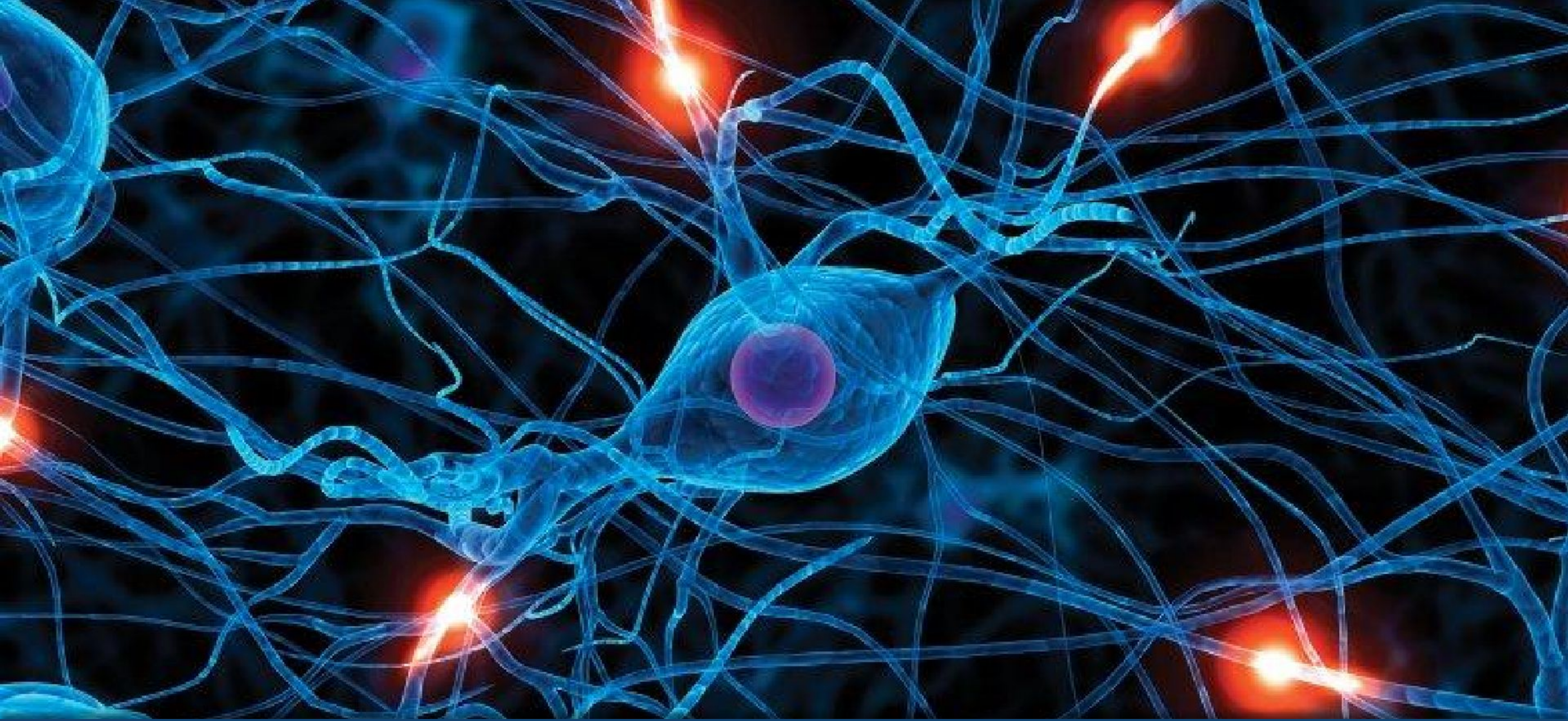


Let’s dive into Deep Learning

with TensorFlow

Google Developers
Experts

AI Experiments

Teachable Machine:
https://teachablemachine.withgoogle.com/train

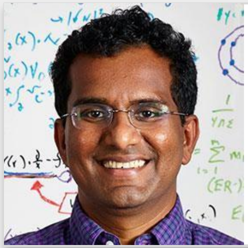
Quick Draw:
https://quickdraw.withgoogle.com/

Experiments with Google:
https://experiments.withgoogle.com/collection/ai

Hype or Reality?

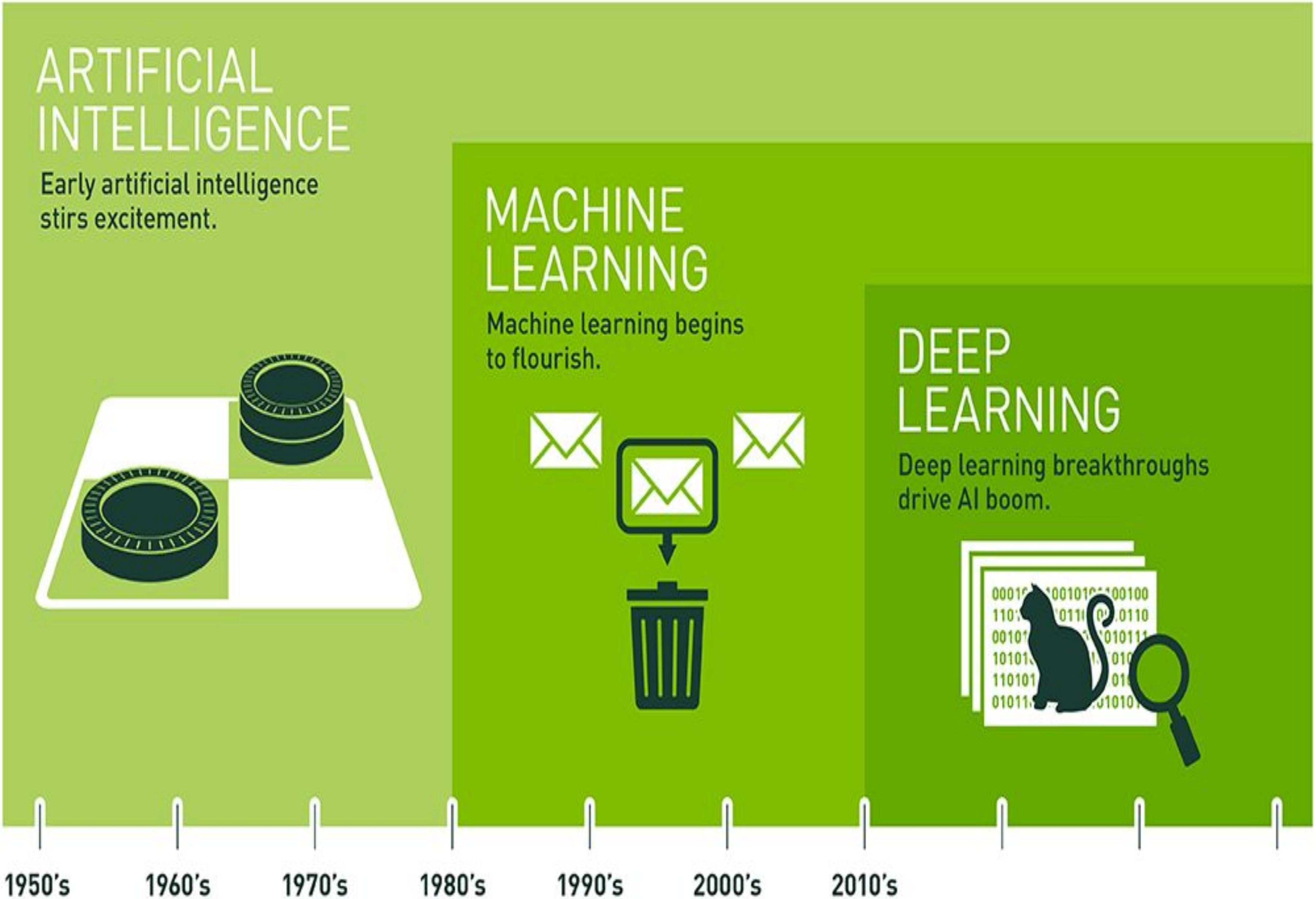
I have worked all my life in Machine Learning, and I have never seen one algorithm knock over benchmarks like Deep Learning
– Andrew Ng (Stanford & Baidu)

Deep Learning is an algorithm which has no theoretical limitations of what it can learn; the more data you give and the more computational time you provide, the better it is *– Geoffrey Hinton (Google)*



Human-level artificial intelligence has the potential to help humanity thrive more than any invention that has come before it – *Dileep George (Co-Founder Vicarious)*

For a very long time it will be a complementary tool that human scientists and human experts can use to help them with the things that humans are not naturally good *– Demis Hassabis (Co-Founder* DeepMind*)*



ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

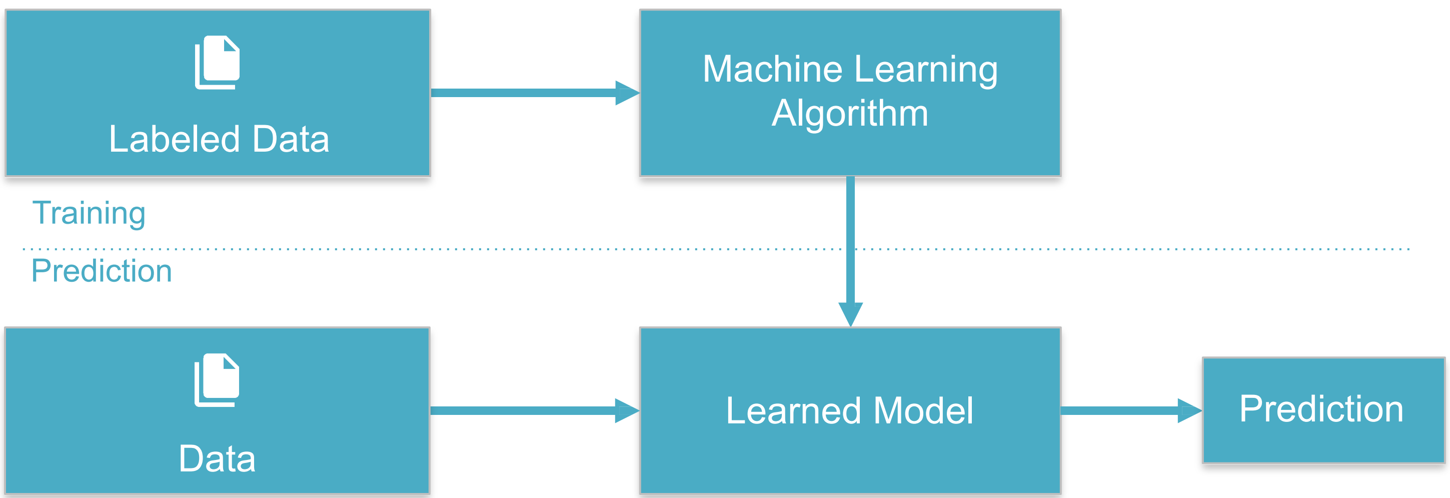
Basic Terminologies

- Features
- Labels
- Examples
    - Labelled example
    - Unlabelled example
- Models (Train and Test)
    - Classification model
    - Regression model

Machine Learning - Basics

Introduction

Machine Learning is a type of Artificial Intelligence that provides computers with the ability to learn without being explicitly programmed.



Provides various techniques that can learn from and make predictions on data

Machine Learning - Basics

Learning Approaches

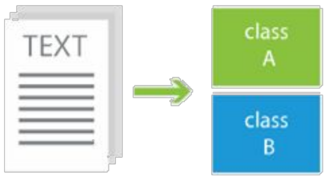
Supervised Learning: Learning with a labeled training set
Example: email spam detector with training set of already labeled emails

Unsupervised Learning: Discovering patterns in unlabeled data
Example: cluster similar documents based on the text content

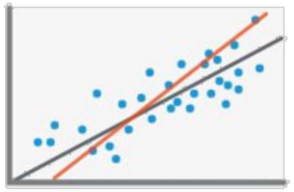
Reinforcement Learning: learning based on feedback or reward
Example: learn to play chess by winning or losing

Machine Learning - Basics

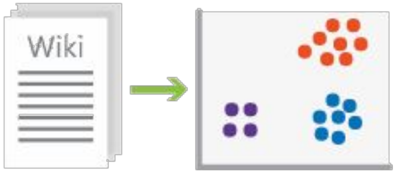
Problem Types



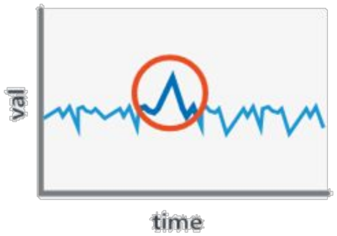
Classification
(supervised – predictive)



Regression
(supervised – predictive)



Clustering
(unsupervised – descriptive)



Anomaly Detection
(unsupervised– descriptive)

What is Deep Learning?

Part of the machine learning field of learning representations of data. Exceptional effective at learning patterns.

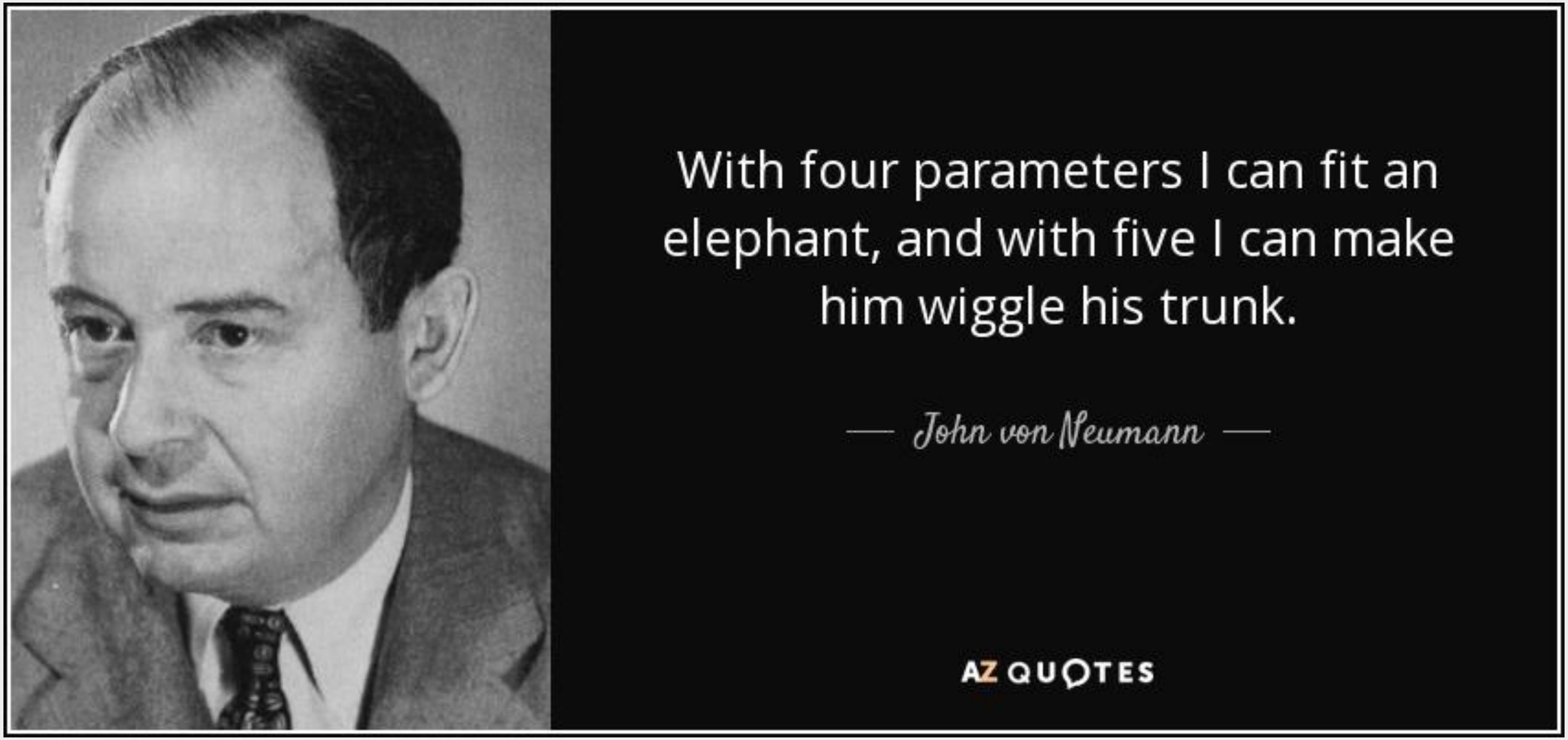
Utilizes learning algorithms that derive meaning out of data by using a hierarchy of multiple layers that mimic the neural networks of our brain.

If you provide the system tons of information, it begins to understand it and respond in useful ways.

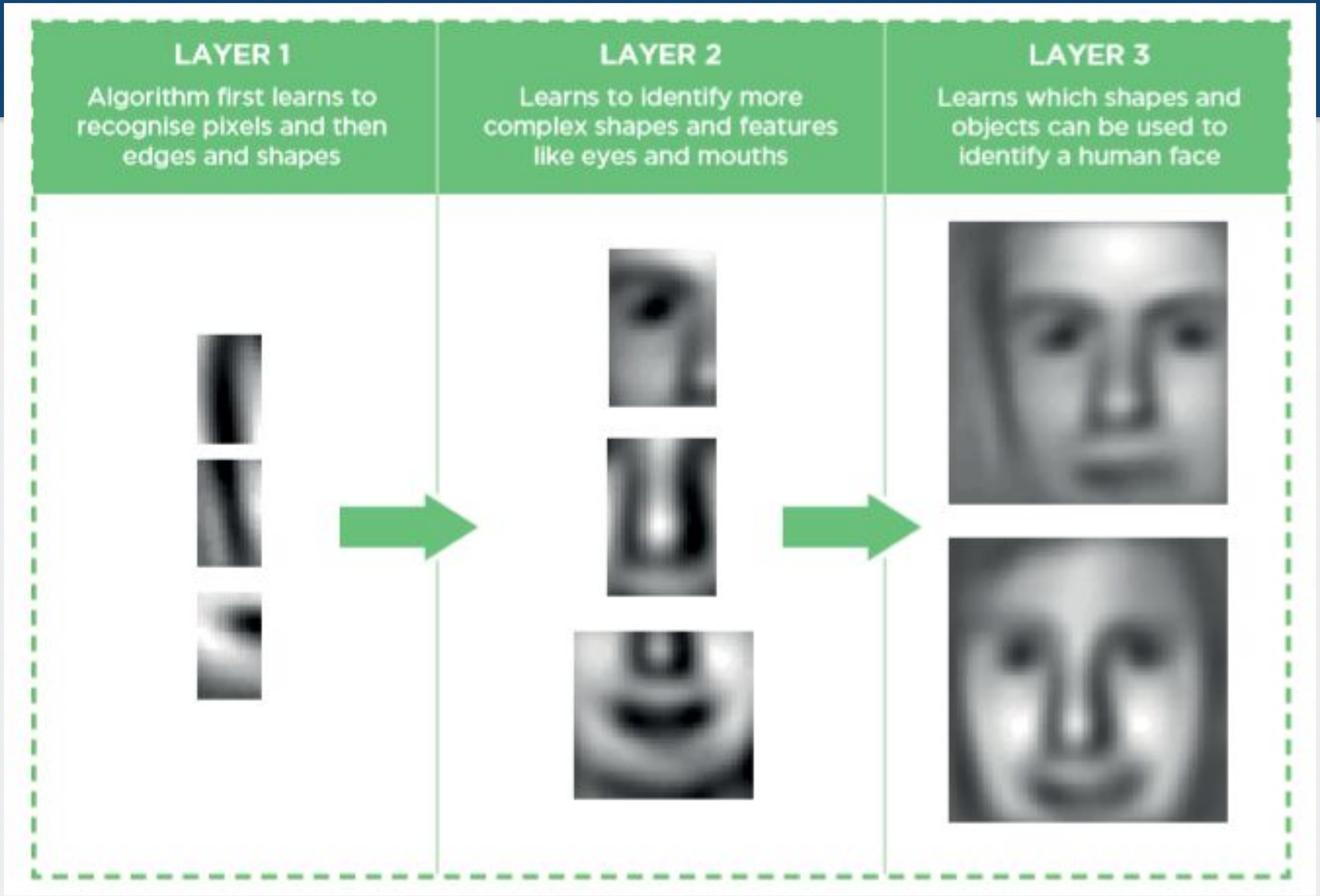
Feature Engineering

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- A machine learning **model can't directly see, hear, or sense input** examples. Machine learning models typically expect examples to be represented as **real-numbered vectors**.
- **Feature engineering** means transforming raw data into a feature vector of 1's and 0's which Machine can understand.

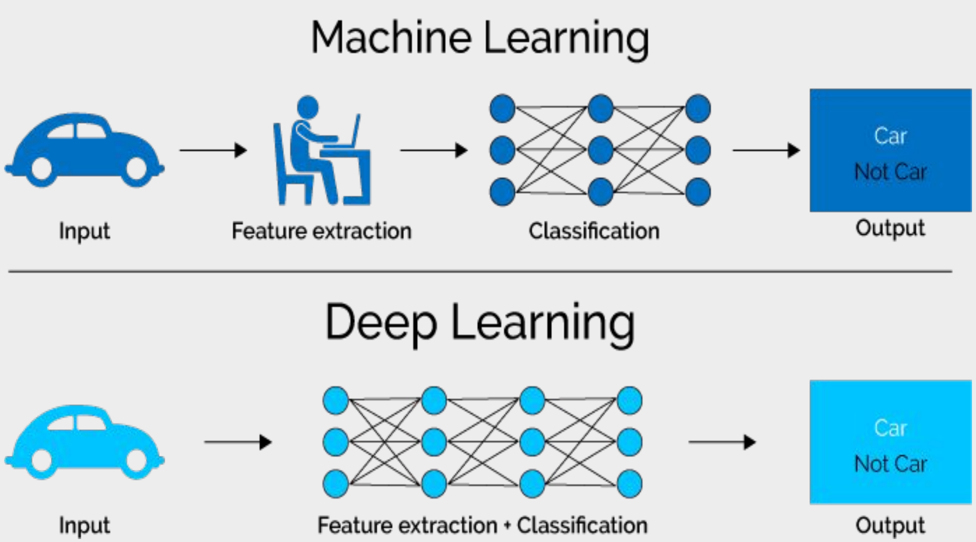

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
— John von Neumann —
AZ QUOTES

How does DL works?



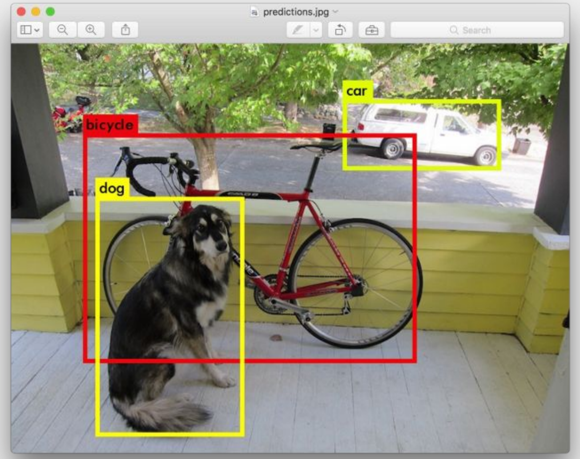
Why DL over traditional ML?

- Deep Learning requires high-end machines contrary to traditional Machine Learning algorithms
- Thanks to **GPUs and TPUs**
- **No more Feature Engineering!!**
- **ML:** most of the applied features need to be identified by an domain expert in order to reduce the complexity of the data and make patterns more visible to learning algorithms to work
- **DL:** they try to learn high-level features from data in an incremental manner.



Why DL over traditional ML?

- **The problem solving approach:**
  - **Deep Learning** techniques tend to solve the problem **end to end**
  - **Machine learning** techniques need the problem statements to **break down to different parts** to be solved first and then their results to be combine at final stage
- For example for a multiple object detection problem, Deep Learning techniques like Yolo net take the image as input and provide the location and name of objects at output
- But in usual Machine Learning algorithms uses SVM, a bounding box object detection algorithm then HOG as input to the learning algorithm in order to recognize relevant objects



What changed?

Big Data
(Digitalization)

Computation
(Moore's Law, GPUs)

Algorithmic Progress

When to use DL or not over Others?

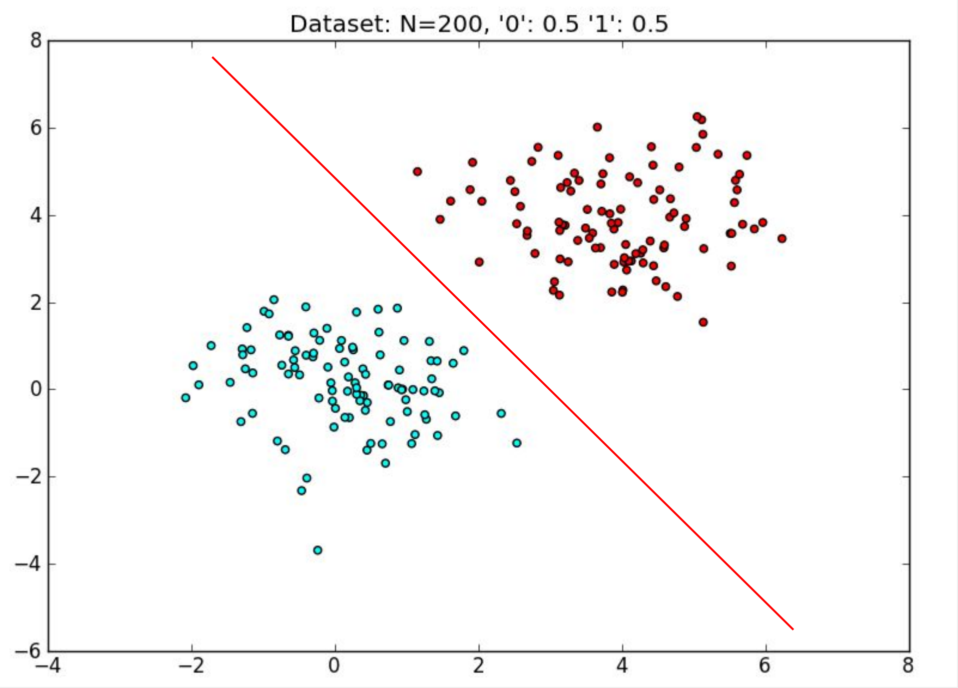
1. Deep Learning outperforms other techniques if the **data size is large**. But with small data size, traditional Machine Learning algorithms are preferable
2. Finding large amount of “Good” data is always a painful task but hopefully not now on, Thanks to the all new **Google Dataset Search Engine** ☺
3. Deep Learning techniques need to have **high end infrastructure** to train in reasonable time
4. When there is **lack of domain understanding for feature introspection**, Deep Learning techniques outshines others as you have to worry less about feature engineering
5. **Model Training time:** a Deep Learning algorithm may take weeks or months whereas, traditional Machine Learning algorithms take few seconds or hours
6. **Model Testing time:** DL takes much lesser time as compare to ML
7. DL never reveals the “how and why” behind the output- it’s a **Black Box**
8. Deep Learning really shines when it comes to **complex problems such as image classification, natural language processing, and speech recognition**
9. Excels in tasks where the basic unit (pixel, word) has very little meaning in itself, but **the combination of such units has a useful meaning**

TensorFlow

- TensorFlow is an open-source library for Machine Intelligence
- It was developed by the **Google Brain** and released in 2015
- It provides high-level APIs to help implement many machine learning algorithms and develop complex models in a simpler manner.

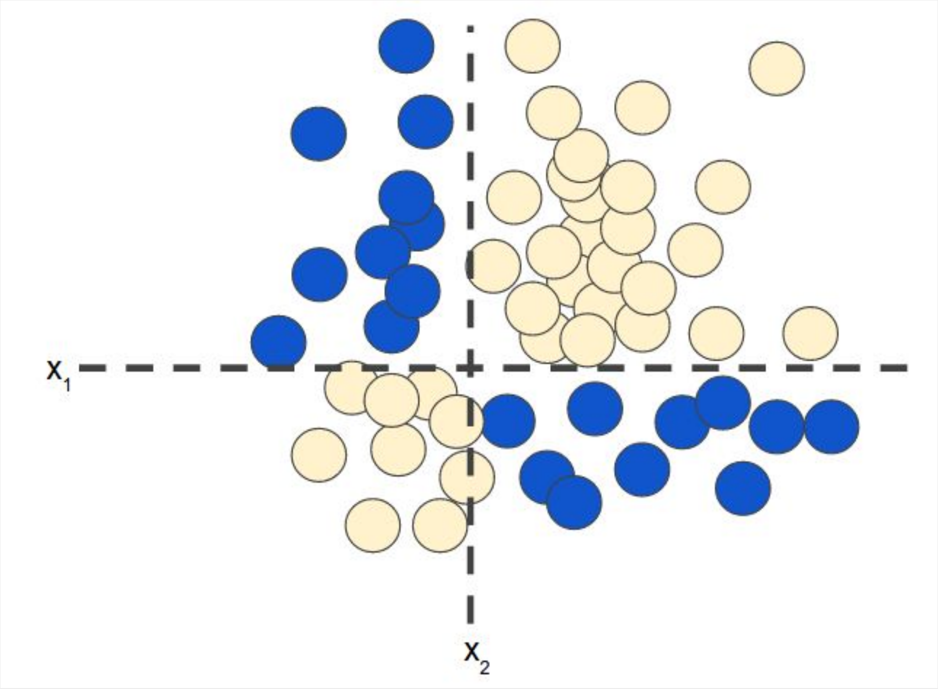
- **What is a tensor?**
  - A mathematical object, analogous to but more general than a vector, represented by an array of components that are functions of the coordinates of a space.
  - TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays known as ‘tensors’.

How do you Classify these Points?



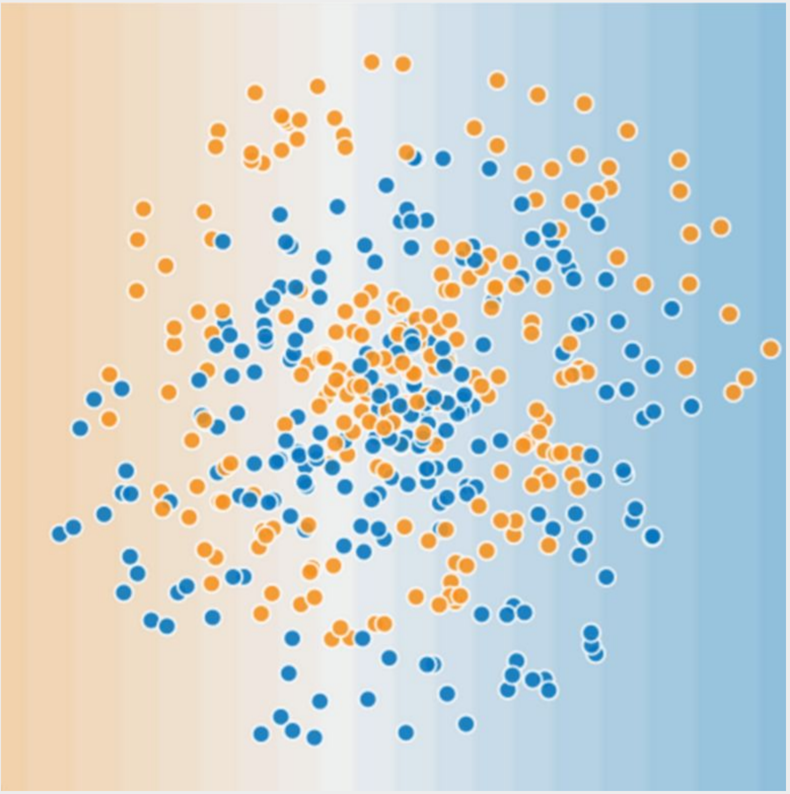
$$b + w_1 x_1$$

Okay, how do you Classify these Points?



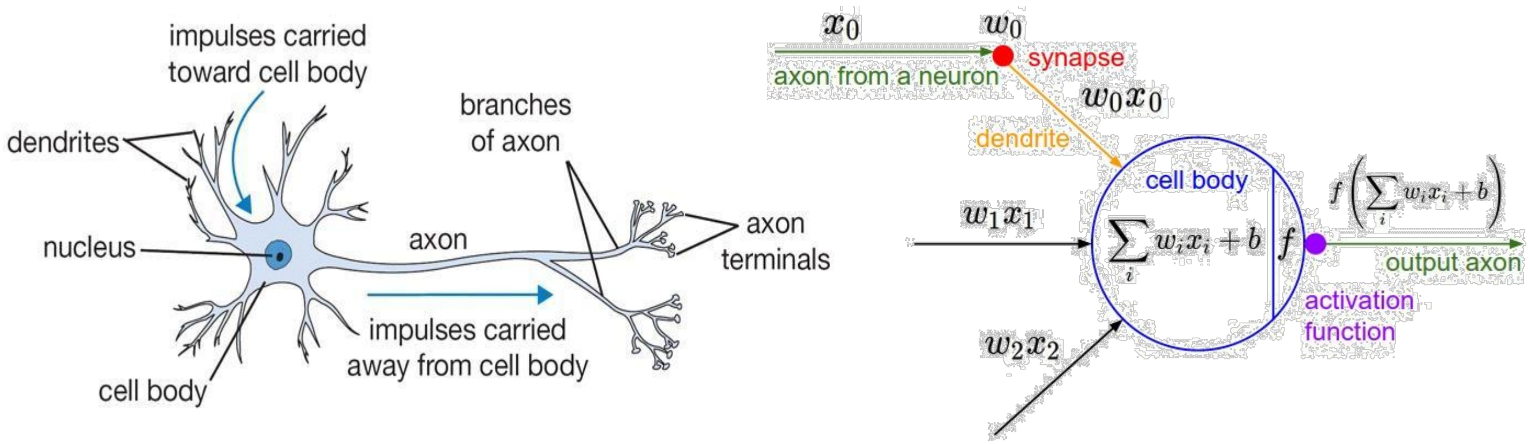
$$x_3 = x_1 x_2$$

Okay okay, but now?



Non linearities are tough to model. In complex datasets, the task becomes very cumbersome. What is the solution?

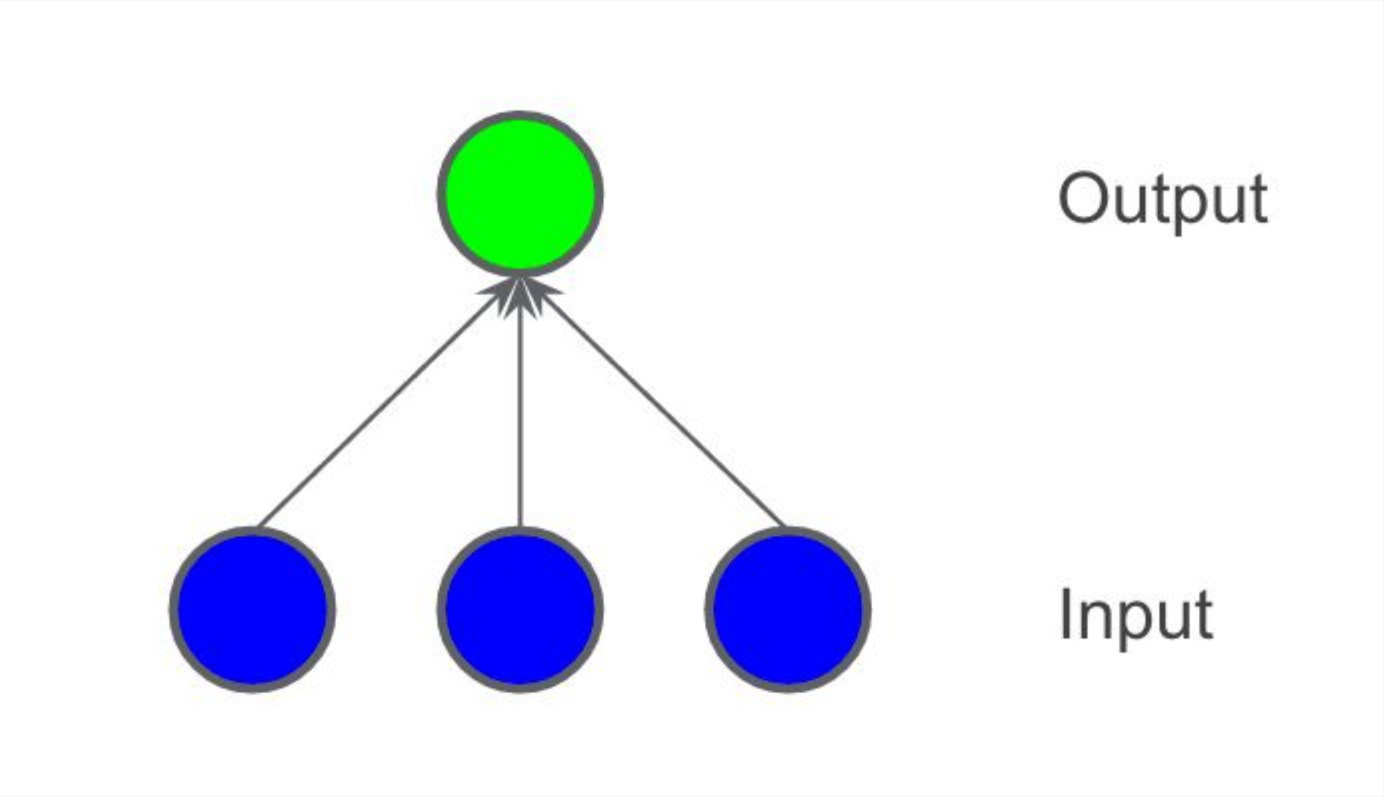
Inspired by the human Brain



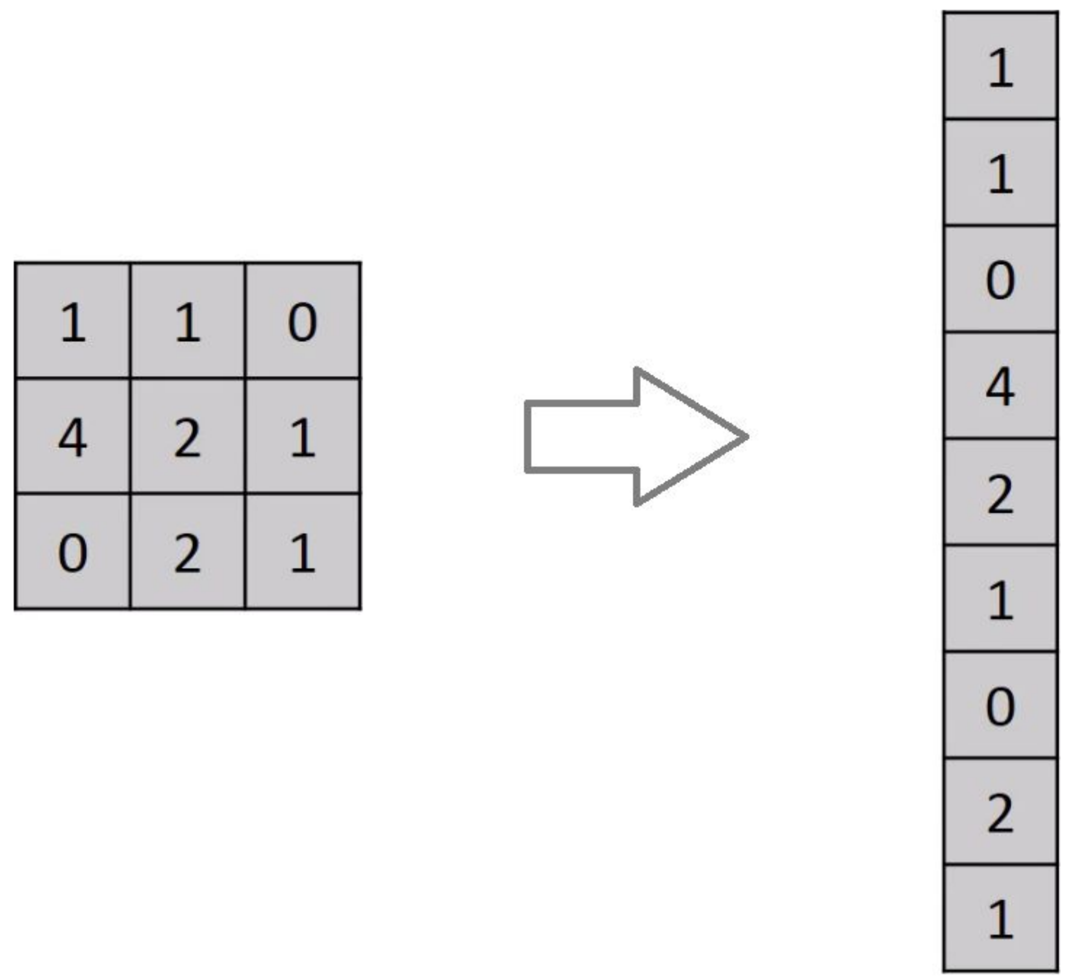
An artificial neuron contains a nonlinear activation function and has several incoming and outgoing weighted connections.

Neurons are trained to filter and detect specific features or patterns (e.g. edge, nose) by receiving weighted input, transforming it with the activation function und passing it to the outgoing connections.

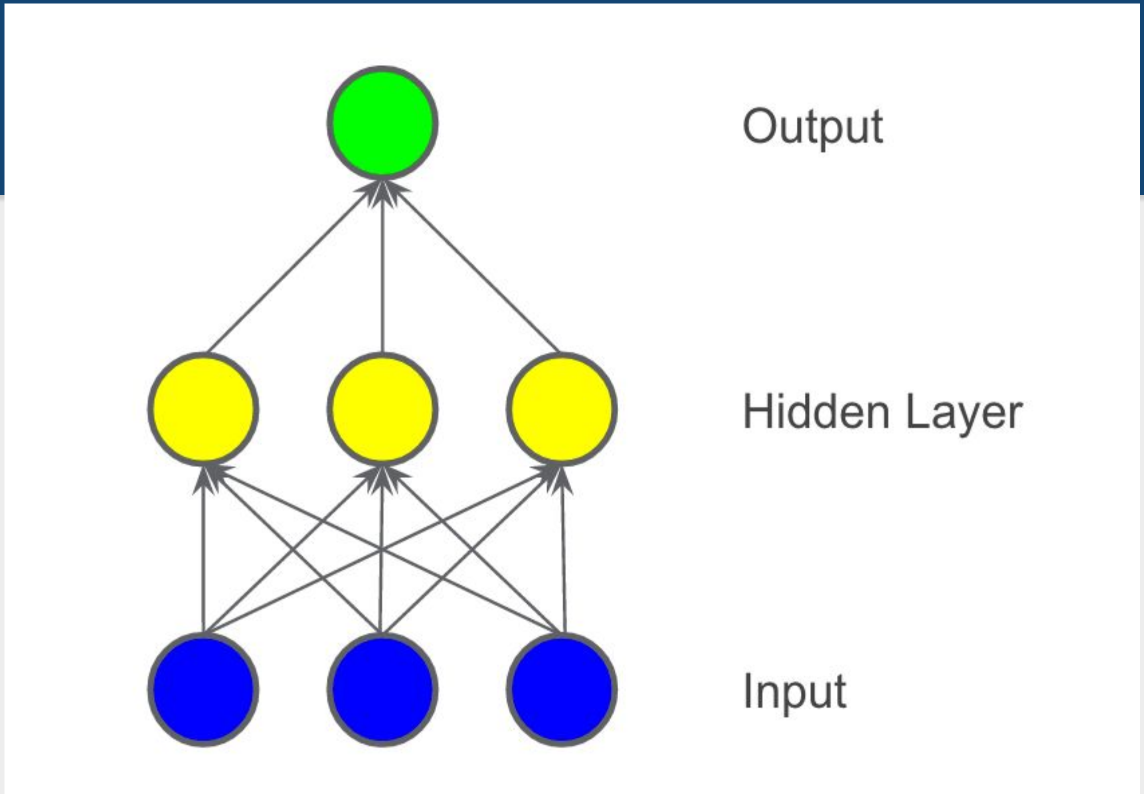
Modelling a Linear Equation



Flattened input image

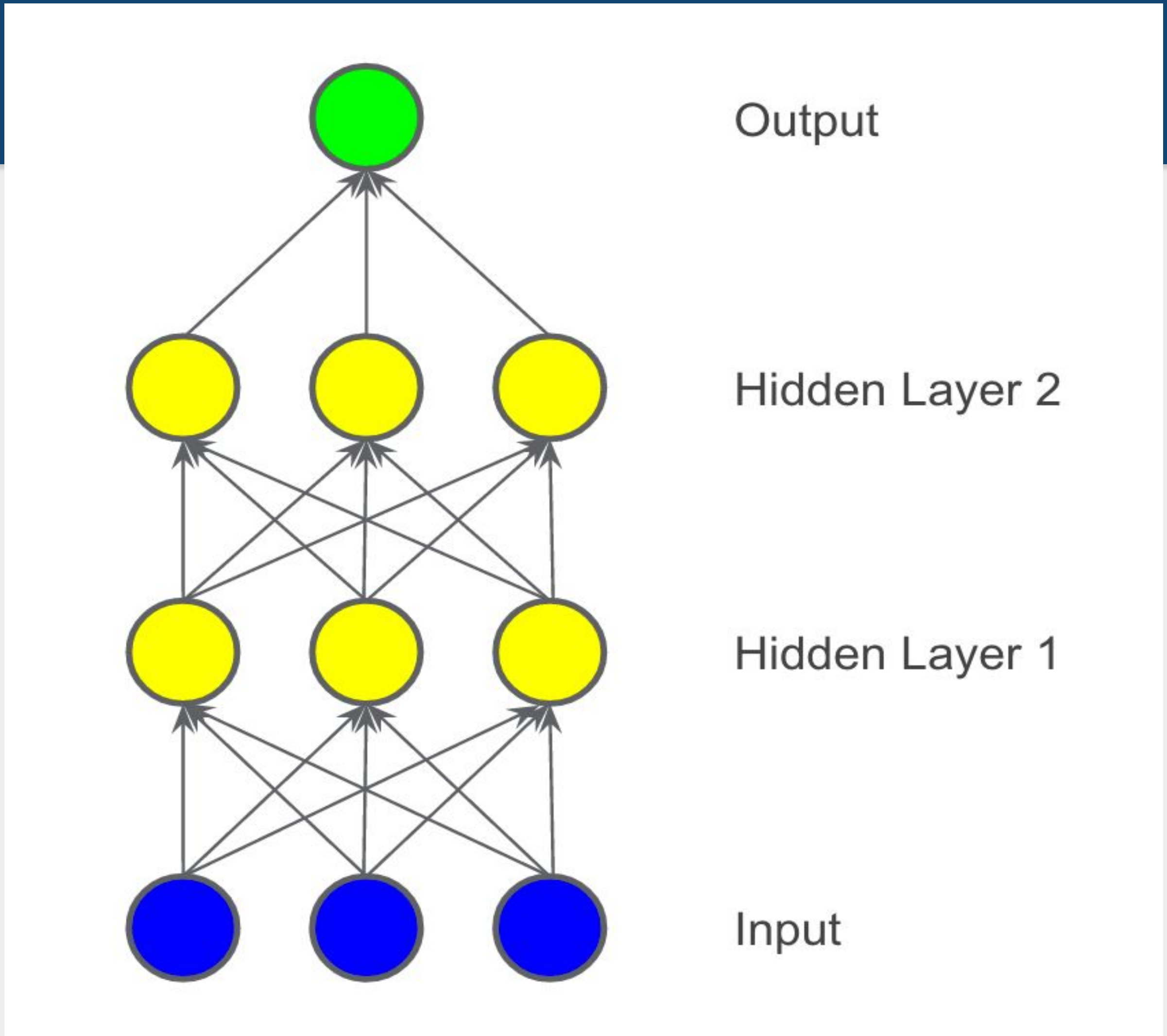


How to deal with Non-linear Problems?

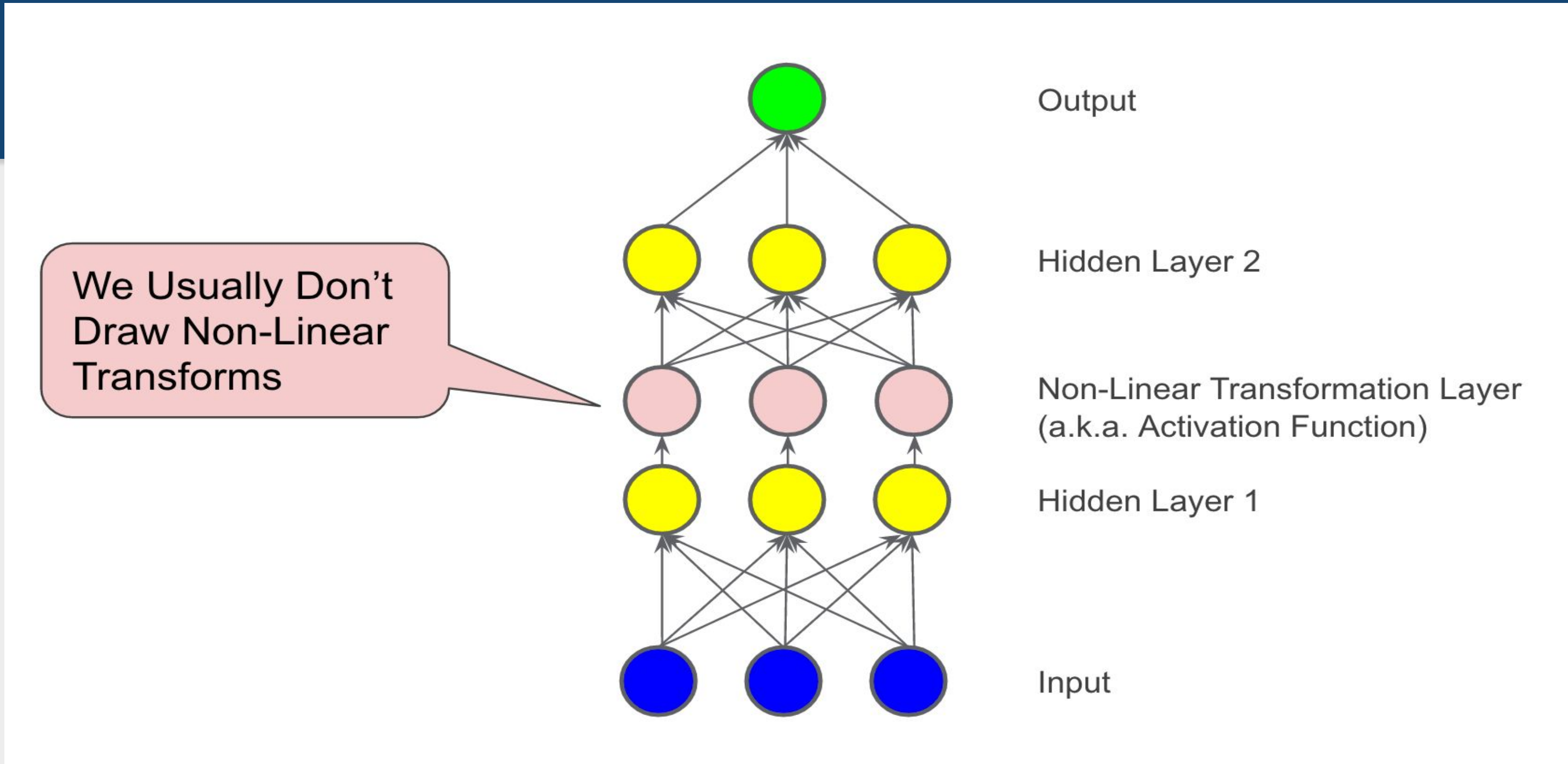


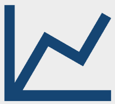
We added a hidden layer of intermediary values. Each yellow node in the hidden layer is a weighted sum of the blue input node values. The output is a weighted sum of the yellow nodes.

Is it linear? What are we missing?



Activation Functions

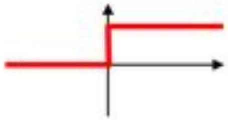
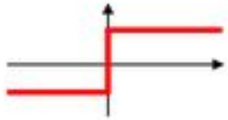
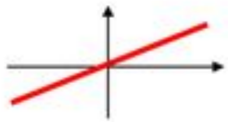




Non-linearity is needed to learn complex (non-linear) representations of data, otherwise the NN would be just a linear function.


"Kaun hain ye log? Kaha se ate hain yeh?"
gifs.com

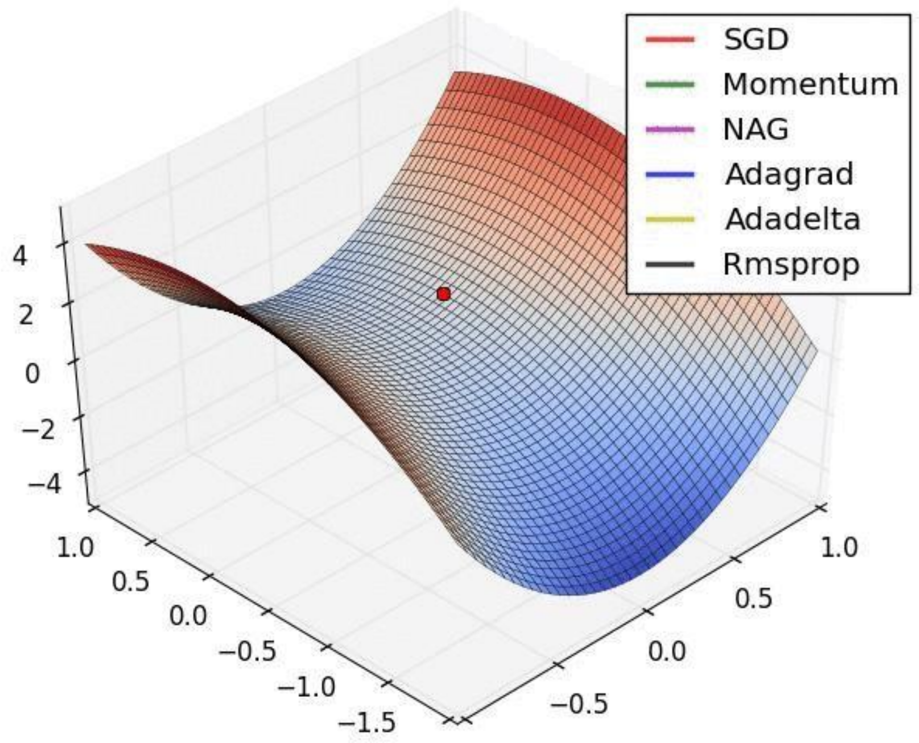
Non-linear Activation Functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

Let’s build our first NN, DNN, CNN,,,,

Visit:
https://colab.research.google.com/drive/1otXTb62ZWxOmjaOM33PuJo1_PJ0zu0et?usp=sharing

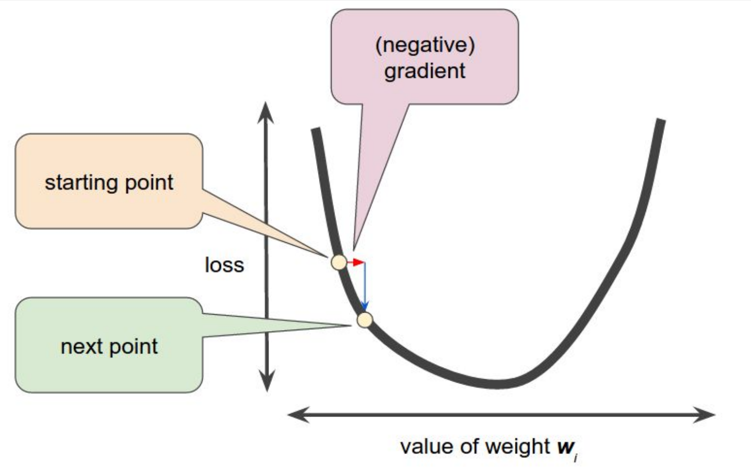
Gradient Descent



Gradient Descent finds the (local) minimum of the cost function (used to calculate the output error) and is used to adjust the weights

Gradient Descent

- Convex problems have only one minimum; that is, only one place where the slope is exactly 0. **That minimum is where the loss function converges**
- The gradient descent algorithm then calculates the gradient of the loss curve at the starting point. In brief, a gradient is a vector of partial derivatives
- A gradient is a vector and hence has magnitude and direction
- **The gradient always points in the direction of the minimum.** The gradient descent algorithm takes a step in the direction of the **negative gradient** in order to reduce loss as quickly as possible



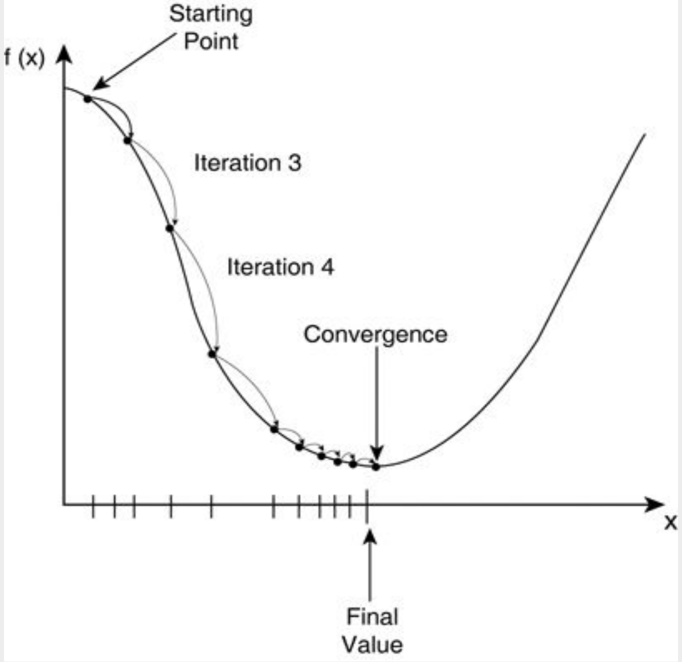
Gradient Descent

- The algorithm given below signifies Gradient Descent algorithm
- In our case,
- Θ_j will be w_i
- ***α*** is the learning rate
- $J(\Theta)$ is the cost function

Repeat until convergence {

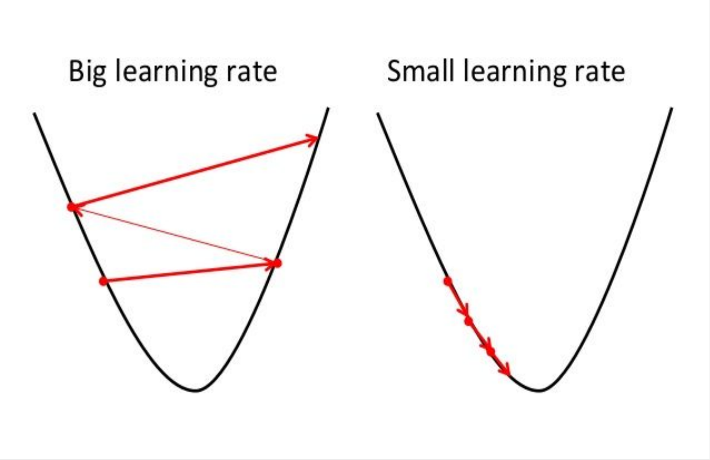
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}



The Learning Rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) to determine the next point.
- For example, if the gradient magnitude is 2.5 and the learning rate is 0.01, then the gradient descent algorithm will pick the next point 0.025 away from the previous point.
- **A Hyperparameter!**
- Think of it as in real life. Some of us slow learners while some others are quicker learners
- **Small learning rate -> learning will take too long**
- **Large learning rate -> may overshoot the minima**

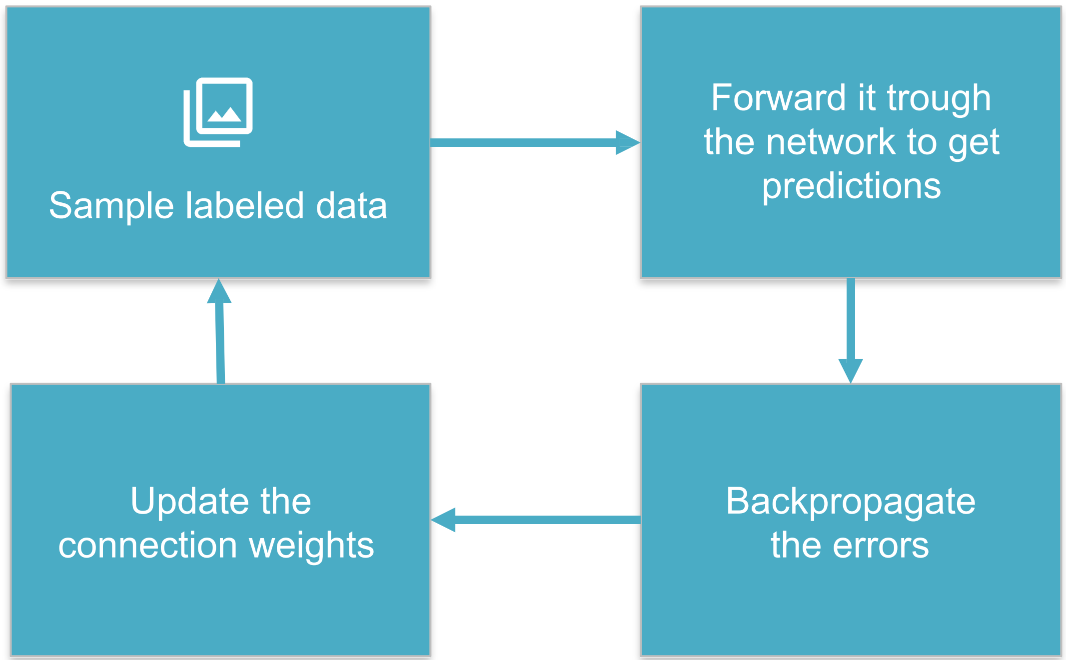


But how the model will LEARN?

BACKPROPAGATION

Deep Learning

The Training Process



Learns by generating an error signal that measures the difference between the predictions of the network and the desired values and then using this error signal to change the weights (or parameters) so that predictions get more accurate.

Still not so Perfect!

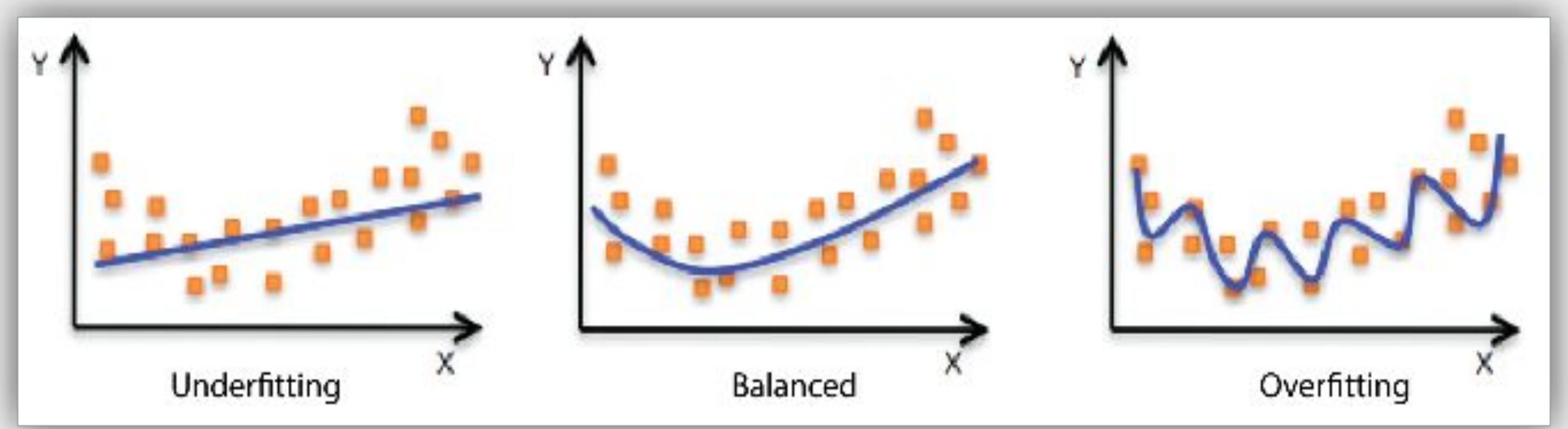
Backprop can go wrong

- **Vanishing Gradients:**
- The gradients for the lower layers (closer to the input) can become very small. In deep networks, computing these gradients can involve taking the product of many small terms

- **Exploding Gradients:**
- If the weights in a network are very large, then the gradients for the lower layers products of many large terms. In this case you can have exploding gradients: gradients that get too large to converge

Ooooooverfitting = Game Over

- An overfit model gets a low loss during training but does a poor job predicting new data
- Overfitting is caused by making a model more complex than necessary.
- The fundamental tension of machine learning is between fitting our data well, but also fitting the data as simply as possible

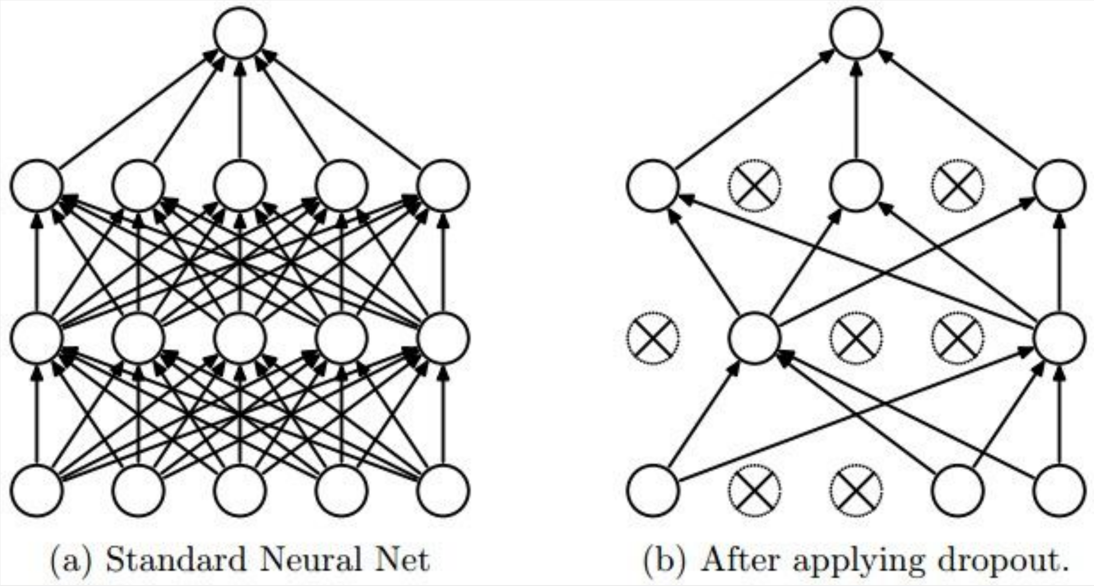


Solution

Dropout Regularization

It works by randomly "dropping out" unit activations in a network for a single gradient step. The more you drop out, the stronger the regularization:

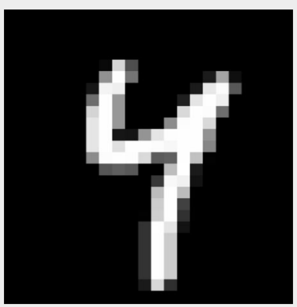
0.0 -> No dropout regularization.
1.0 -> Drop out everything. The model learns nothing
values between 0.0 and 1.0 -> More useful



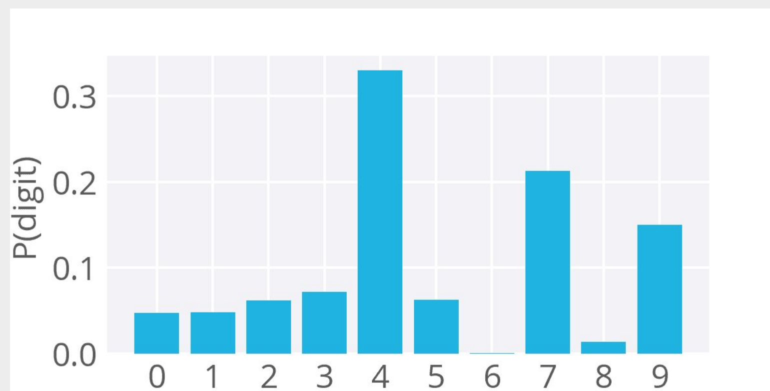
Softmax

Now the problem with sigmoid function in multi-class classification is that the values calculated on each of the output nodes may not necessarily sum up to one.

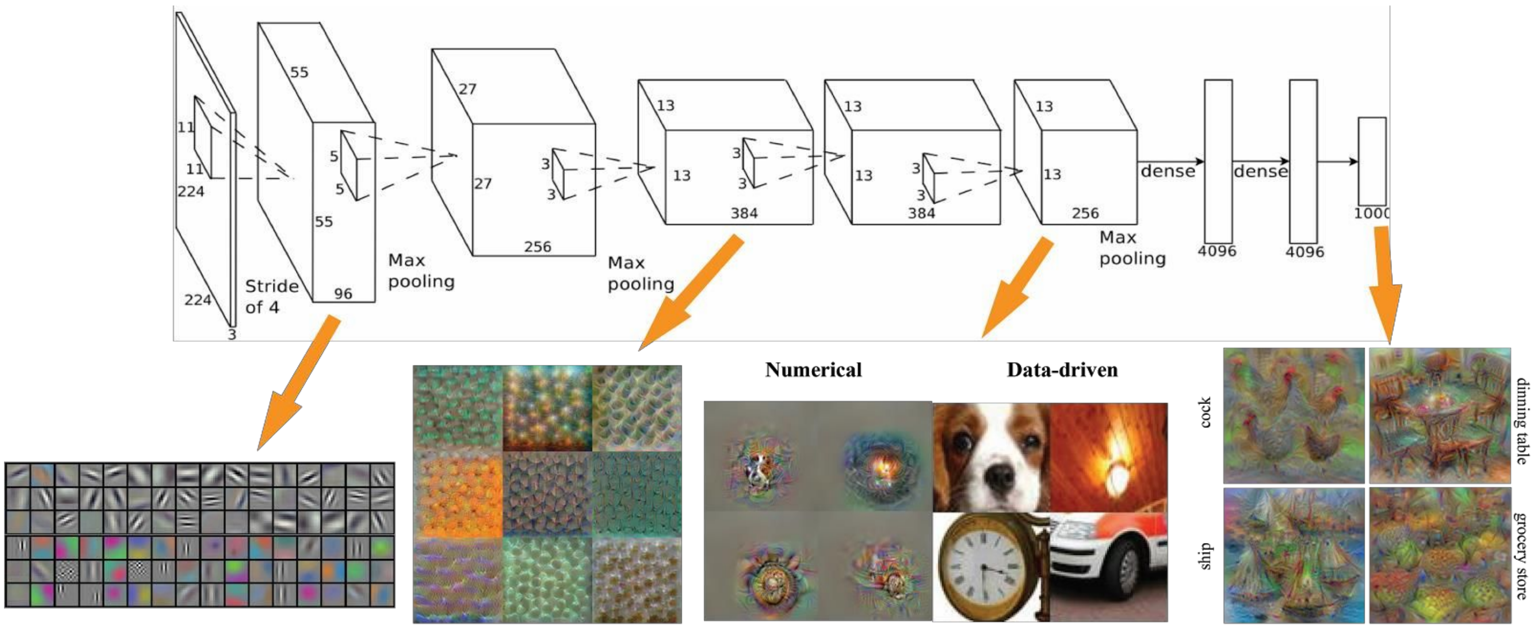
The softmax function used for multi-classification model returns the probabilities of each class.



$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$



Convolutional Neural Nets (CNN)



Conv 1: Edge+Blob

Conv 3: Texture

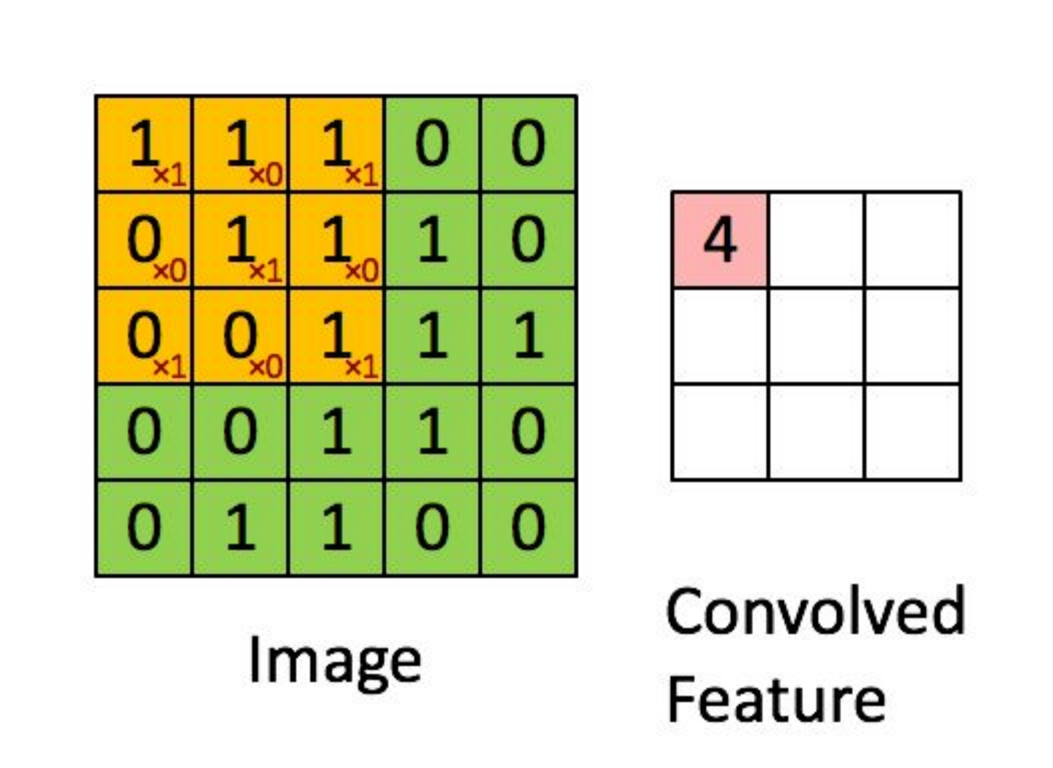
Conv 5: Object Parts

Fc8: Object Classes

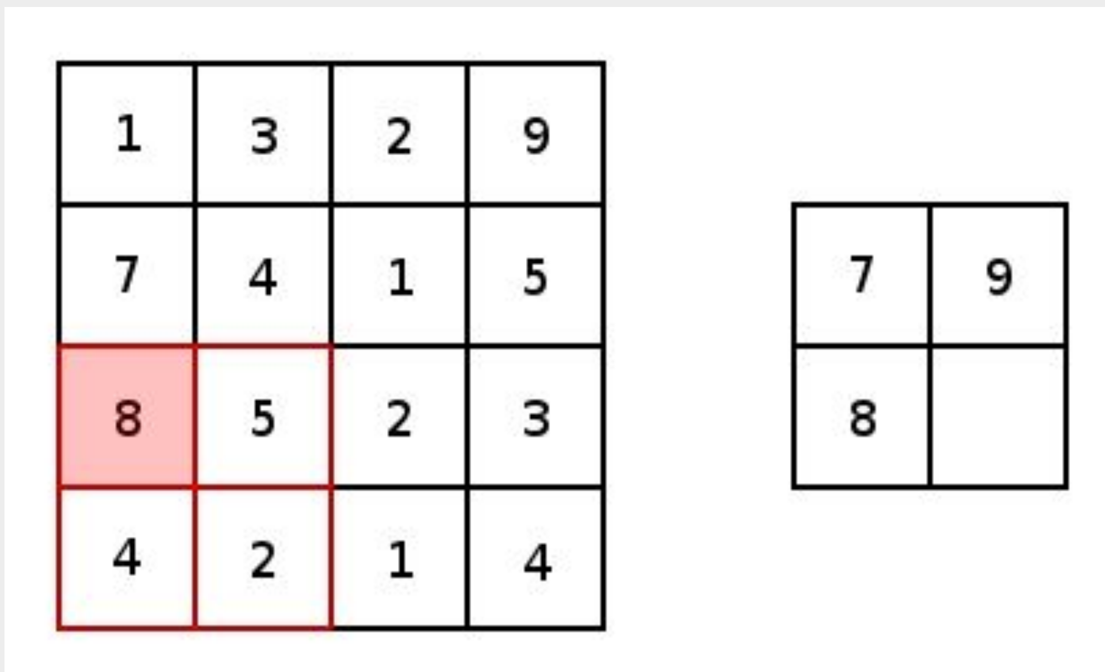
Convolution layer is a feature detector that automagically learns to filter out not needed information from an input by using convolution kernel.

Pooling layers compute the max or average value of a particular feature over a region of the input data *(downsizing of input images).* Also helps to detect objects in some unusual places and reduces memory size.

Convolution



Max Pooling



Takeaways

Humans are genius!!!
Machines that learn to represent the world from experience.

Deep Learning is no magic! Just statistics in a black box, but exceptional effective at learning patterns.

We haven't figured out creativity and human-empathy.
Neural Networks are not the solution to every problem.

Transitioning from research to consumer products. Will make the tools you use every day work better, faster and smarter.

Questions…??

Comments

Suggestions ☺

Happy Learning!

Let's Connect

@CharmiChokshi