STEVENS INSTITUTE OF TECHNOLOGY

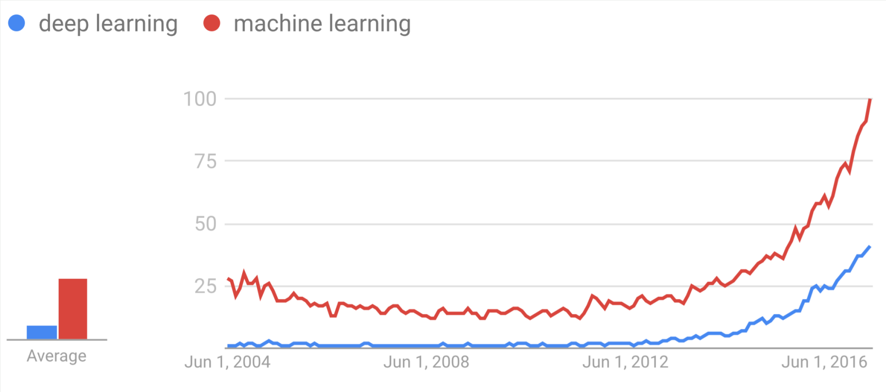
from Machine Learning to Deep Learning:

How Artificial Intelligence is Changing the World!

Rensheng Wang,
https://sit.instructure.com/courses/26266
January 25, 2018

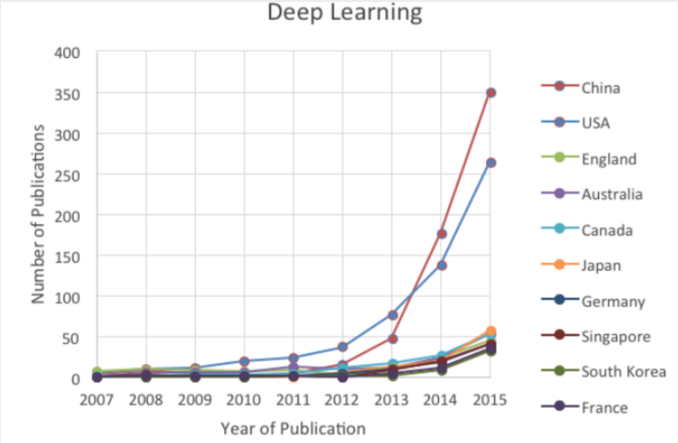
Machine Learning vs. Deep Learning

- Interest in google trends



Machine Learning vs. Deep Learning

- Academic publications about deep learning



What is Machine Learning?

- Machine Learning is a core transformative way by which we are rethinking everything we are doing
- Machine Learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed
- Machine Learning provides various techniques that can learn from and make predictions on data
- Learning approaches:
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

What is Deep Learning?

- Deep learning is also known as deep structured learning, hierarchical learning or deep machine learning
- Deep learning s a branch of machine learning based on a set of algorithms
- It attempts to model high-level abstractions in data by using a deep graph with multiple processing layers
- Deep learning is composed of multiple linear and non-linear transformations.

What is NOT Deep?

- Linear models are not deep (by definition)
- Neural nets with 1 hidden layer are not deep (only 1 layer, – no feature hierarchy)
- SVM and Kernel methods are not deep (only 2 layers: kernel + linear)
- Classification trees are not deep (operate on original input space, no new features generated)

Why Deep Learning is So Hype?

- Old wine in new bottle! Neural networks are nothing new.
- We have been trying to map neurons in the brain since the 1890s and using those principles in computer science since the late 60s.
- Whats changed is the scale at which we can do these things.
- Increased computing power has allowed us to map and process much larger neural networks than ever before. We also have a lot more data that we can use to train these networks.

What Deep Learning Can Do?

- Computer Writing Captions:
  - With the use of neural networks, computers can not only recognize pictures of cats, but they can actually describe what those cats are doing
  - You can even ask questions of a computer about the image you showed it. So it could tell you that the cat is white or the floor is with blue carpet.
- Seeing for the Blind
  - If computers not only recognize images, but understand and interpret them the same way our brains can, then we can effectively mimic sight for people who dont have any.

What Deep Learning Can Do?

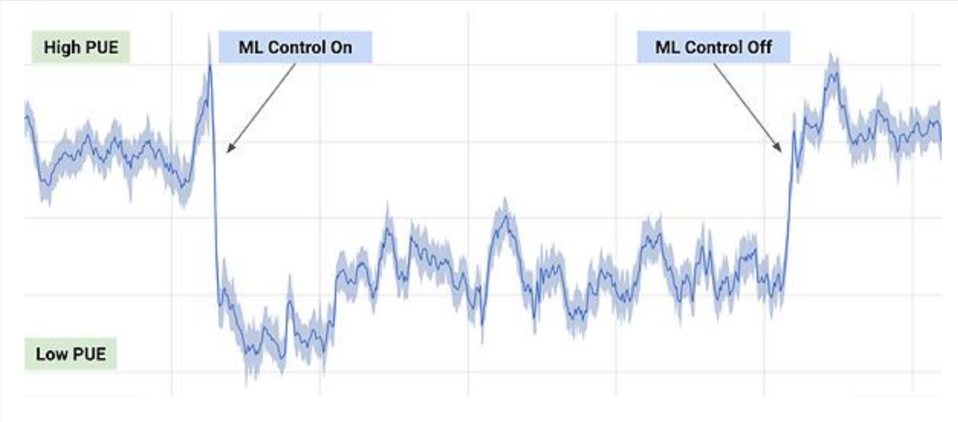
- Improved speech recognition:
  - Speech recognition from 89% accuracy to 99% accuracy.
  - High accuracy improves the chance of communications between human and intelligent devices
- Predicting user behavior
  - Recommender systems for users & machines
- Healthcare
  - In July 2016, a collaboration between DeepMind and Moorfields Eye Hospital for the analysis of anonymised eye scans, searching for early signs of diseases leading to blindness.
  - In August 2016, University College London Hospital started to develop an algorithm that can automatically differentiate between healthy and cancerous tissues in head and neck areas.

Google DeepMind

- DeepMind founded in 2010 and acquired by Google in 2014 with $500million.
- Google use DeepMind AI to cut data center energy bills
- Alpha Go!
- DeepMind AI fake the most realistic human voices

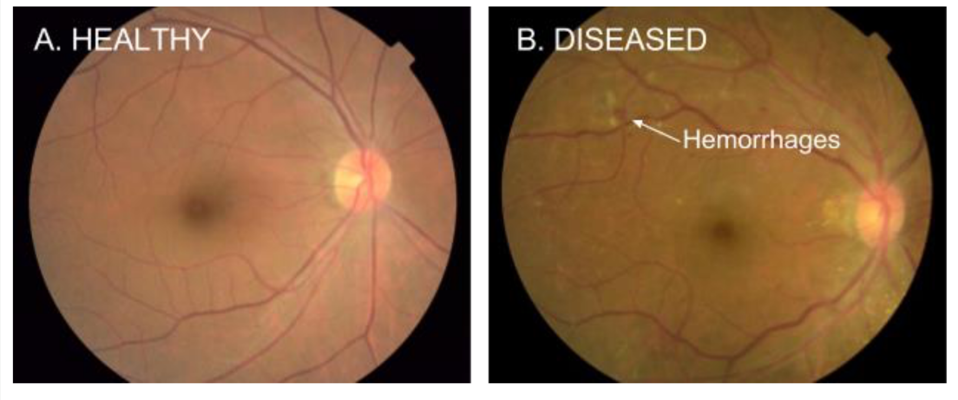
Google DeepMind

- Save Energy



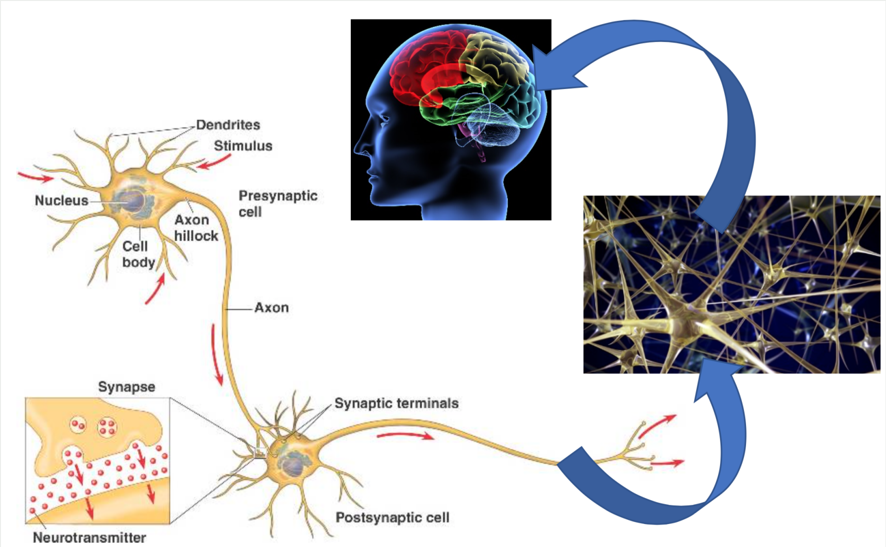
Google DeepMind

- Detection of diabetic eye disease



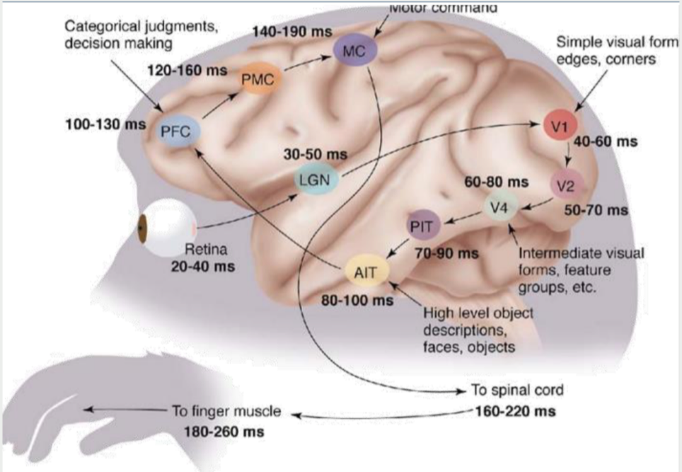
Human Brain

- Deep Learning ⇔ Machine Learning ⇔ Neural Network ⇔ Human Brain



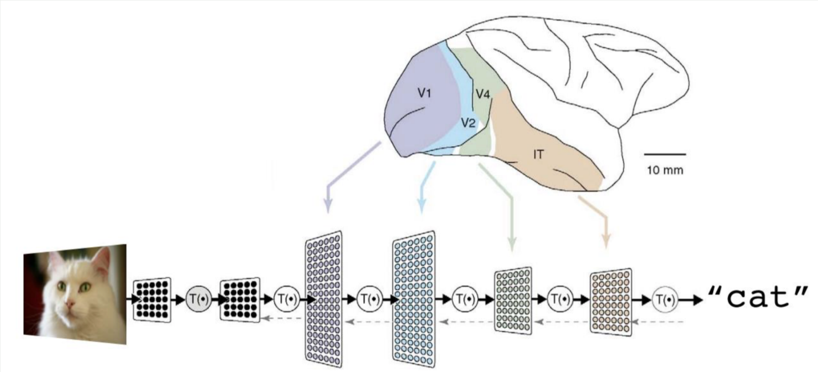
Inspired by the Brain

- Our brain has lots of neurons connected together and the strength of the connections between neurons represents long term knowledge.



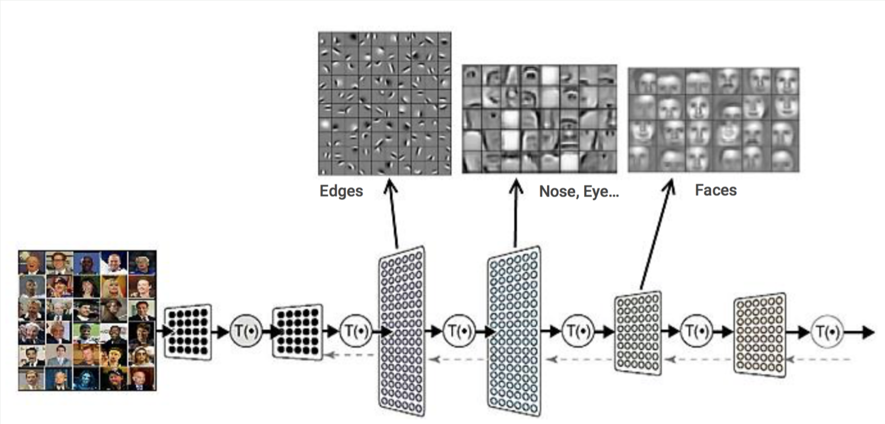
Deep Learning Architecture

- Deep neural network consists of a hierarchy of layers, whereby each layer transforms the input data into more abstract representations (eg., from edge -¿ nose -¿ face)
- The output layer combines those features to make predictions



Deep Learning Architecture

- How to learn?



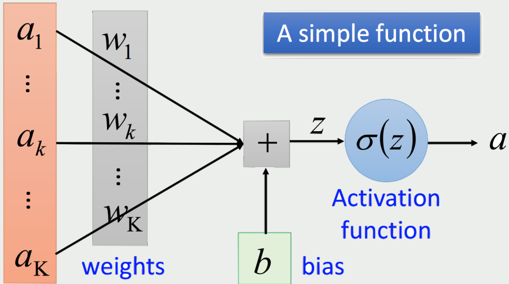
Neuron

- The neuron can be described as a information processing unit

$$z = w_1 a_1 + w_2 a_2 + \cdots + w_k a_k + b$$

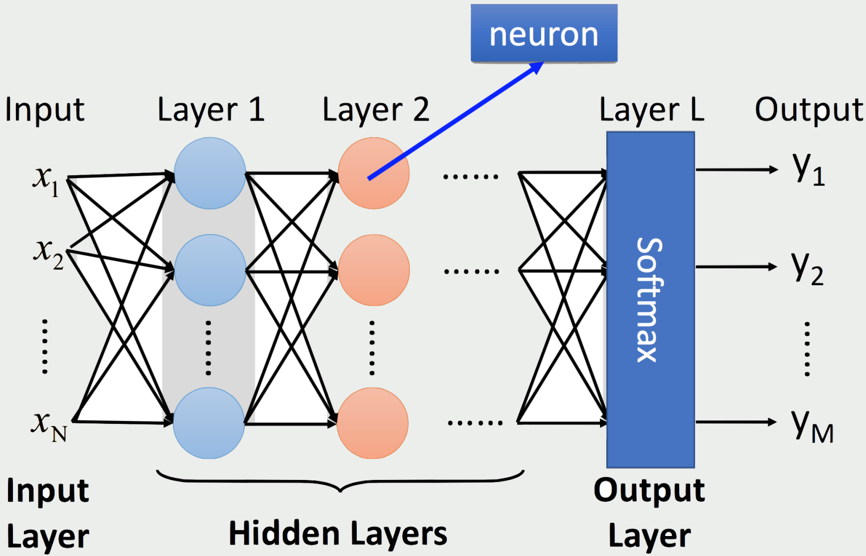
- The activation function of a node defines the output of that node given an input or set of inputs. As a digital network of activation function, it can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behavior of the linear perceptron in neural networks.

- The common activation function is sigmoid function $\sigma(z) = \dfrac{1}{1+e^{-z}}$



Neural Network

- There are multiple neurons and they can convey the information through multiple layers.



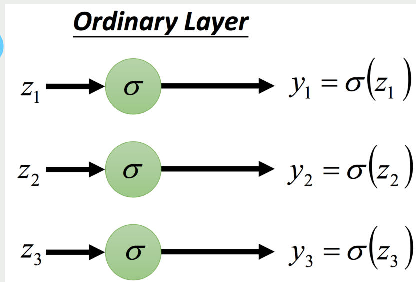
Output Layer

- The output after different layers in neural network can be any values. They are usually difficult to interpret.
- We use softmax layer as output layer. → with SOFTMAX:

$0 \le |y_n| \le 1$ ⇒ o/p lies b/w 0 & 1

(eg: $y_1 = 0.7$
$y_2 = 0.2$
$y_1 = 0.1$)

Normalizing step



Softmax Layer

$$y_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

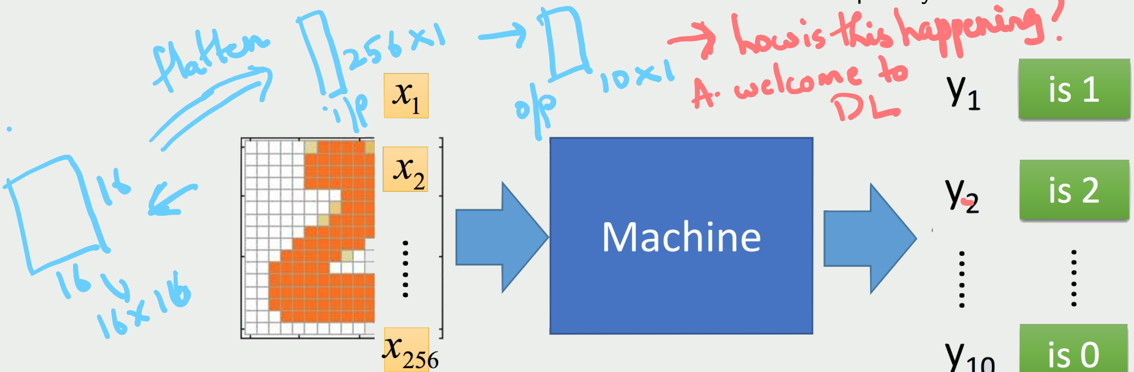
$$y_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$y_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

↳ these no.s can be interpreted as probabilities

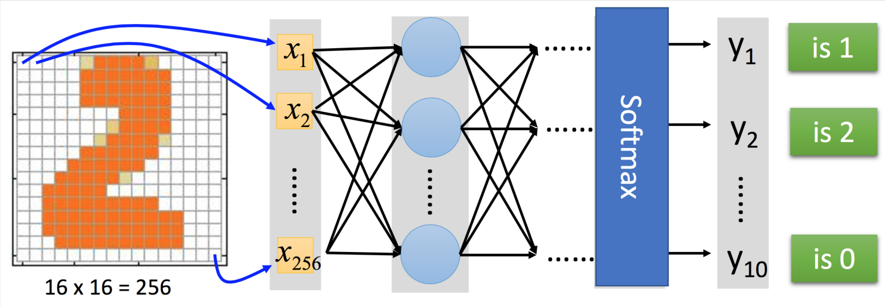
Example: HandWriting Recogintion

- For a 16×16 image with digit 2, it can be reformed as a 256× 1 input vector $\mathbf{x} = [x_1,\ x_2, \cdots, x_{256}]^T$
- Output is the probability vector $\mathbf{y} = [y_1, y_2, \cdots, y_{10}]^T$ for every single digit from 1, 2,$\cdots$, 9, 0.
- From input $\mathbf{x}$ to $\mathbf{y}$, we need to find a machine (i.e., a function) to link them.
- This machine can be formed as neural network with multiple layers.

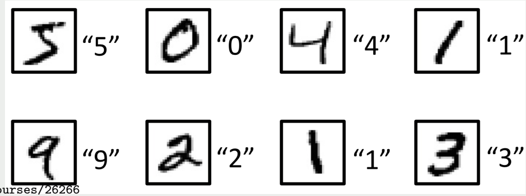


Learn Target

- The learning target is to make, y_2, the probability of image to be "2" maximum.



- To make the machine robust, we need more images and their labels for training



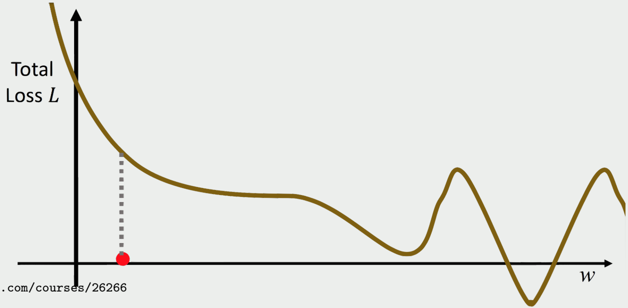
How to Learn?

with matrix multiplicn

- The learning target is to find the network **parameters** to **minimize** the difference between the output layers and the targets.
- The differences are called function loss. The loss can be "square errors" or "cross entropy".
- It is an optimization problem.
- There are too many unknown parameters $\{w_1, w_2, \cdots, b_1, b_2, \cdots\}$. Imagine 8 layers with 1000 neuron each layer.
- The feasible solution is **gradient descent**.

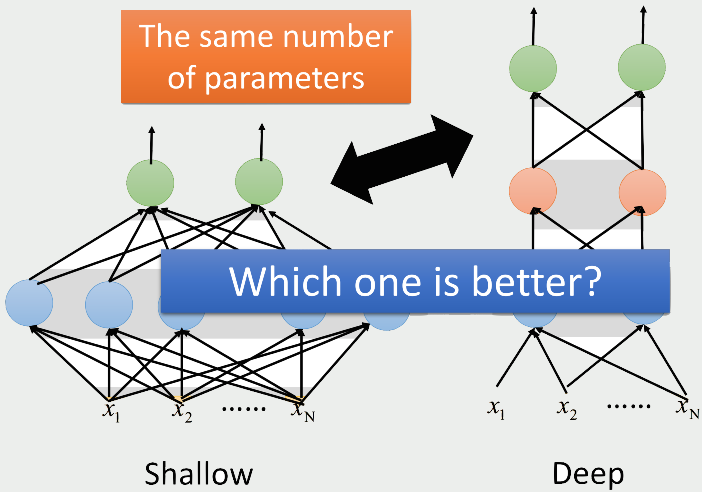
Gradient Descent

- The optimization problem is to find the network parameters $\{w_1, \cdots, b_1, \cdots\}$ to minimize the total loss L. diffnce b/w o/p & desired results
- Randomly generate the initial values for $\{w_1, \cdots, b_1, \cdots\}$.
- Then compute the gradients $\partial L / \partial w$.
  - If $\partial L / \partial w < 0$, increase w
  - If $\partial L / \partial w > 0$, decrease w
- update $w \leftarrow w - \eta \frac{\partial L}{\partial w}$, where η is learning rate, until total loss converges.



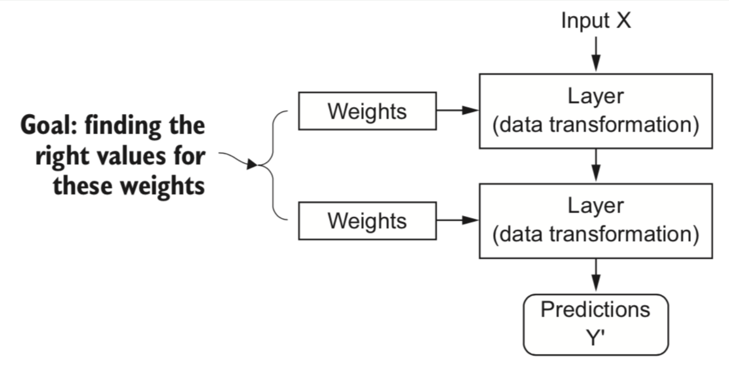
Deeper is Better?

- More layers means more accurate results?
- Prefer deep & thin or shallow & fat?



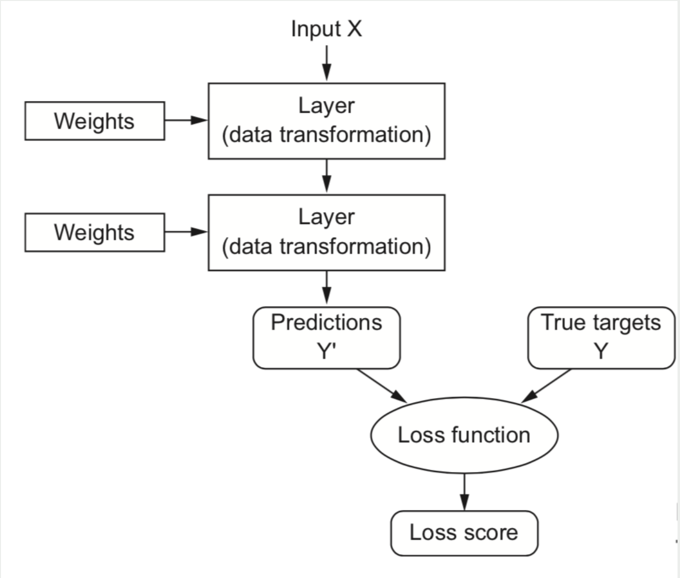
Procedures in Deep Learning (I)

- A neural network is parameterized by its weights



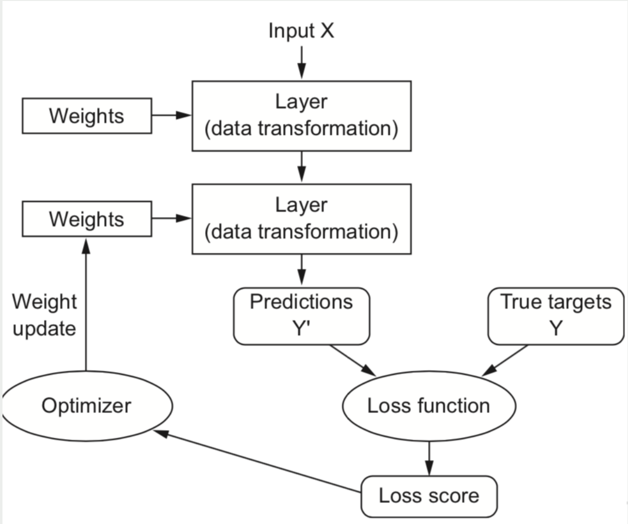
Procedures in Deep Learning (II)

- A neural network is parameterized by its weights



Procedures in Deep Learning (III)

- A neural network is parameterized by its weights



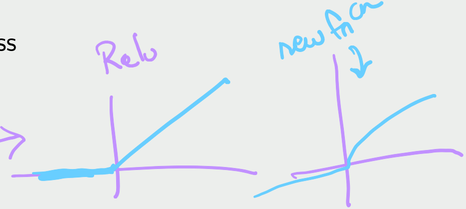
How to Improve Learning Expereience?

- A loss function measures the quality of the networks output
  - MSE vs. Cross Entropy

used when o/p is closer to '1'

$$\text{MSE} = \sum_i (y_i - \hat{y}_i)^2 \qquad \text{Cross-Entropy} = -\sum_i \hat{y}_i \ln y_i$$

  - When softmax output layer is used, choose cross-entropy loss
- Mini-batch → breaking down the task into
- New activation function



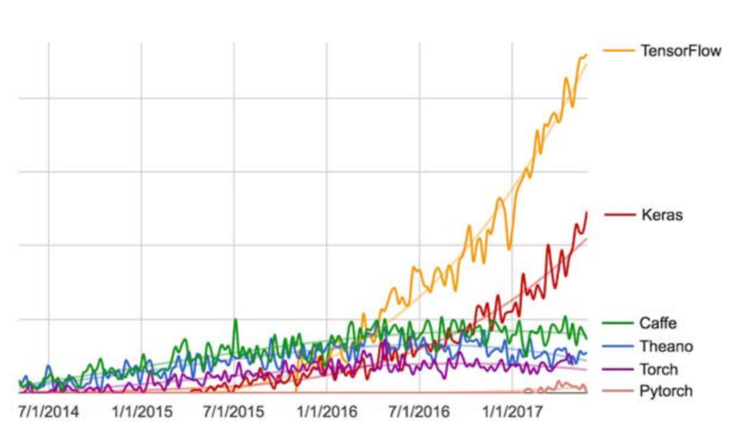
- Adaptive learning rate
- Momentum

What Deep Learning Has Achieved So Far

- Near-human-level image classification
- Near-human-level speech recognition
- Near-human-level handwriting transcription
- Improved machine translation
- Improved text-to-speech conversion
- Digital assistants such as Google Now and Amazon Alexa
- Near-human-level autonomous driving
- Improved ad targeting, as used by Google, Baidu, and Bing
- Improved search results on the web
- Ability to answer natural-language questions
- Superhuman Go playing

Introduction to Keras

- Keras is a deep-learning framework for Python that provides a convenient way to define and train almost any kind of deep-learning model. Keras was initially developed for researchers, with the aim of enabling fast experimentation.
- Google web search interest for different deep-learning frameworks over time



Introduction to Keras

- Keras has the following key features:
  - It allows the same code to run seamlessly on CPU or GPU.
  - It has a user-friendly API that makes it easy to quickly prototype deep-learning models.
  - It has built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
  - It supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, and so on. This means Keras is appropriate for building essentially any deep-learning model, from a generative adversarial network to a neural Turing machine.

Summary of Introducation Lecture

- Learning means finding a combination of model parameters that minimizes a loss function for a given set of training data samples and their corresponding targets.
- Learning happens by drawing random batches of data samples and their targets, and computing the gradient of the network parameters with respect to the loss on the batch.
- The entire learning process is made possible by the fact that neural networks are chains of differentiable tensor operations, and thus its possible to apply the chain rule of derivation to find the gradient function mapping the current parameters and current batch of data to a gradient value.
- Two key concepts: loss and optimizers.
- The loss is the quantity youll attempt to minimize during training, so it should represent a measure of success for the task youre trying to solve.
- The optimizer specifies the exact way in which the gradient of the loss will be used to update parameters.