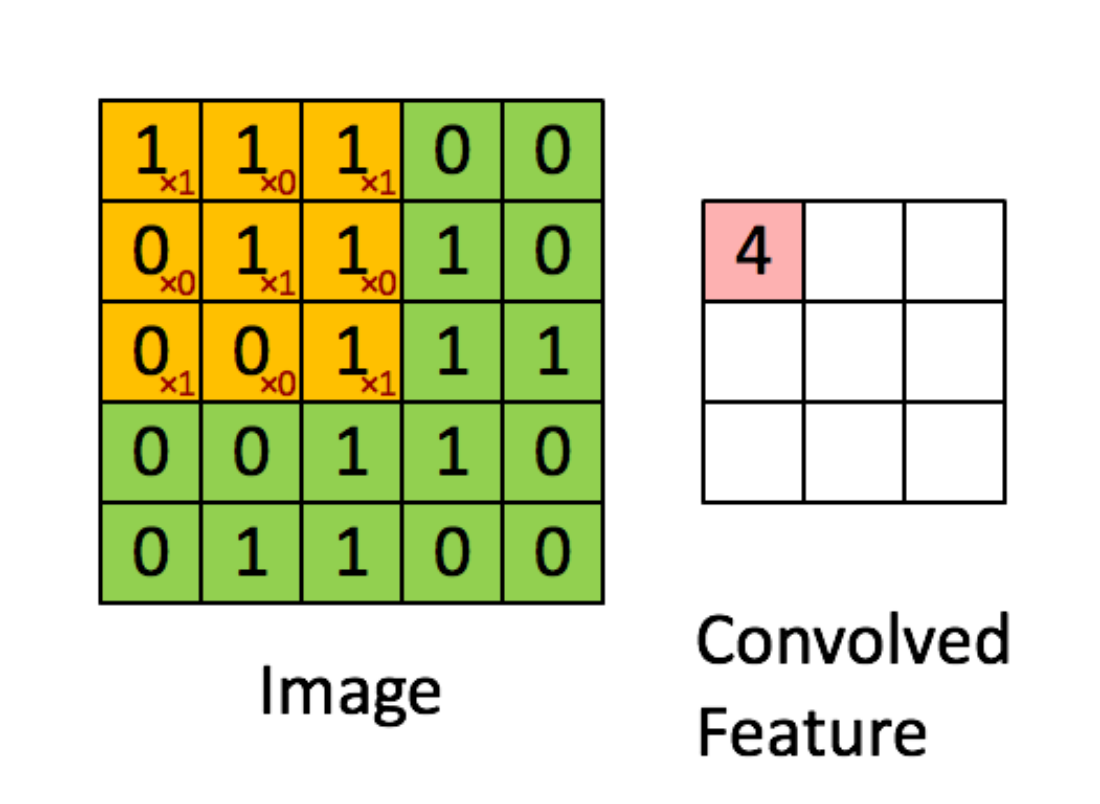


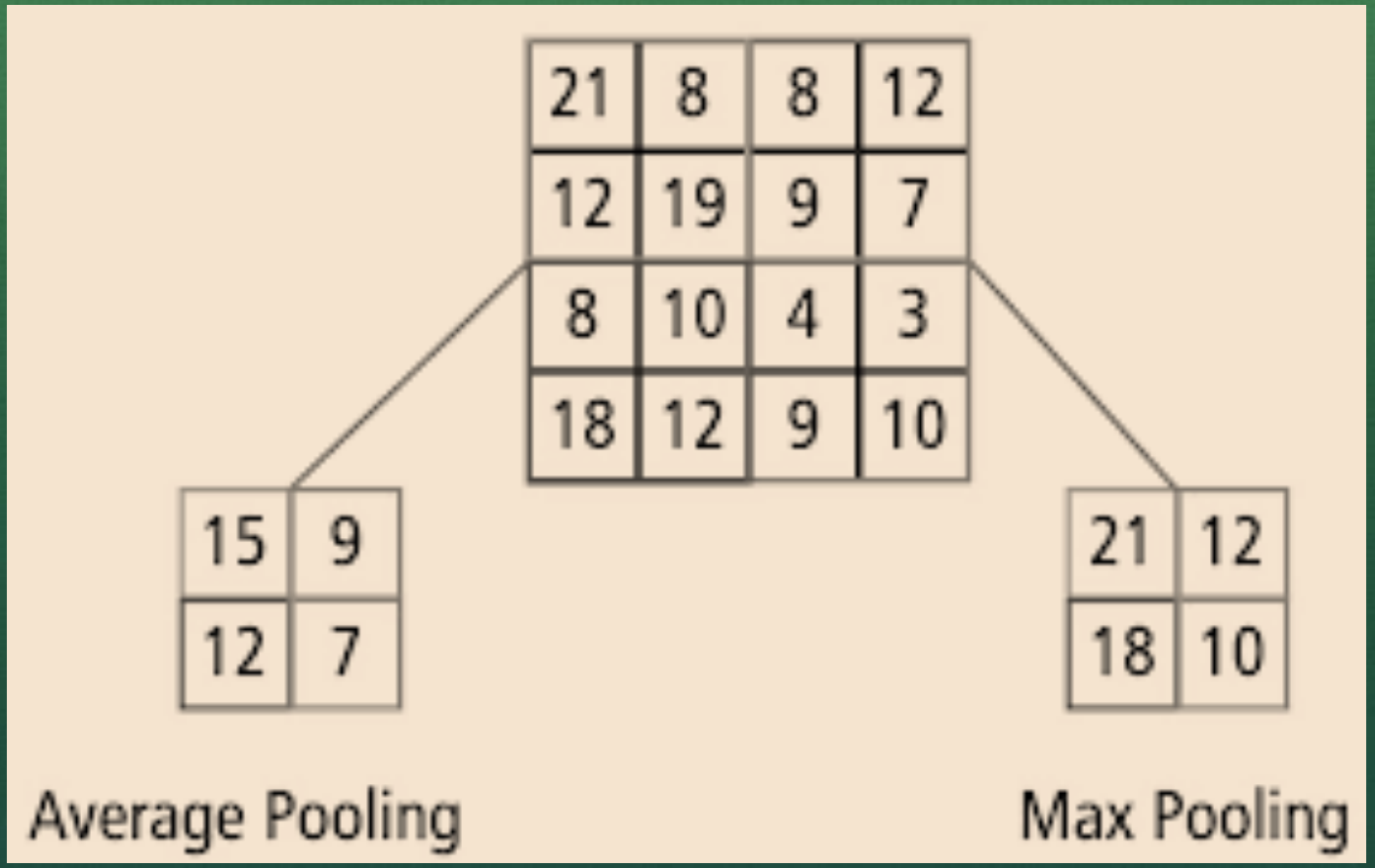
Image Source: http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/

Convolution in CNN

Convolution provides three important ideas that help improve machine learning systems

1. Sparse interactions: kernel smaller than input
2. Parameter sharing: The network has tied weights (shared)
3. Equivariant representations: parameter sharing causes equivariance to translation

Average/Max Pooling



Stride and padding

- Stride: the step of the convolution operation.
- It defines the shift in filter on an image at each step.
- When the stride is 1 then we move the filters one pixel at a time.
- It is convenient to pad the input volume with zeros around the border.
- The nice feature of zero padding is that it will allow us to control the spatial size of the output volumes.
- Both stride and padding are hyperparameters

Stride and padding

- The size of the output image is affected by stride size and by padding:

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size

Convolution

Max Pooling

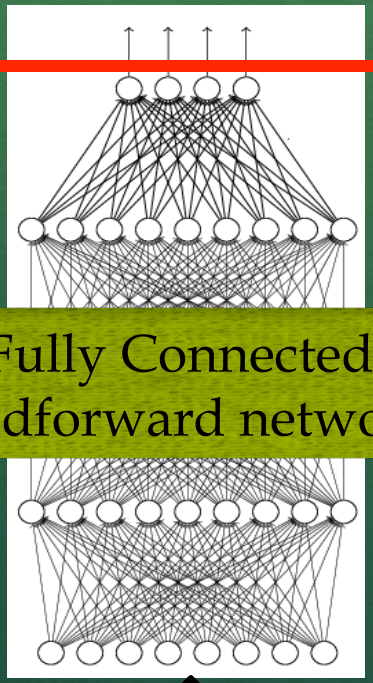
A new image

Convolution

Max Pooling

A new image

Flattened



Fully Connected Feedforward network

Cat/Dog …

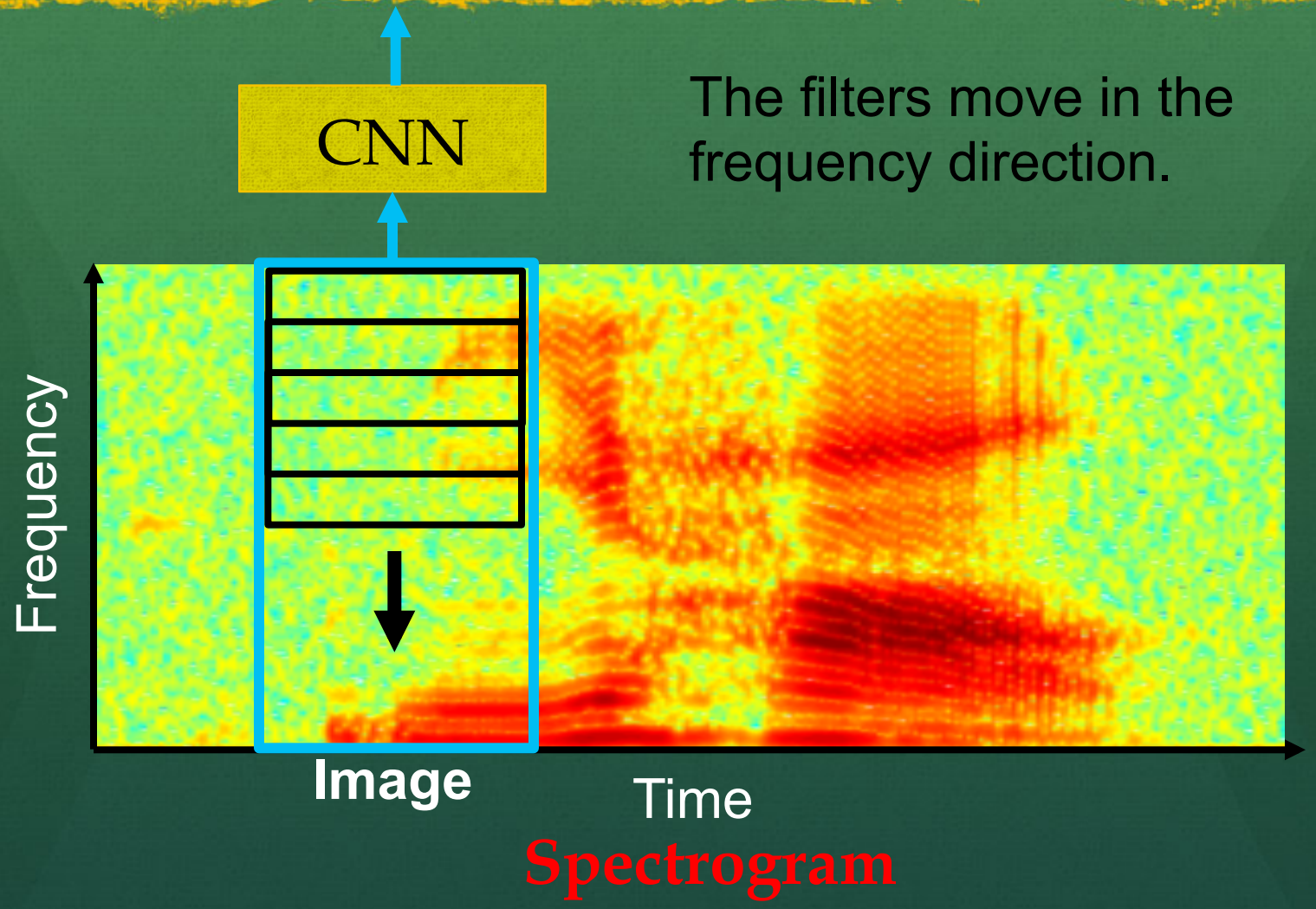
AlphaGo's policy network

The following is quotation from their Nature article:

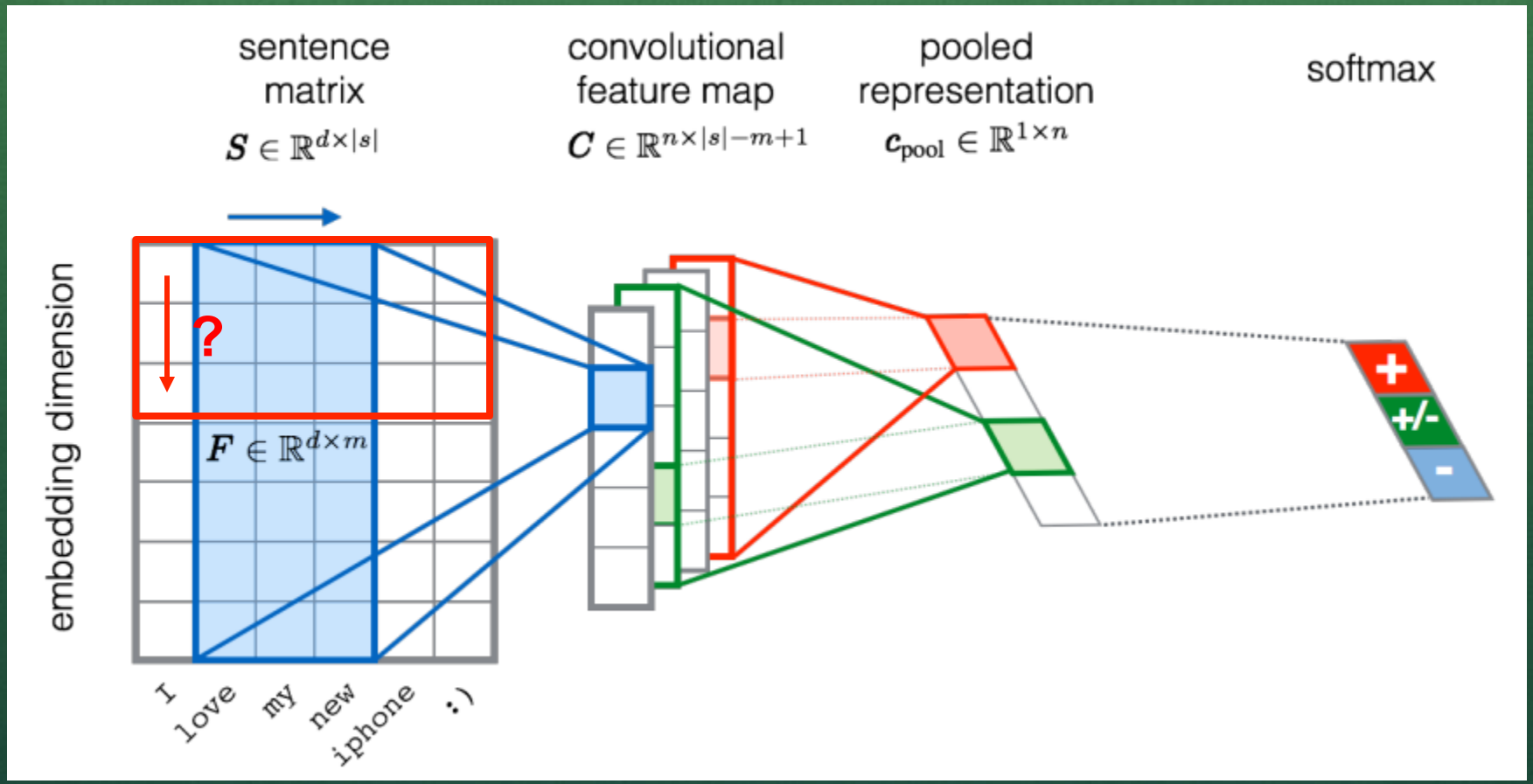
Note: AlphaGo does not use Max Pooling.

Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

CNN in speech recognition



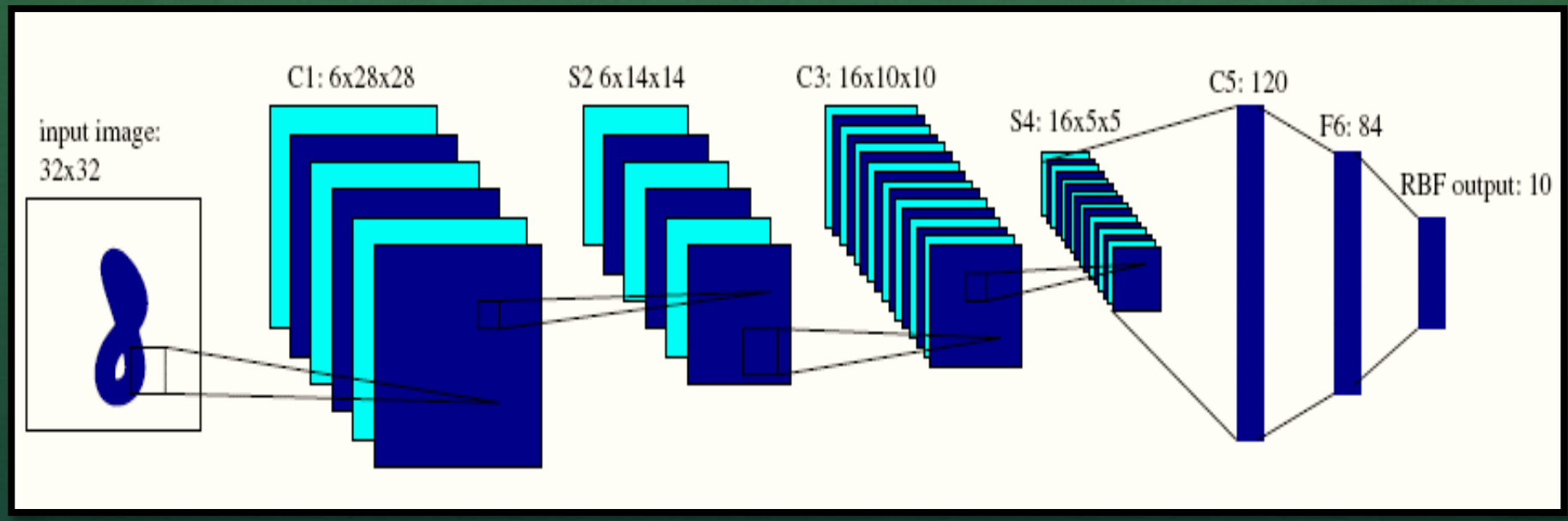
CNN in text classification



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

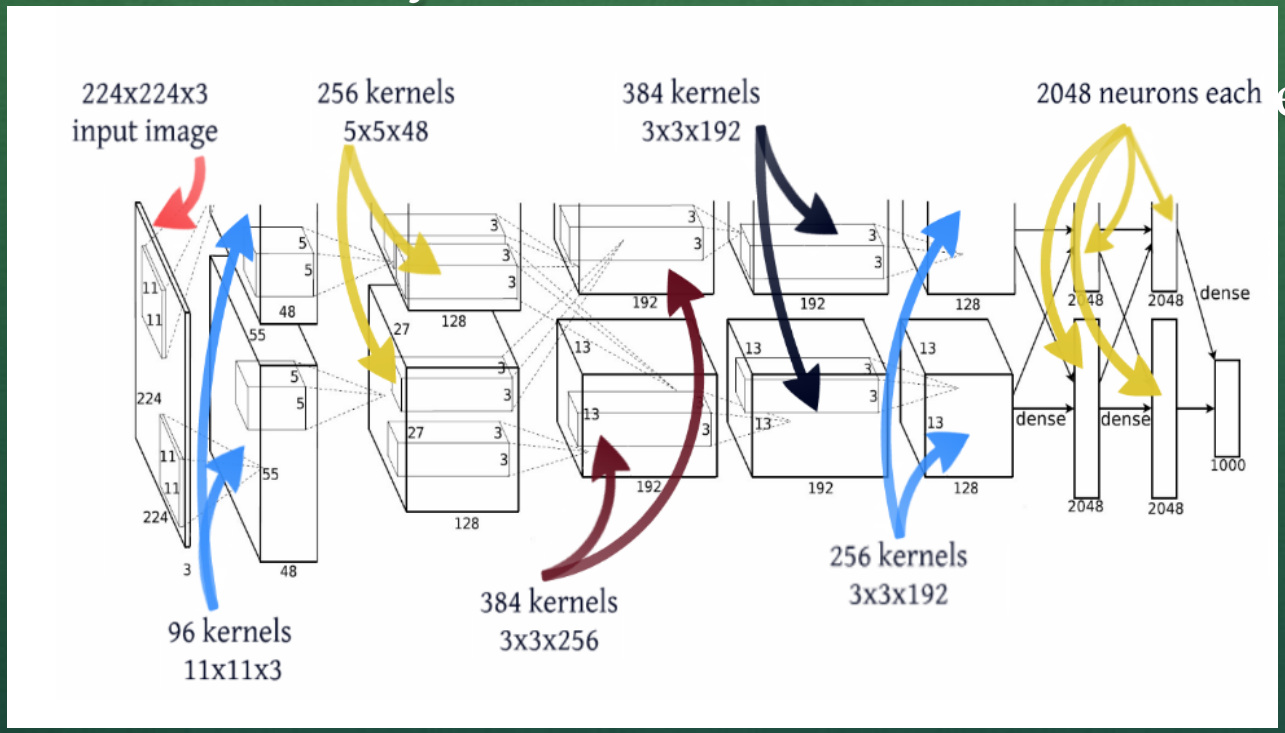
LeNet5

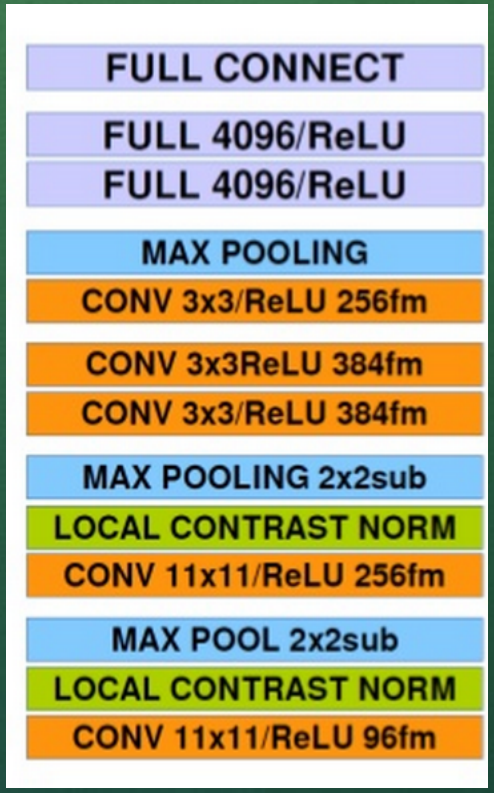
- 7 level CNN
- Recognize handwritten digits on cheques



Alexnet

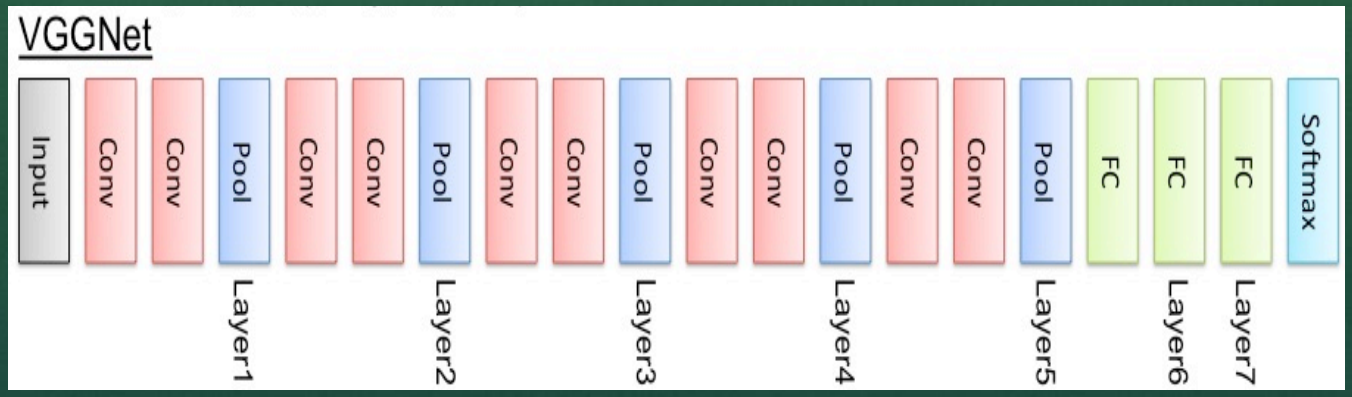
- Developed by Alex Krizhevsky, Ilya Sustskever and Geoffrey Hinton.





VGGNET

- Runner Up ILSVRC 2014
- Karen Simonyan and Andrew Zisserman
- To handle convergence on this deep network -> They trained first smaller versions of VGG first, and then used them as initialization for the deeper network – Pre Training
- VGG16 ~ 533MB
- VGG19 ~ 574MB



Inception

- Winner ILSVRC 2014
- GoogLeNet
- Inception Module: Acts as multi-level feature extractor
- Computes convolutions within the same module of the network.
- The output of these filters are then stacked along the channel dimension, before being fed into the next layer.
- Inceptionv3 ~ 96MB
- Inspired by Inceptionv3- Xception by François Chollet

ResNet

- Network in Network
- Just increasing layers is not sufficient.
- So they modified the Architecture and introduced Residual Learning.

Inceptionv4

- Latest in line: Combines Inception module with residual learning.

Overfitting

- Occurs when a statistical model describes random error or noise instead of the underlying relationship
- Exaggerate minor fluctuations in the data
- Will generally have poor predictive performance

Reducing Overfitting

- Data Augmentation

1. Image translation and horizontal reflection

   Randomly extracting patches

   Four corner and one center patches with reflection for testing

2. Altering the intensities of the RGB channels in training images

   Approximately captures an important property of natural images

   reduces the top-1 error rate by over 1%

Reducing Overfitting

- Dropout

  Zero the output of each hidden neuron with probability 0.5.

  No longer contribute to forward pass and backward propagation

  Neural network samples a different architecture every time

  Reduce complex co-adaptations of neurons

  Used in two fully-connected layers

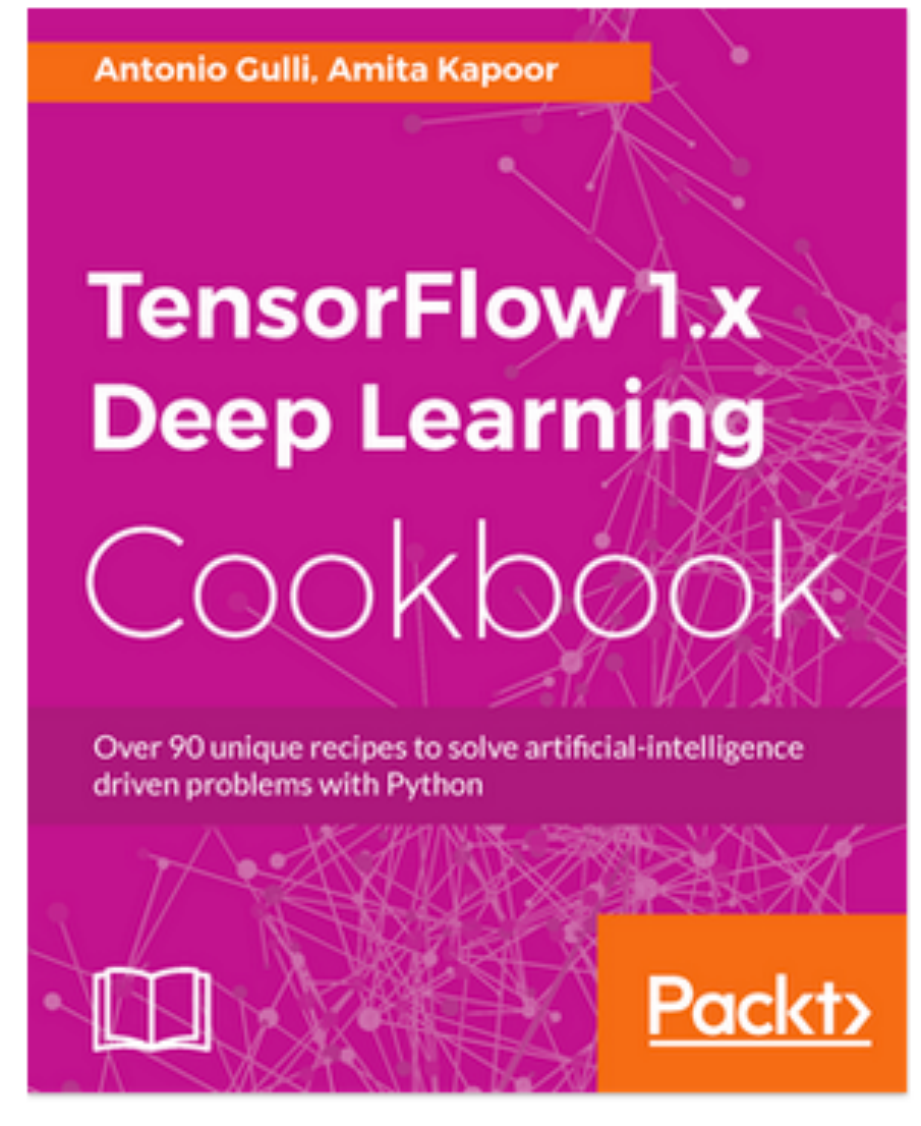
Disadvantages

- From a memory and capacity standpoint the CNN is not much bigger than a regular two layer network.
- At runtime the convolution operations are computationally expensive and take up about 67% of the time.
- CNN's are about 3X slower than their fully connected equivalents (size-wise).

Hands On

- CNN to classify handwritten MNIST
- CNN to classify CIFAR-10
- Transfer Style for image Repainting
- Transfer Learning
- Deep Dream Network
- Creating a ConvNet for Sentiment Analysis
- Generating music with Dilated ConvNets, WaveNet and NSynth
- Answering questions about images (Visual Q&A)

Hands-on


Antonio Gulli, Amita Kapoor
TensorFlow 1.x
Deep Learning
Cookbook
Over 90 unique recipes to solve artificial-intelligence driven problems with Python
Packt

References

- https://keras.io/applications/
- Inceptionv4: https://arxiv.org/abs/1602.07261
- Inceptionv3: https://arxiv.org/pdf/1512.00567.pdf
- Resnet: https://arxiv.org/pdf/1512.03385.pdf
- AlexNet: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
- LeNet: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
- VGGNet: https://arxiv.org/pdf/1409.1556.pdf