Introduction to Deep Learning

Data Science Research Lab Seminar

Poo Kuan Hoong

25th May 2016

Acknowledgement

Andrew Ng: Deep Learning, Self-Taught Learning and Unsupervised Feature Learning [Youtube]

Yann LeCun: Deep Learning Tutorial, ICML, Atlanta, 2013 [PDF]

Geoff Hinton, Yoshua Bengio & Yann LeCun: Deep Learning: NIPS2015 Tutorial [PDF]

Yoshua Bengio: Theano: A Python framework for fast computation of mathematical expressions. [URL]

Andrej Karpathy: Visualizing and Understanding Recurrent Networks, ICLR 2016, [PDF]

Outline

- A brief history of machine learning
- Understanding the human brain
- Neural Network: Concept, implementation and challenges
- Restricted Boltzmann Machine (RBM)
- Deep Belief Network (DBN): Concept and Application
- Convolutional Neural Network (CNN): Concept and Application
- Recurrent Neural Network (RNN): Concept and Application
- Deep Learning: Strengths, weaknesses and applications
- Deep Learning: Platforms, frameworks and libraries
- Demo

Introduction

- In the past 10 years, machine learning and Artificial Intelligence have shown tremendous progress
- The recent success can be attributed to:
  - Explosion of data
  - Cheap computing cost – CPUs and GPUs
  - Improvement of machine learning models
- Much of the current excitement concerns a subfield of it called “deep learning”.

A brief history of Machine learning

- Most of the machine learning methods are based on supervised learning

Input

Feature Representation

Learning Algorithm

A brief history of Machine learning

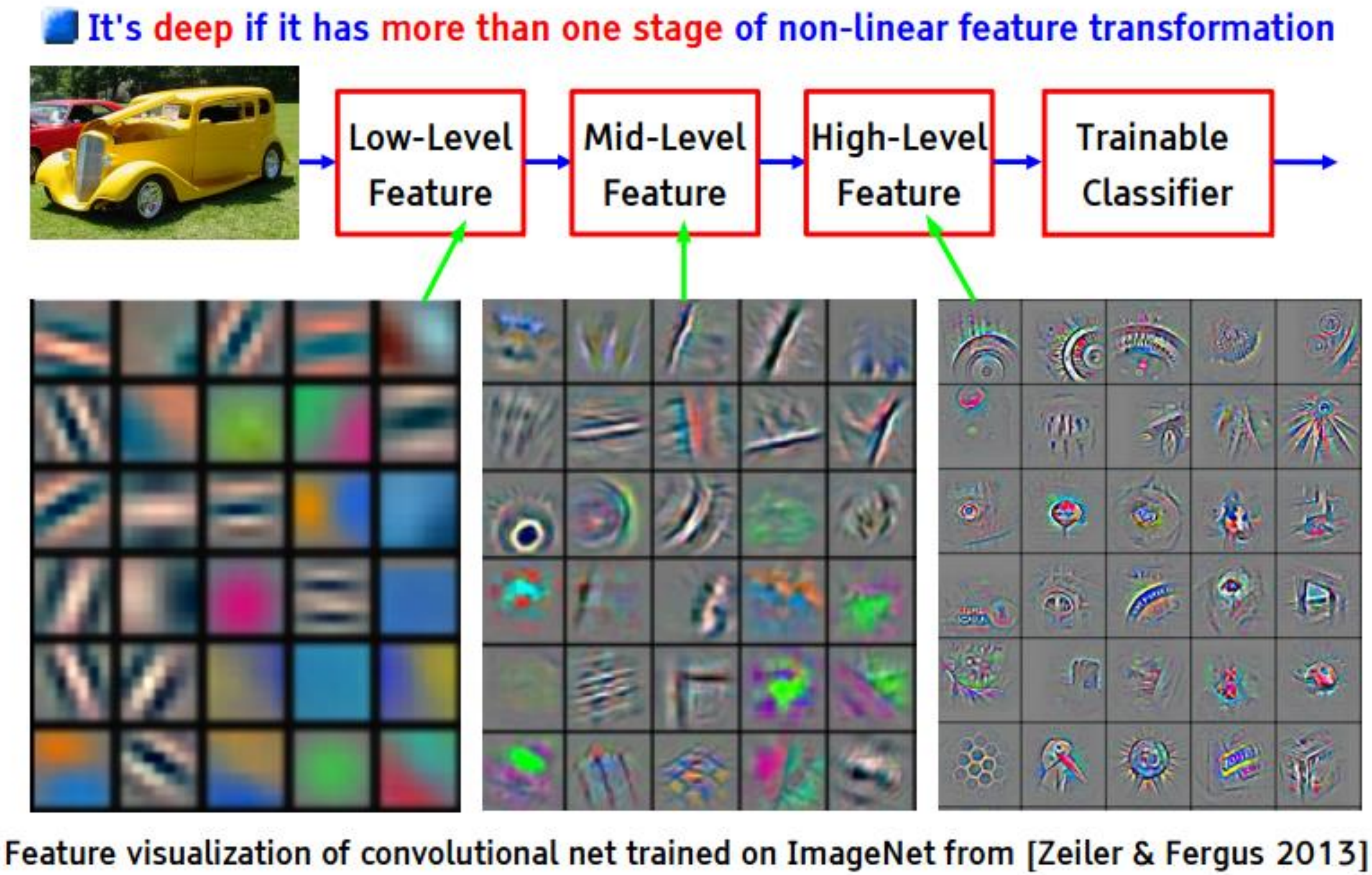
- **The traditional model of pattern recognition (since the late 50's)**
  - Fixed/engineered features (or fixed kernel) + trainable classifier

hand-crafted Feature Extractor → **"Simple" Trainable Classifier** →

$$\begin{bmatrix} 32 & 45 & 21 & .. \\ 12 & 10 & 45 & .. \\ 17 & 33 & 36 & .. \\ ... & ... & ... & \end{bmatrix}$$

$$\begin{bmatrix} 12 & 56 & 18 & .. \\ 92 & 76 & 22 & .. \\ 33 & 63 & 71 & .. \\ ... & ... & ... & \end{bmatrix}$$

Features Training



Trainable Feature Hierarchy

- Hierarchy of representations with increasing level of abstraction
- Each stage is a kind of trainable feature transform
- Image recognition
  - Pixel → edge → texton → motif → part → object
- Text
  - Character → word → word group → clause → sentence → story
- Speech
  - Sample → spectral band → sound → ... → phone → phoneme → word

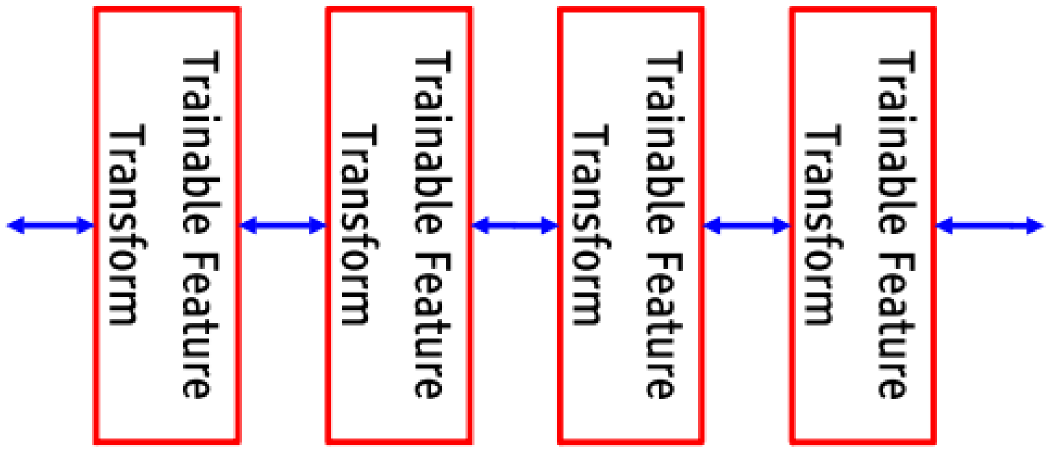
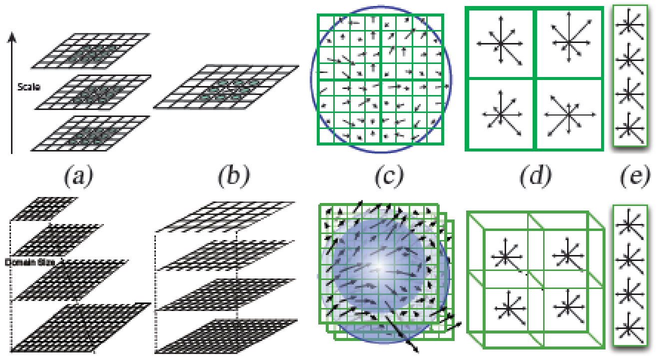
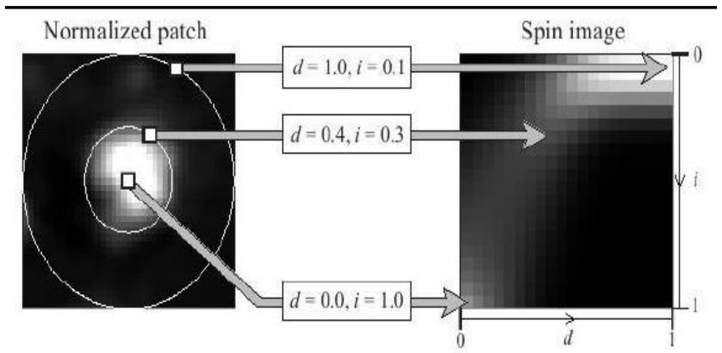


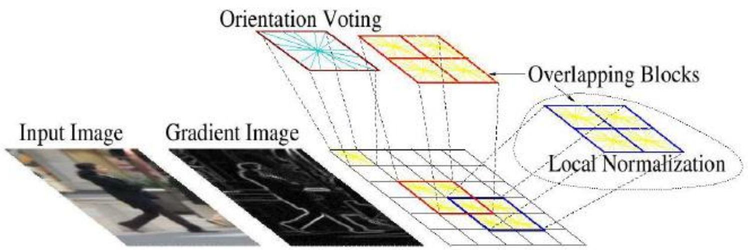
Image Recognition/Computer Vision Features



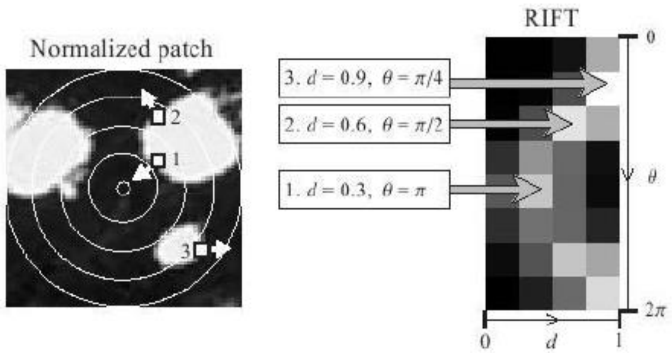
Scale-invariant feature transform (SIFT)



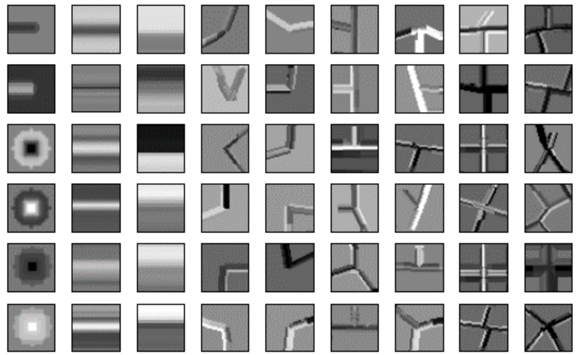
Spin Image



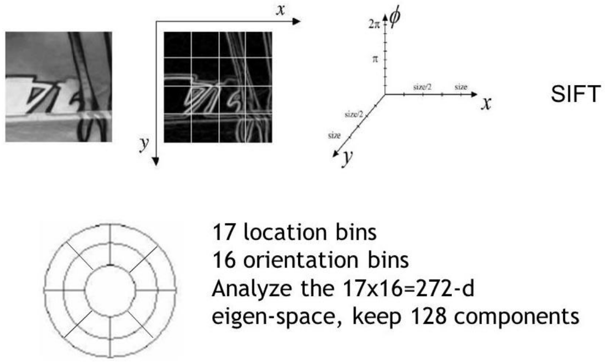
Histogram of Oriented Gradients (HoG)



Rotation Invariant feature transform (RIFT)

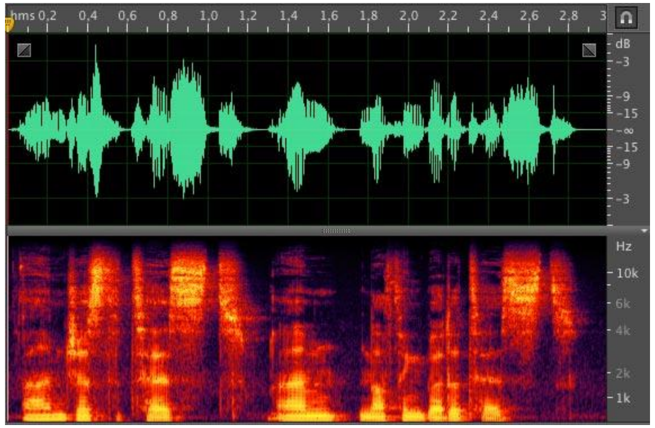


Textons

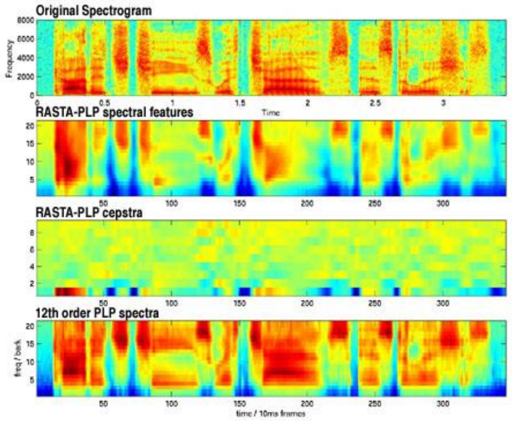


Gradient Location-Orientation Histogram (GLOH)

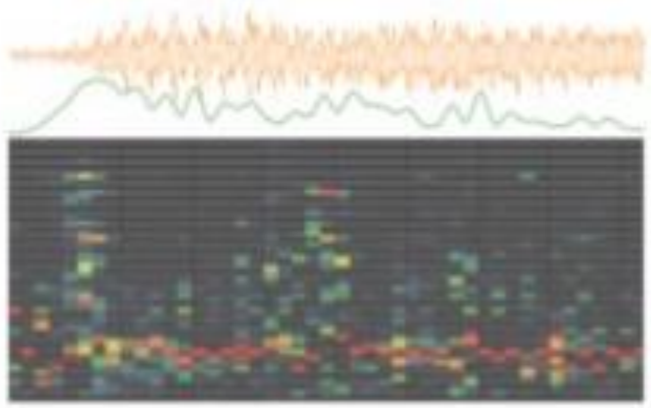
Speech/Audio Features



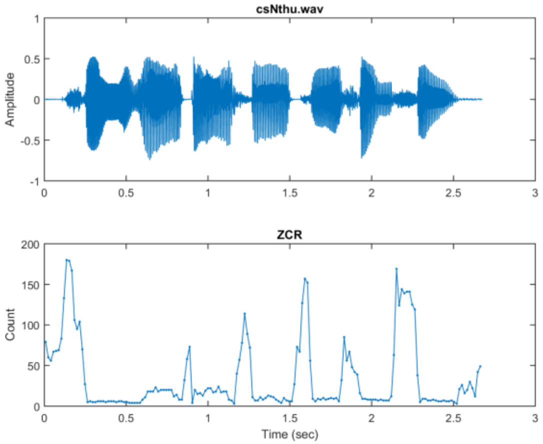
Spectrogram



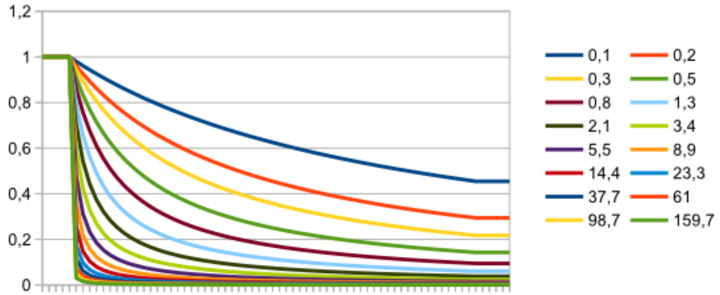
Mel-frequency cepstral coefficients (MFCC)



Flux

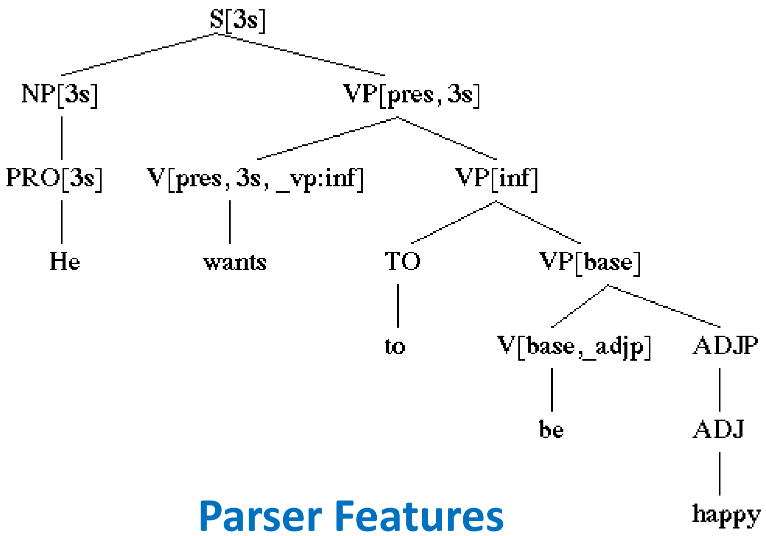


Zero Crossing Rate (ZCR)

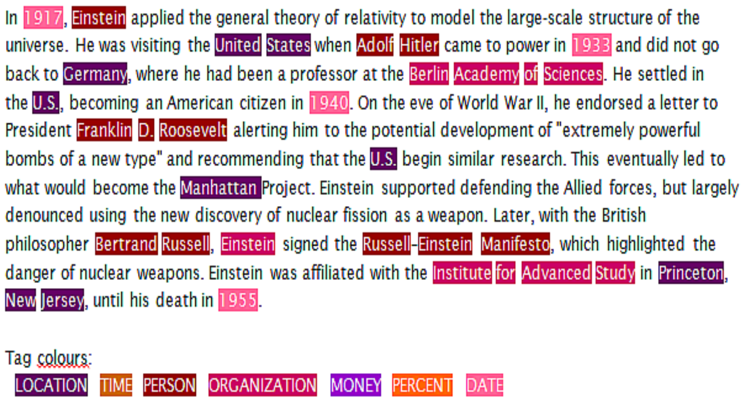


Roll-off

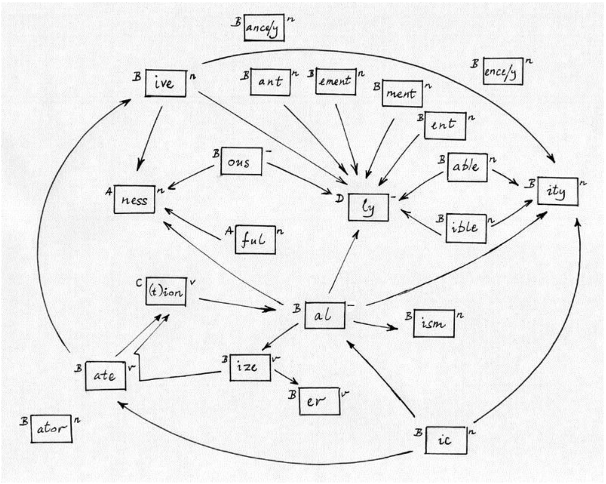
Text/NLP Features



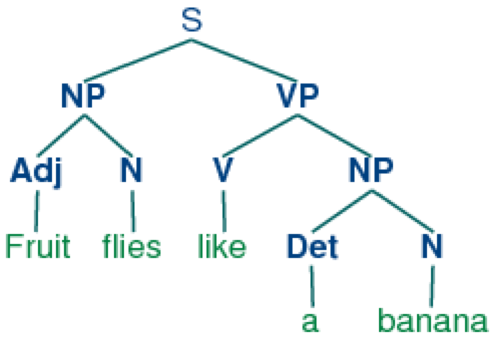
Parser Features



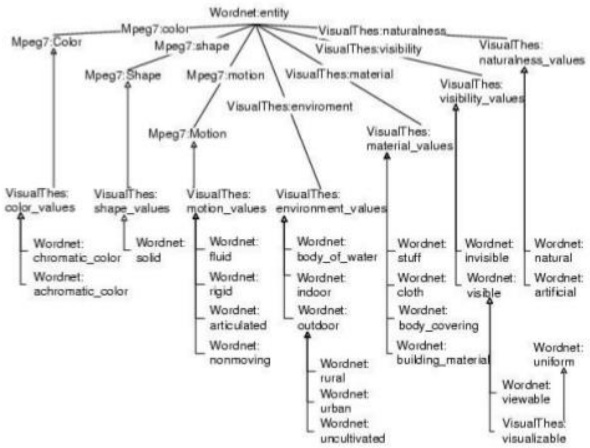
NER/SRL



Stemming



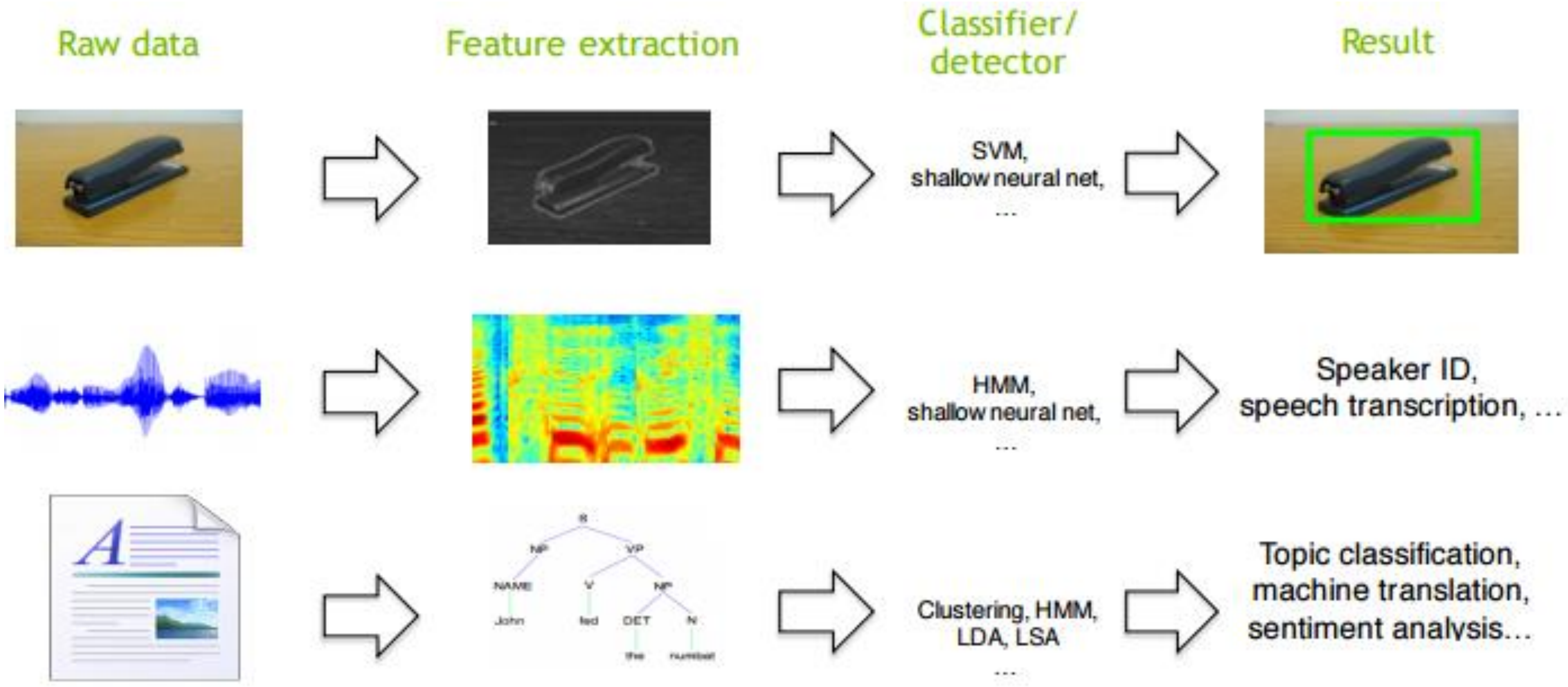
POS tagging



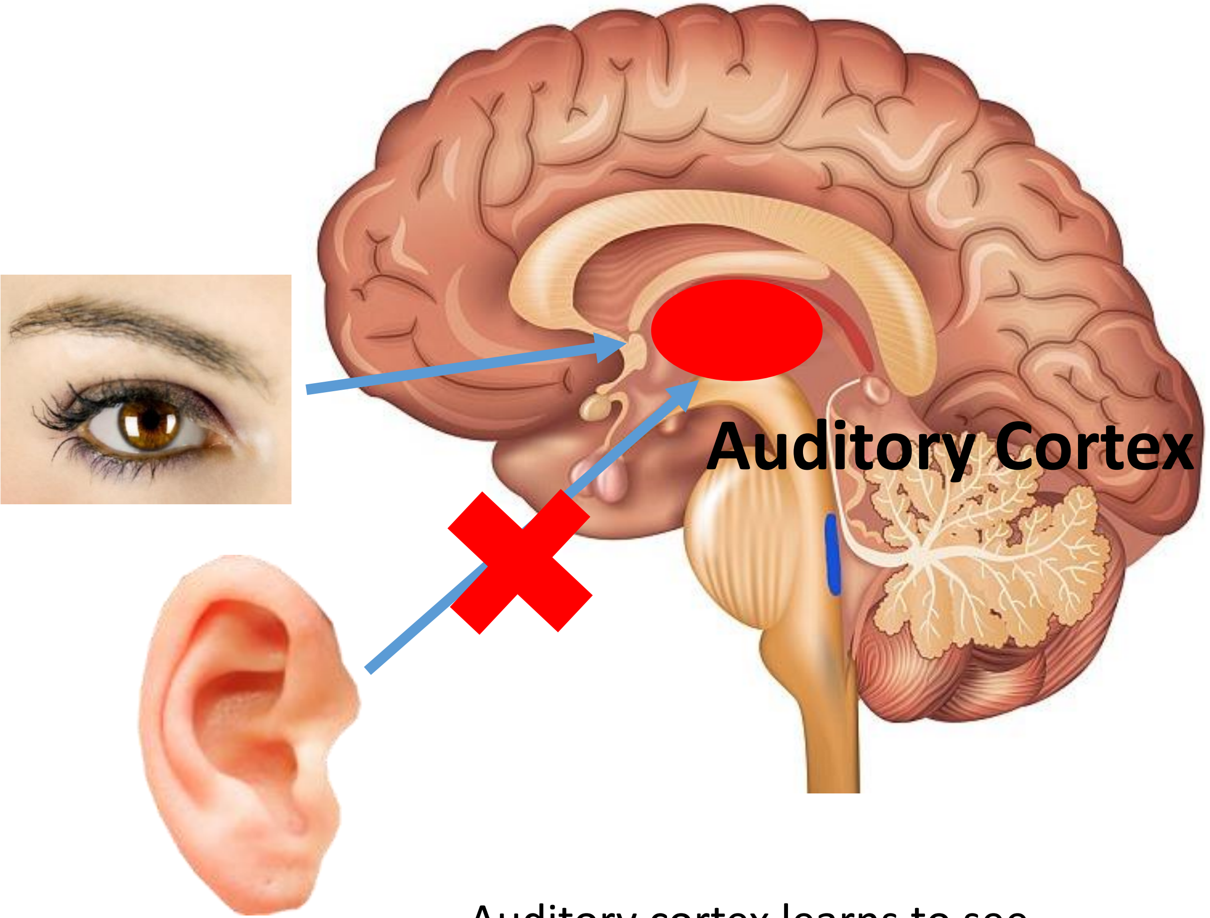
WordNet Features

Traditional machine perception

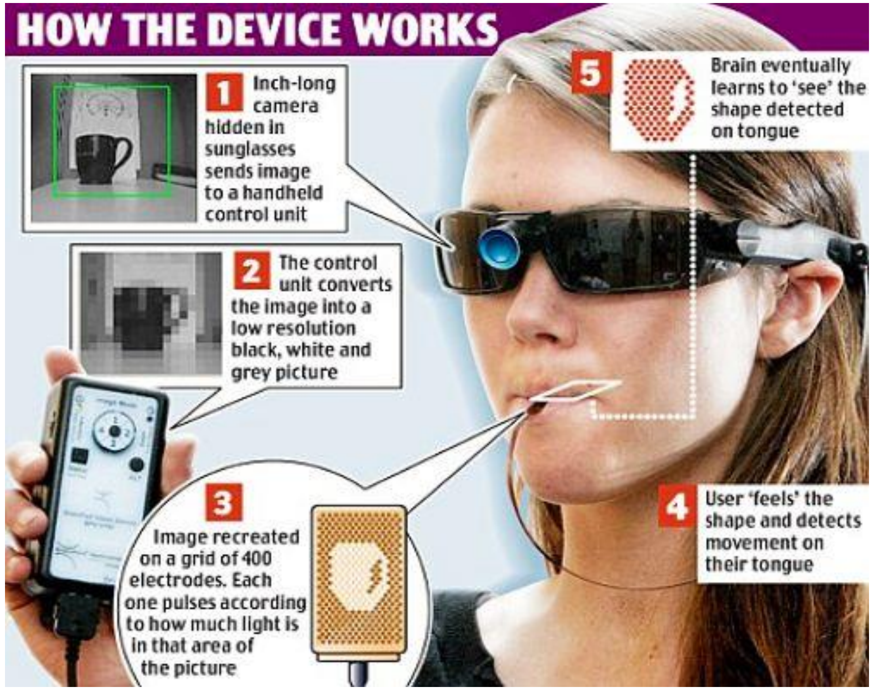
- Hand crafted feature extractors



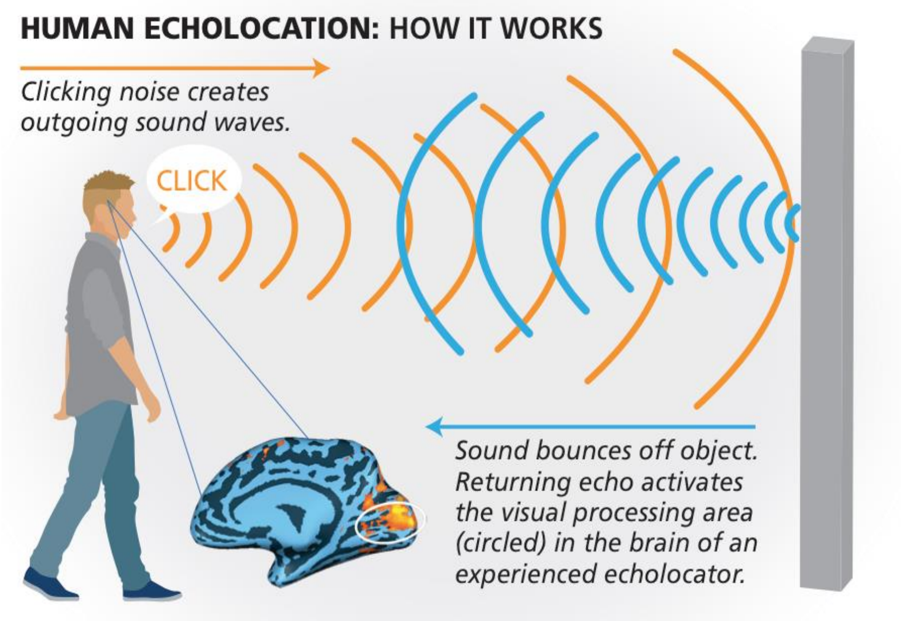
Human Brain



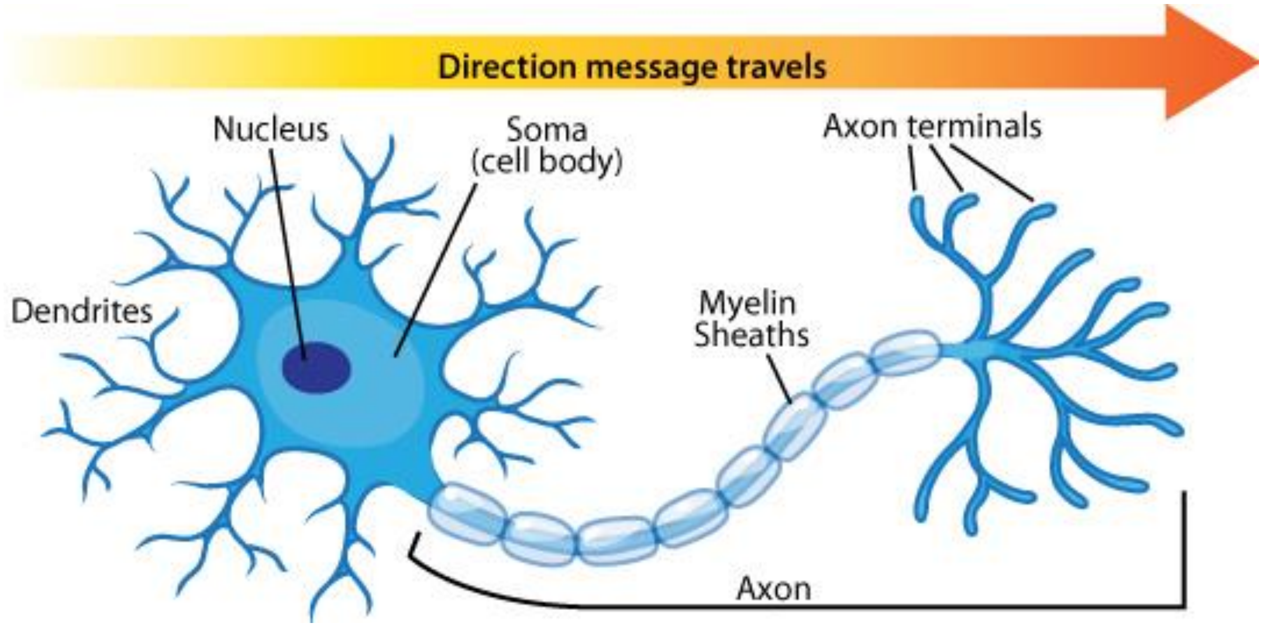
Auditory cortex learns to see. (Same rewiring process also works for touch/ somatosensory cortex.)



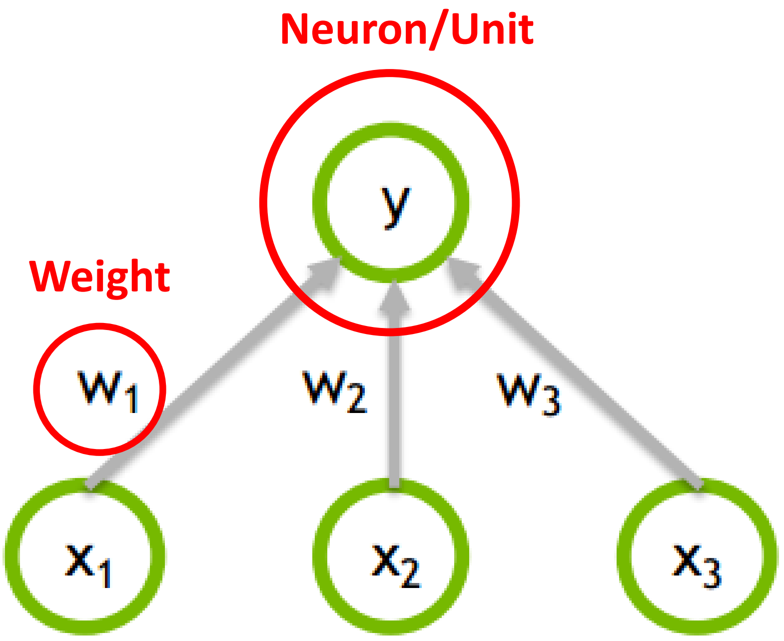
Seeing with tongue



Human Brain



Biological Neuron



$$y=F(w_1x_1+w_2x_2+w_3x_3)$$

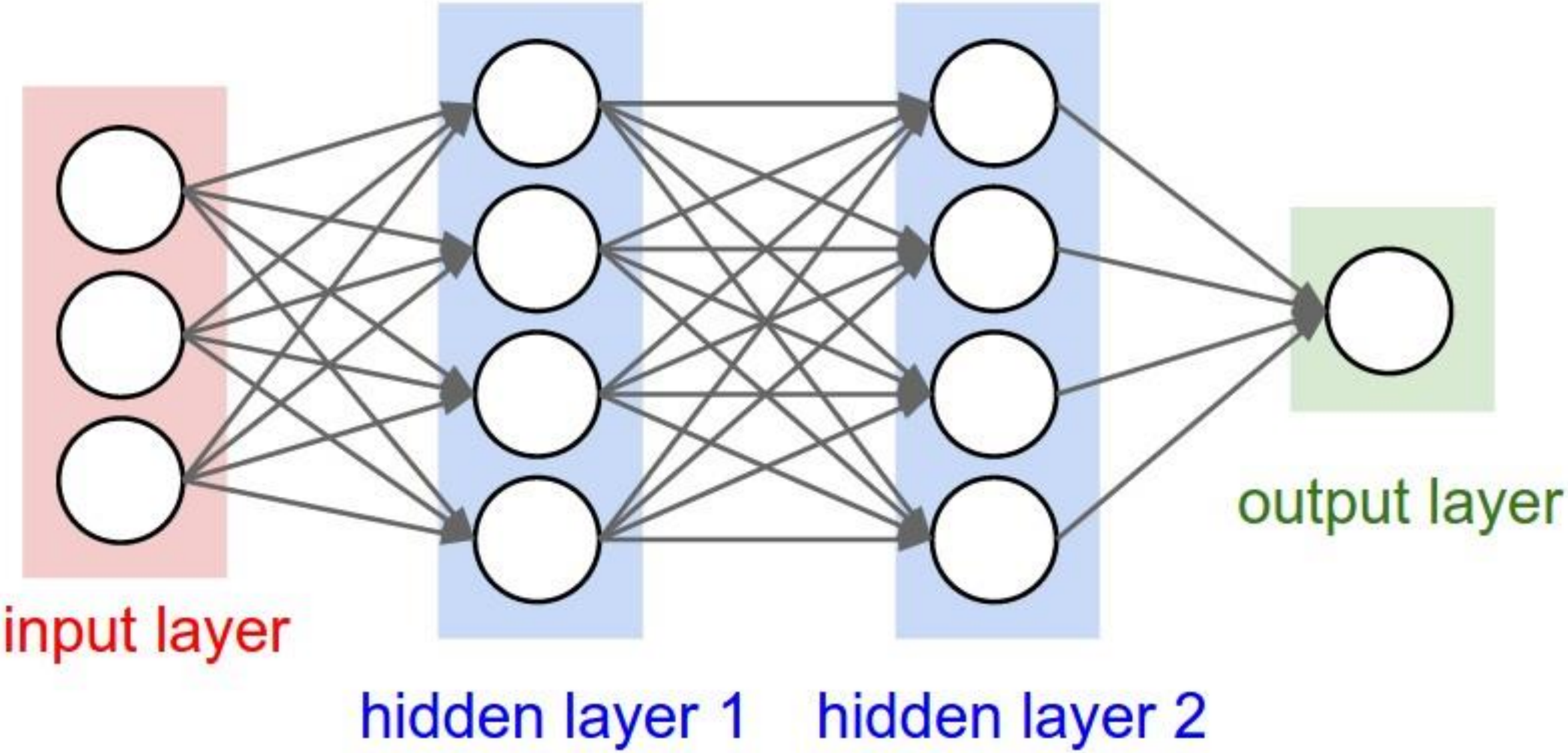
$$F(x)=\max(0,x)$$

Artificial Neuron

Neural Network

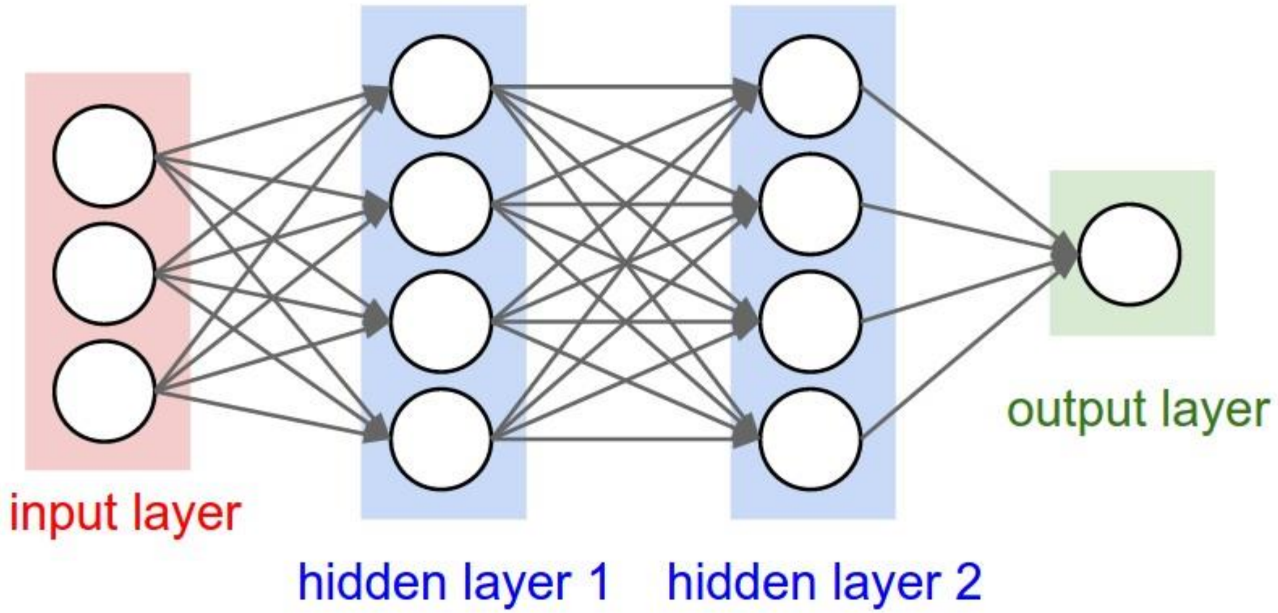
- Deep Learning is primarily about neural networks, where a network is an interconnected web of nodes and edges.
- Neural nets were designed to perform complex tasks, such as the task of placing objects into categories based on a few attributes.
- Neural nets are highly structured networks, and have three kinds of layers - an input, an output, and so called hidden layers, which refer to any layers between the input and the output layers.
- Each node (also called a neuron) in the hidden and output layers has a classifier.

Neural Network



Neural Network: Forward Propagation

- The input neurons first receive the data features of the object. After processing the data, they send their output to the first hidden layer.
- The hidden layer processes this output and sends the results to the next hidden layer.
- This continues until the data reaches the final output layer, where the output value determines the object's classification.
- This entire process is known as Forward Propagation, or Forward prop.



Neural Network: Backward Propagation

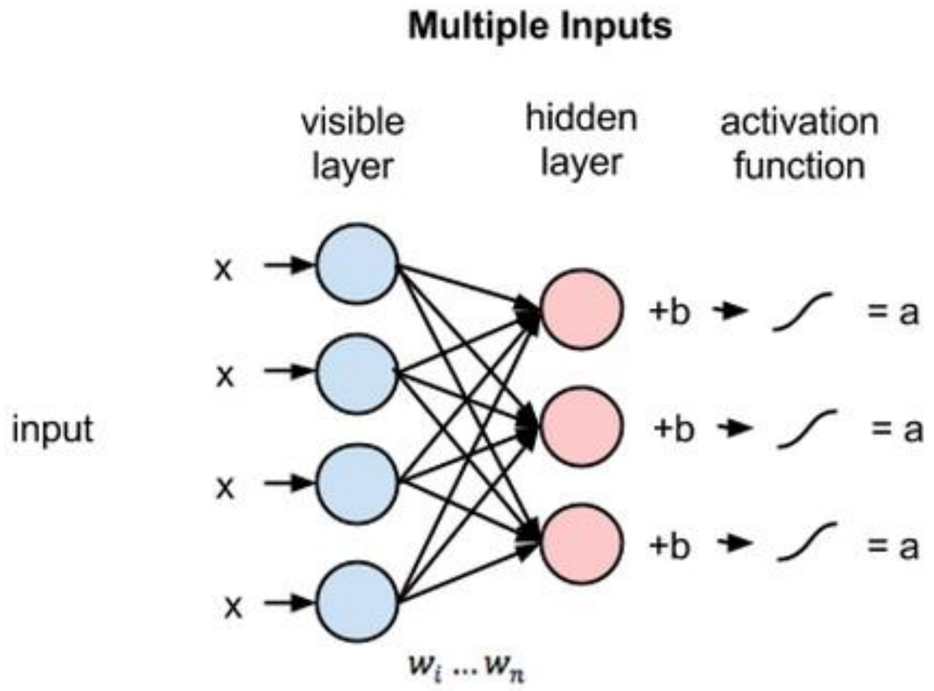
- To train a neural network over a large set of labelled data, you must continuously compute the difference between the network’s predicted output and the actual output.
- This difference is called the cost, and the process for training a net is known as backpropagation, or backprop
- During backprop, weights and biases are tweaked slightly until the lowest possible cost is achieved.
- An important aspect of this process is the gradient, which is a measure of how much the cost changes with respect to a change in a weight or bias value.

The 1990s view of what was wrong with back-propagation

- It required a lot of labelled training data
  - almost all data is unlabeled
- The learning time did not scale well
  - It was very slow in networks with multiple hidden layers.
- It got stuck at local optima
  - These were often surprisingly good but there was no good theory

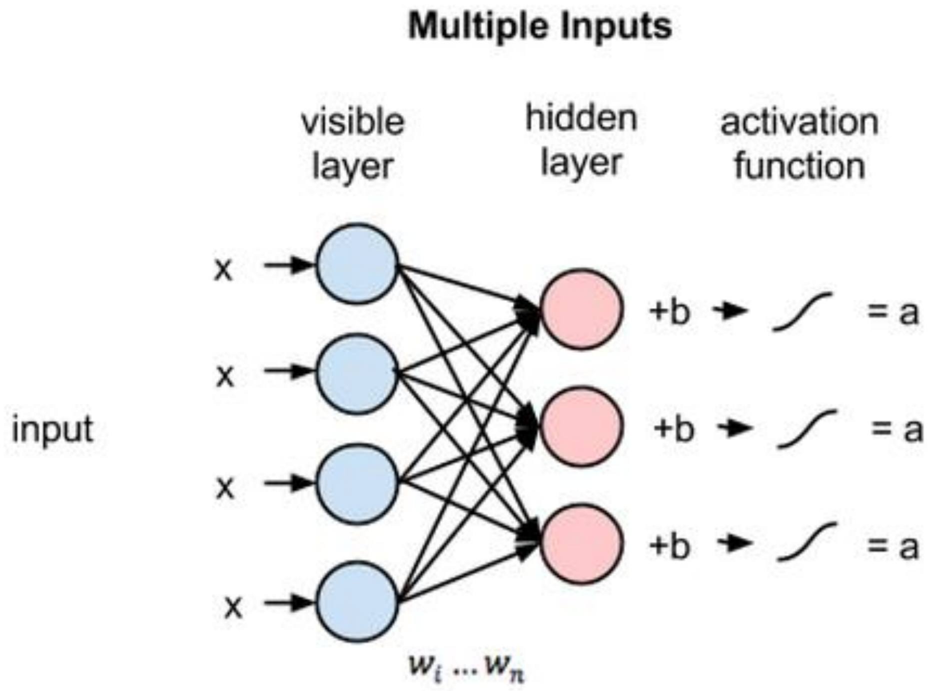
Restricted Boltzmann Machine (RBM)

- Restricted Boltzmann Machine, or RBM by Geoff Hinton of the University of Toronto.
- RBM is a shallow neural net with just 2 layers – the visible layer and the hidden layer. In this net, each node connects to every node in the adjacent layer.
- The “restriction” refers to the fact that no two nodes from the same layer share a connection.



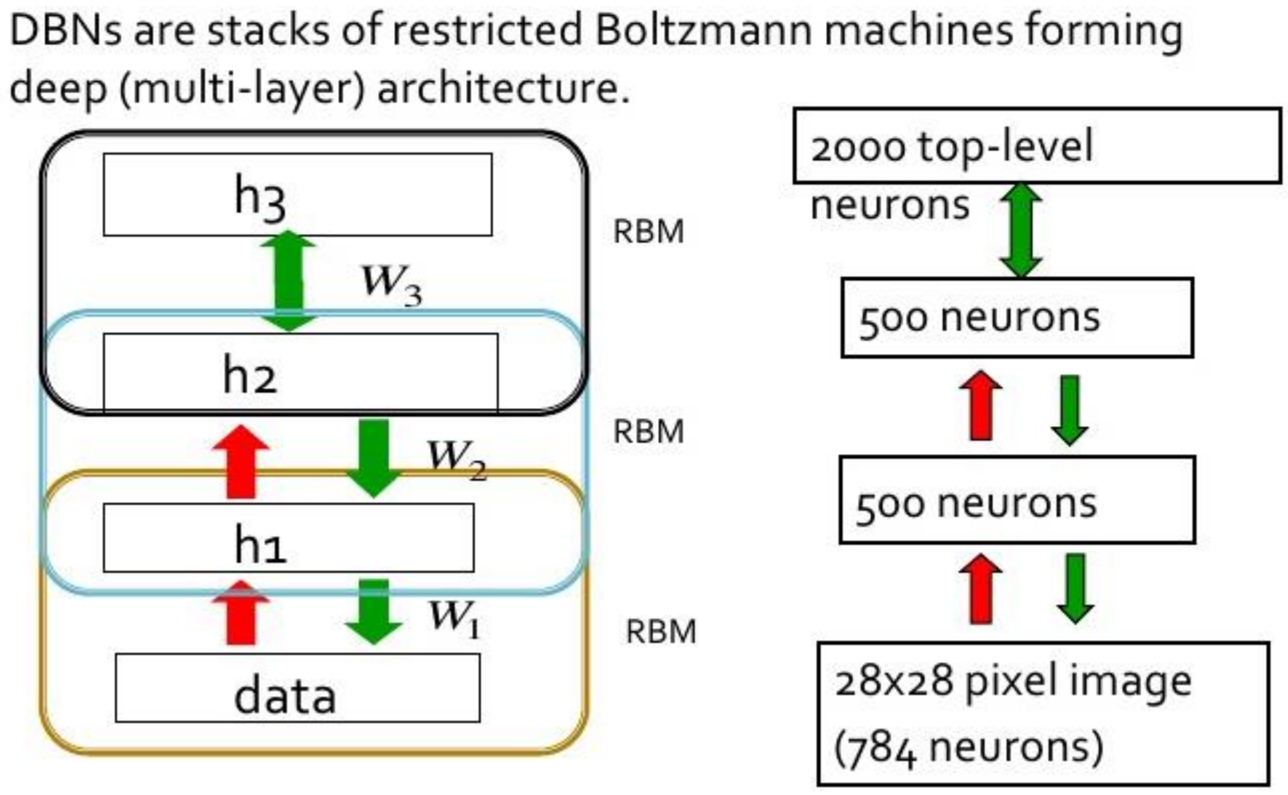
Restricted Boltzmann Machine (RBM)

- The goal of an RBM is to recreate the inputs as accurately as possible. During a forward pass, the inputs are modified by weights and biases and are used to activate the hidden layer.
- In the next pass, the activations from the hidden layer are modified by weights and biases and sent back to the input layer for activation.
- At the input layer, the modified activations are viewed as an input reconstruction and compared to the original input.
- Because RBMs try to reconstruct the input, the data does not have to be labelled.
- RBM can be used as features extractor.
- Belong to a family of features extractor called auto encoders.



Deep Belief Network (DBN)

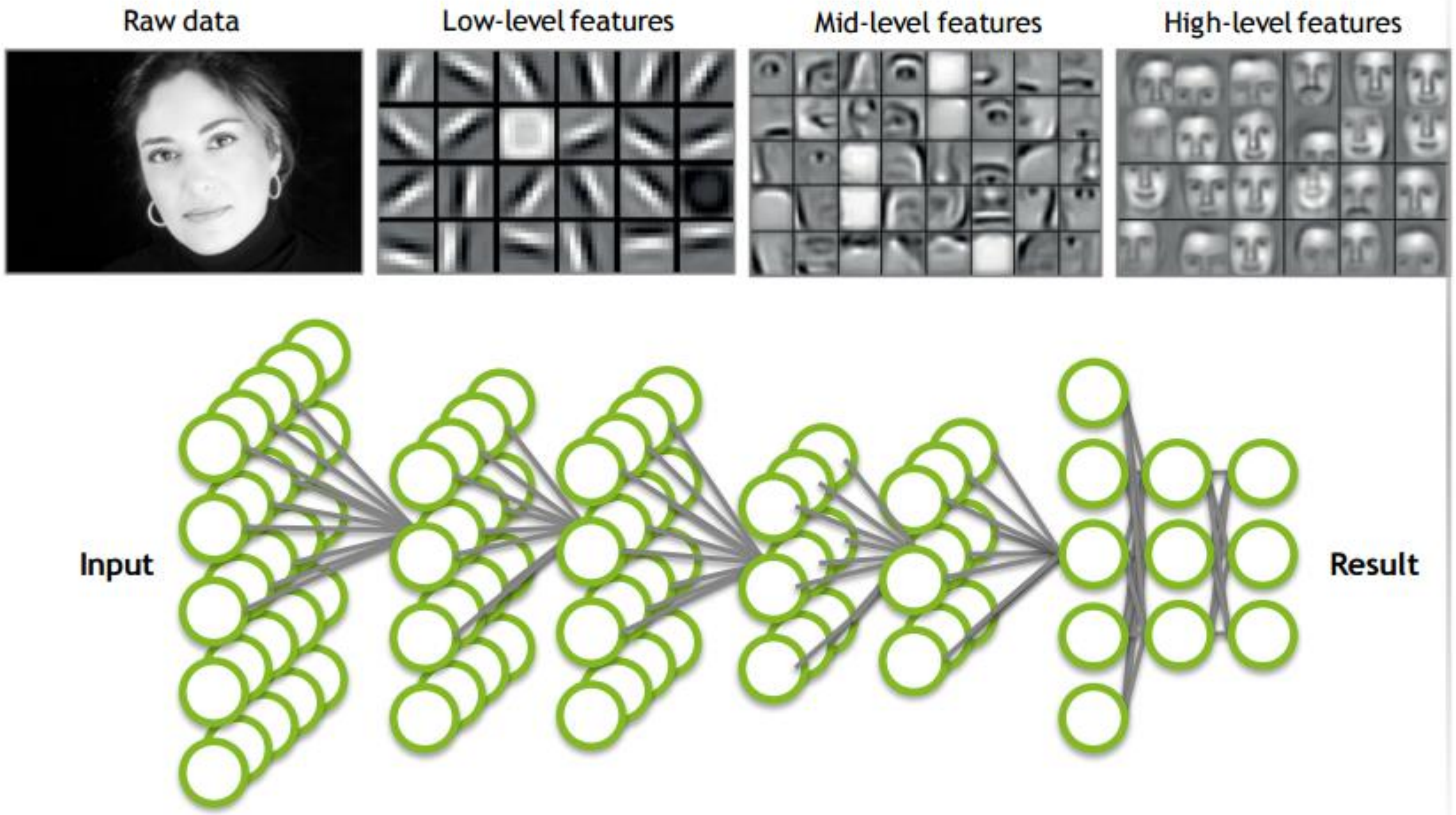
- The Deep Belief Network, or DBN, was also conceived by Geoff Hinton.
- Used by Google for their work on the image recognition problem.
- DBN is trained two layers at a time, and these two layers are treated like an RBM.
- Throughout the net, the hidden layer of an RBM acts as the input layer of the adjacent one. So the first RBM is trained, and its outputs are then used as inputs to the next RBM. This procedure is repeated until the output layer is reached.



Deep Belief Network (DBN)

- DBN is capable of recognizing the inherent patterns in the data. In other words, it’s a sophisticated, multilayer feature extractor.
- The unique aspect of this type of net is that each layer ends up learning the full input structure.
- Layers generally learn progressively complex patterns – for facial recognition, early layers could detect edges and later layers would combine them to form facial features.
- DBN learns the hidden patterns globally, like a camera slowly bringing an image into focus.
- DBN still requires a set of labels to apply to the resulting patterns. As a final step, the DBN is fine-tuned with supervised learning and a small set of labeled examples.

Deep Network (Deep Net)



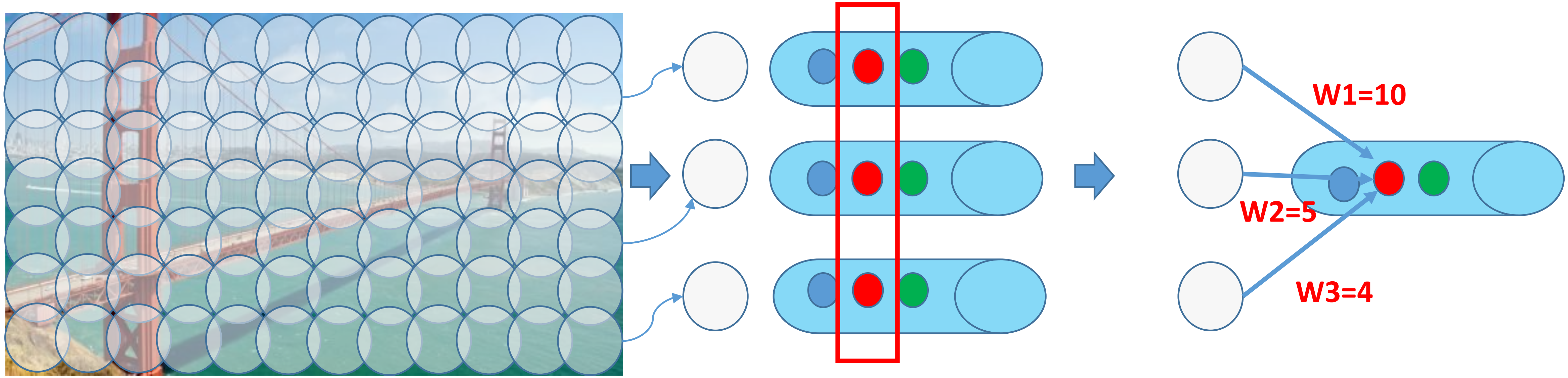
Application components:

Task objective
e.g. Identify face
Training data
10-100M images
Network architecture
~10 layers
1B parameters
Learning algorithm
~30 Exaflops
~30 GPU days

Convolutional Neural Network (CNN)

- CNN inspired by the Visual Cortex.
- CNNs are deep nets that are used for image, object, and even speech recognition.
- Pioneered by Yann Lecun (NYU)
- Deep supervised neural networks are generally too difficult to train.
- CNNs have multiple types of layers, the first of which is the convolutional layer.

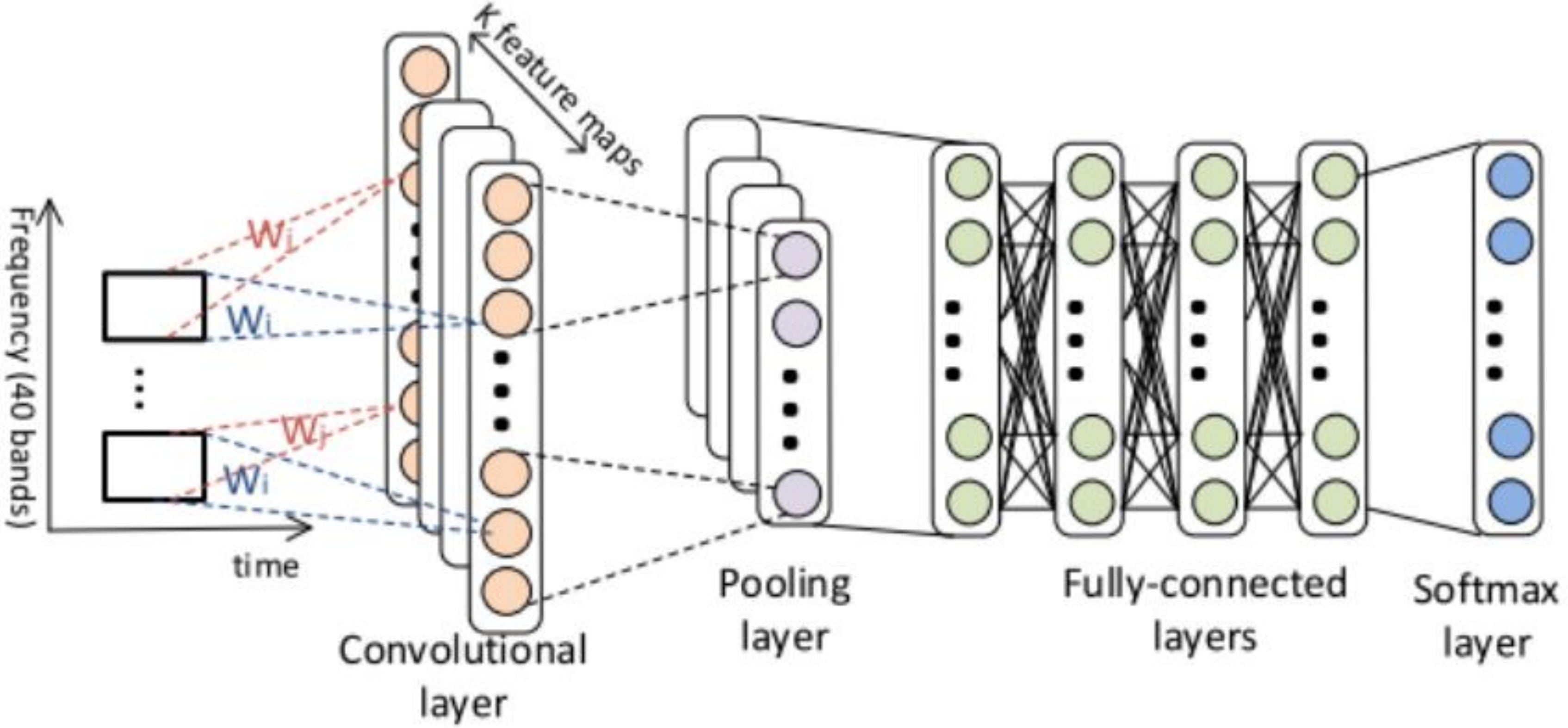
Convolutional Neural Network (CNN)



Filter 2/ Neural 2

- A series of filters forms layer one, called the convolutional layer. The weights and biases in this layer determine the effectiveness of the filtering process.
- Each flashlight represents a single neuron. Typically, neurons in a layer activate or fire. On the other hand, in the convolutional layer, neurons search for patterns through convolution. Neurons from different filters search for different patterns, and thus they will process the input differently.

Convolutional Neural Network (CNN)

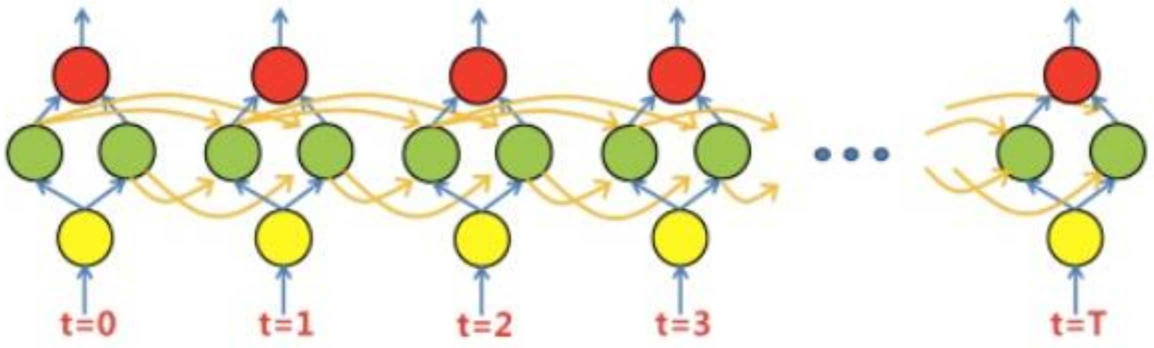


CNN: Application

- Classify a scene in an image
  - Image Classifier Demo (NYU): http://horatio.cs.nyu.edu/
- Describe or understanding an image
  - Toronto Deep Learning Demo: http://deeplearning.cs.toronto.edu/i2t
  - MIT Scene Recognition Demo: http://places.csail.mit.edu/demo.html
- Handwriting recognition
  - Handwritten digits recognition: http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html
- Video classification
  - Large-scale Video Classification with Convolutional Neural Networks http://cs.stanford.edu/people/karpathy/deepvideo/

Recurrent Neural Network (RNN)

- The Recurrent Neural Net (RNN) is the brainchild of Juergen Schmidhuber and Sepp Hochreiter.
- RNNs have a feedback loop where the net's output is fed back into the net along with the next input.
- RNNs receive an input and produce an output. Unlike other nets, the inputs and outputs can come in a sequence.
- Variant of RNN is Long Term Short Memory (LSTM)



RNN: Application

- RNN is suitable for time series data, where an output can be the next value in a sequence, or the next several values

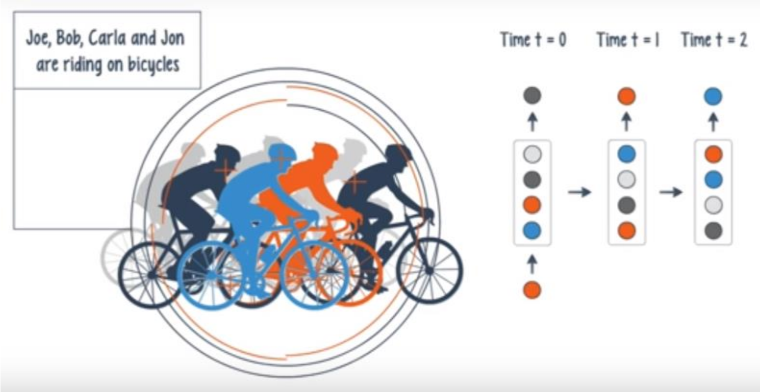
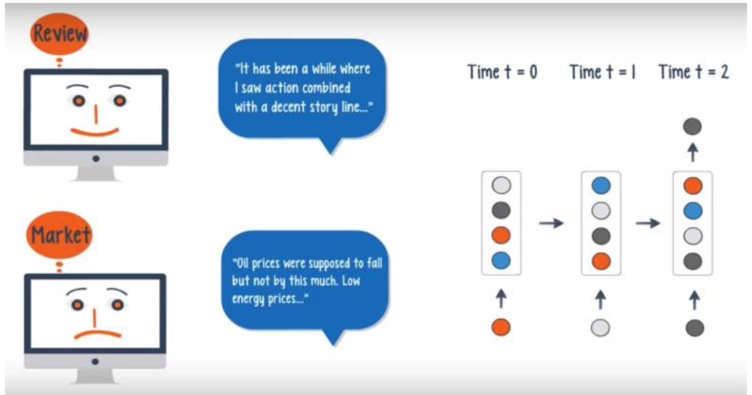
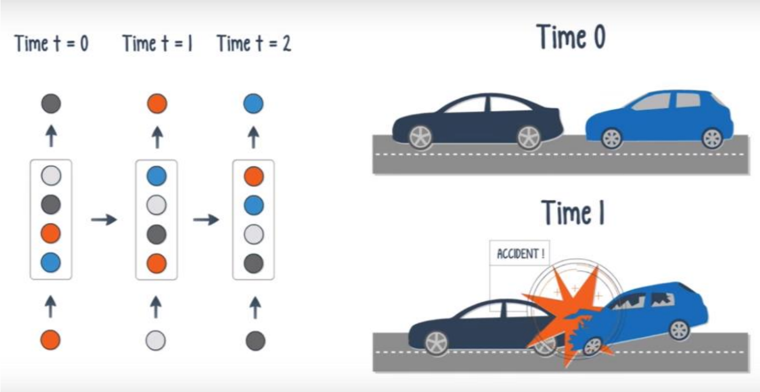


Image captioning



Document Classification



Classify Image frame by frame

Deep Learning: Benefits

- Robust
  - No need to design the features ahead of time – features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- Generalizable
  - The same neural net approach can be used for many different applications and data types
- Scalable
  - Performance improves with more data, method is massively parallelizable

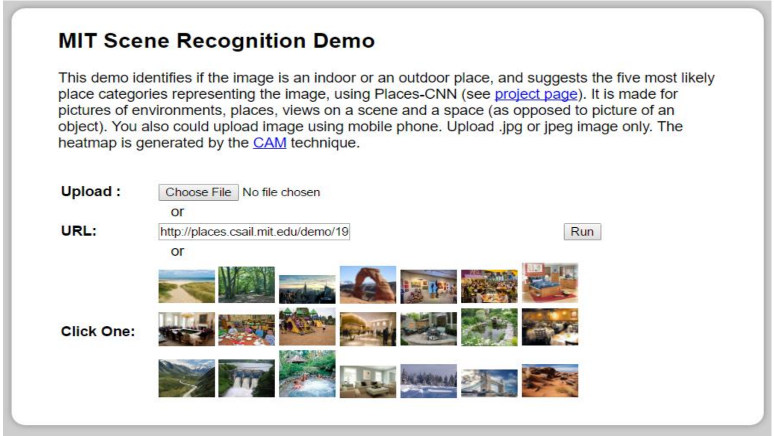
Deep Learning: Weaknesses

- Deep Learning requires a large dataset, hence long training period.
- In term of cost, Machine Learning methods like SVMs and other tree ensembles are very easily deployed even by relative machine learning novices and can usually get you reasonably good results.
- Deep learning methods tend to learn everything. It’s better to encode prior knowledge about structure of images (or audio or text).
- The learned features are often difficult to understand. Many vision features are also not really human-understandable (e.g, concatenations/combinations of different features).
- Requires a good understanding of how to model multiple modalities with traditional tools.

Deep Learning: Applications

Deep Learning: Application

https://deepmind.com/alpha-go



http://places.csail.mit.edu/demo.html

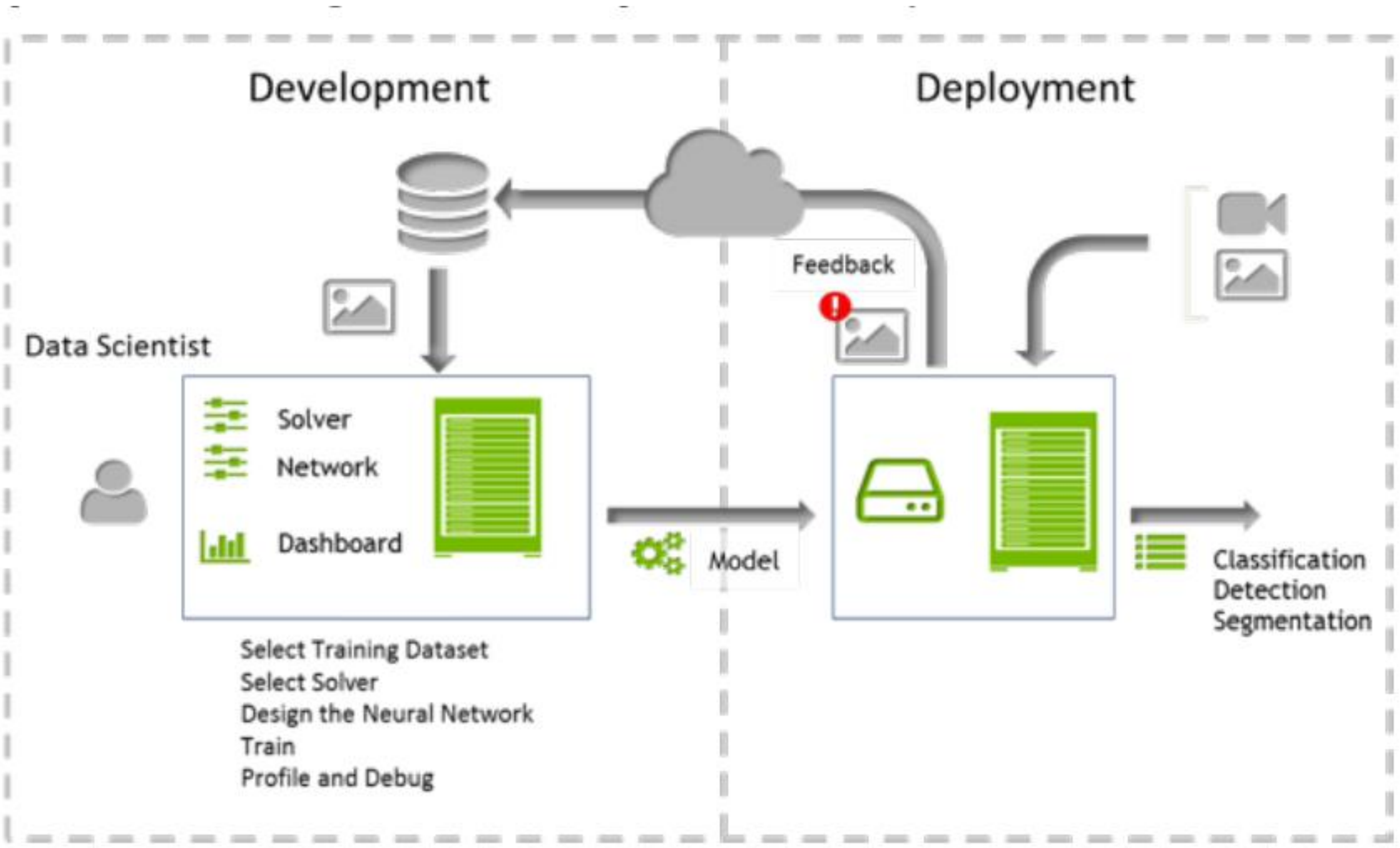
Dulight--Eyes for visually impaired

Robotic grasping

Deep Q-learning playing Atari

Pedestrian detection using DL

Deep Learning Development Cycle



Deep Learning: Platform & Frameworks & Libraries

Platform

- Ersatz Labs - cloud-based deep learning platform [http://www.ersatz1.com/]
- H20 – deep learning framework that comes with R and Python interfaces [http://www.h2o.ai/verticals/algos/deep-learning/]

Framework

- Caffe - deep learning framework made with expression, speed, and modularity in mind. Developed by the Berkeley Vision and Learning Center (BVLC) [http://caffe.berkeleyvision.org/]
- Torch - scientific computing framework with wide support for machine learning algorithms that puts GPUs first. Based on Lua programming language [http://torch.ch/]

Library

- Tensorflow - open source software library for numerical computation using data flow graphs from Google [https://www.tensorflow.org/]
- Theano - a python library developed by Yoshua Bengio's team [http://deeplearning.net/software/theano/]

Learned Models

- Trained Models can be shared with others
- Save the training time
- For example: AlexNet, GoogLeNet, ParseNet, etc
- URLs:
  - https://github.com/BVLC/caffe/wiki/Model-Zoo
  - http://deeplearning4j.org/model-zoo

Demo

MNIST Handwritten digits Database

- The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples.
- It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.
- The database is available from here.