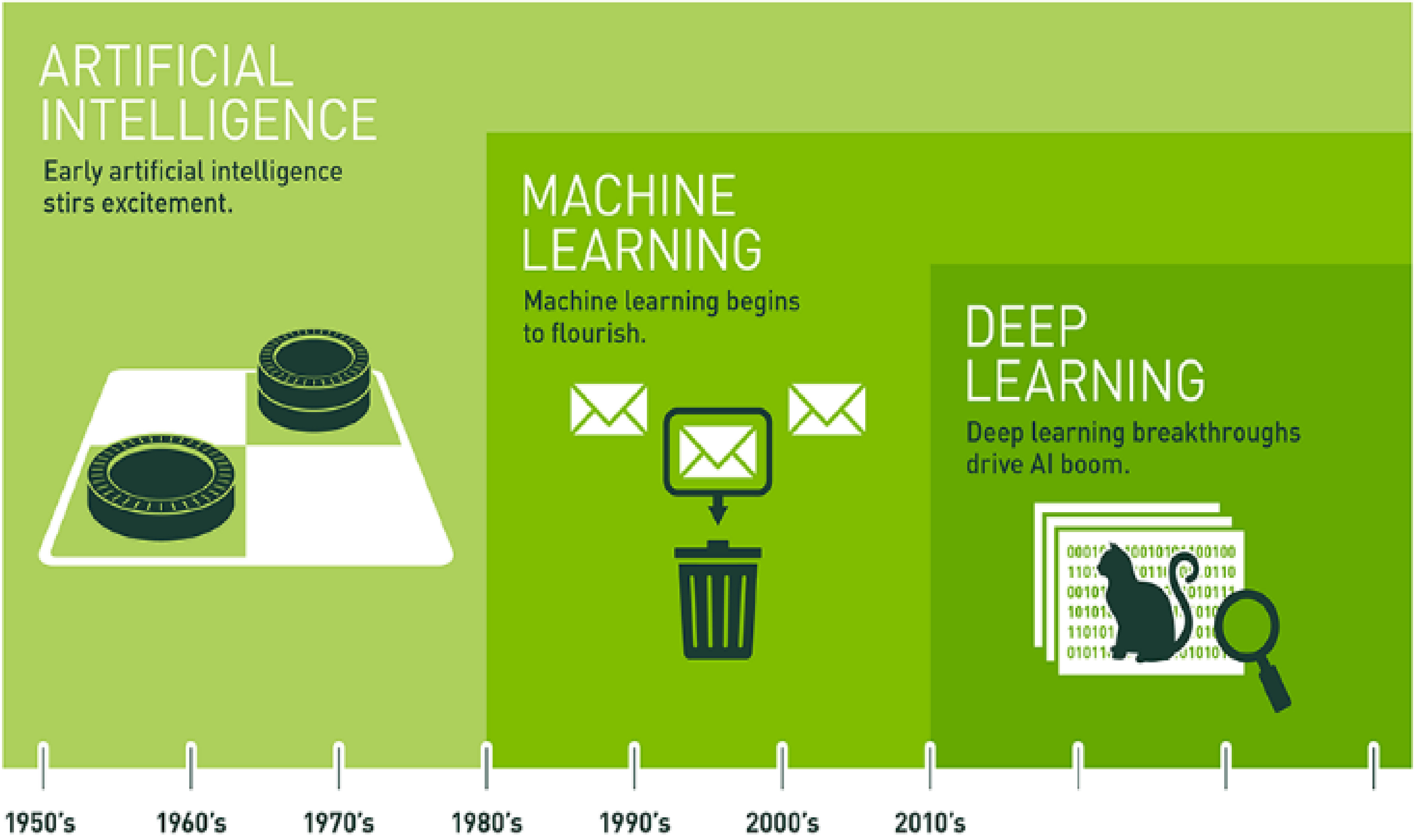
Introduction to Deep Learning



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Manufacturing

- Predictive maintenance or condition monitoring
- Warranty reserve estimation
- Propensity to buy
- Demand forecasting
- Process optimization
- Telematics

Retail

- Predictive inventory planning
- Recommendation engines
- Upsell and cross-channel marketing
- Market segmentation and targeting
- Customer ROI and lifetime value

Healthcare and Life Sciences

- Alerts and diagnostics from real-time patient data
- Disease identification and risk stratification
- Patient triage optimization
- Proactive health management
- Healthcare provider sentiment analysis

Travel and Hospitality

- Aircraft scheduling
- Dynamic pricing
- Social media – consumer feedback and interaction analysis
- Customer complaint resolution
- Traffic patterns and congestion management

Financial Services

- Risk analytics and regulation
- Customer Segmentation
- Cross-selling and up-selling
- Sales and marketing campaign management
- Credit worthiness evaluation

Energy, Feedstock, and Utilities

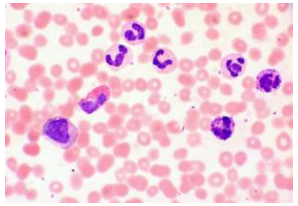
- Power usage analytics
- Seismic data processing
- Carbon emissions and trading
- Customer-specific pricing
- Smart grid management
- Energy demand and supply optimization

Aipoly video

https://www.youtube.com/watch?v=XMdct-5bERQ

Deep Learning treads where others dare not

- **Computer vision** is now completely dominated by Deep Learning :
- **DeepMind**-detects diabetic retinopathy slightly better than doctors
- **Google Photos**-automatically can tag photos
- **IBM**-working to detect skin cancer
- **SigTuple**-medical diagnostics from blood smears



- Interpreting the huge amount of **genomic data**: **Deep Genomics**

- **Detecting cancer** from blood **(Freenome)**, accelerating **Drug discovery (Merck)**

- **Speech** recognition : **Google Assistant, Cortana, Siri**

- **Natural Language Processing** : **Google Translate** has been increasing in accuracy

- **Text generation** through Deep Learning: **Legal Zoom** creates documents without lawyers

- **Music composition** using Deep Learning : **Aviva Technologies, Sony**

- Improving retail : **AmazonGo**

Deep Learning: Applications

DEEP LEARNING EVERYWHERE

INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation

MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery

MEDIA & ENTERTAINMENT

Video Captioning
Video Search
Real Time Translation

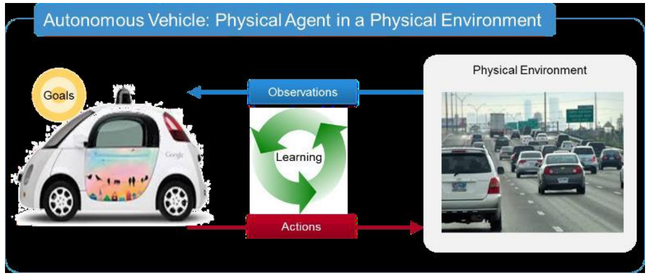
SECURITY & DEFENSE

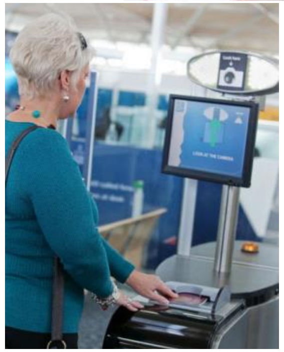
Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES

Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

Deep Learning by Accenture





Claim Process Automation

- To detect automatically the amount of damage from car image
- Order spares, settle claim
- Detect fraud

Perception modules process sensory information to

•Recognize road boundaries, recognize pedestrians

•Recognize other vehicles, recognize words in road signs

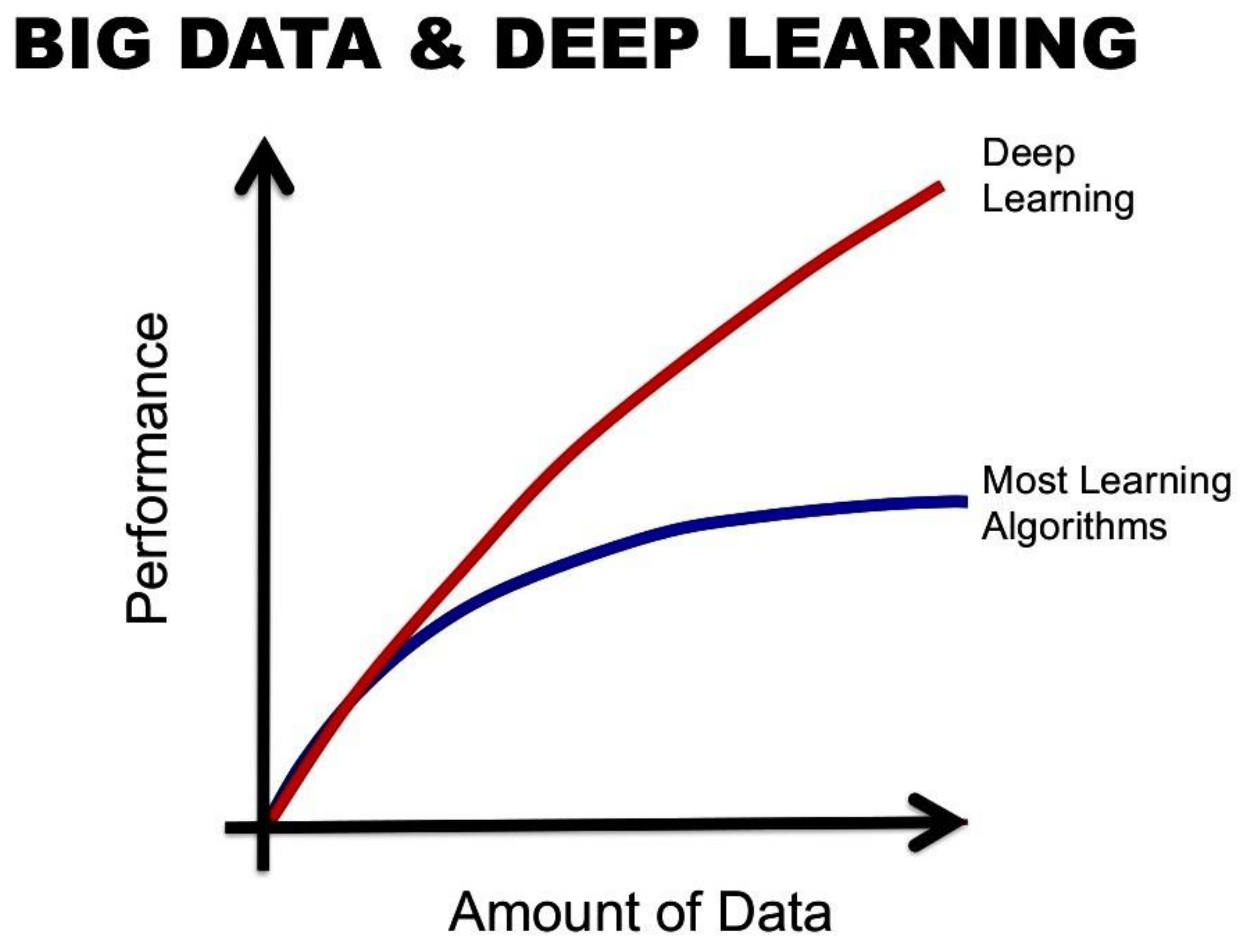
Video Trawling Analytics System (VTAS)

- Provide a video ingestion and search solution to augment forensic post-event investigations
- Deployed by the Criminal Investigation Department of Singapore Police Force.

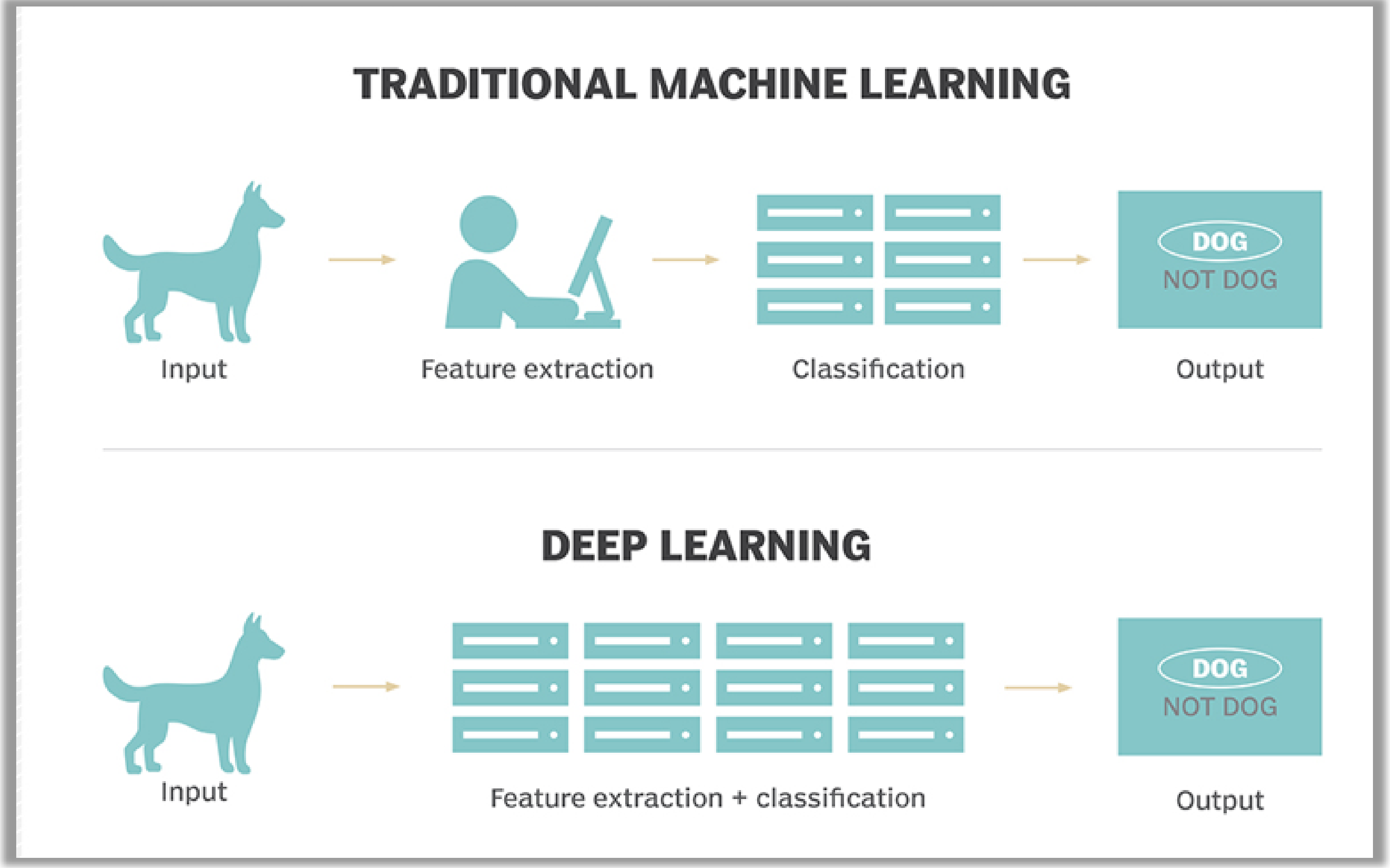
Unique Identity Service Platform

- Automated Passport Control Gates using Face Recognition in London Airports
- Self clearance for EU passport holders in Amsterdam Schiphol

Why Deep Learning



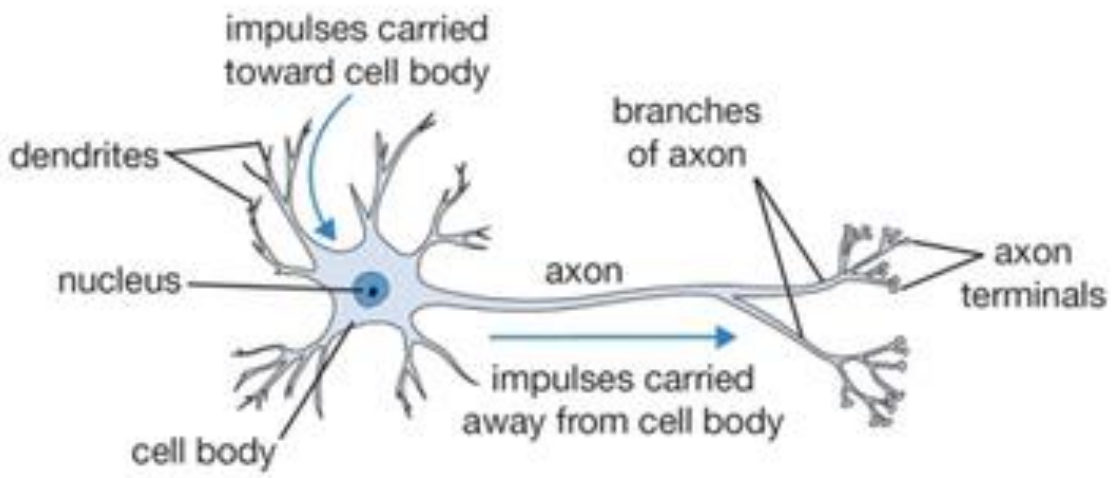
Why Deep Learning



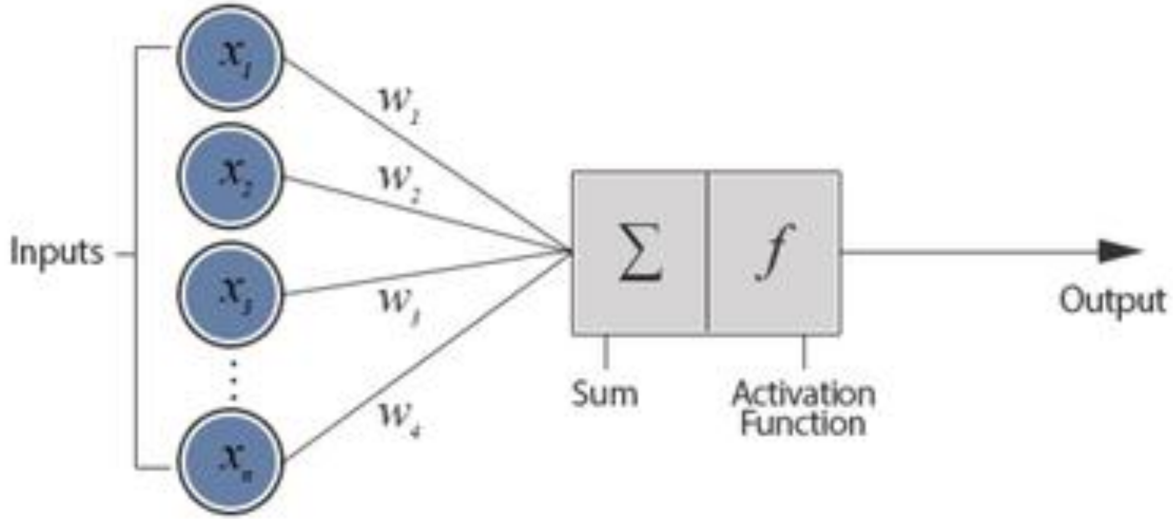
Neural Networks

Lets talk technical …

Biological Inspiration

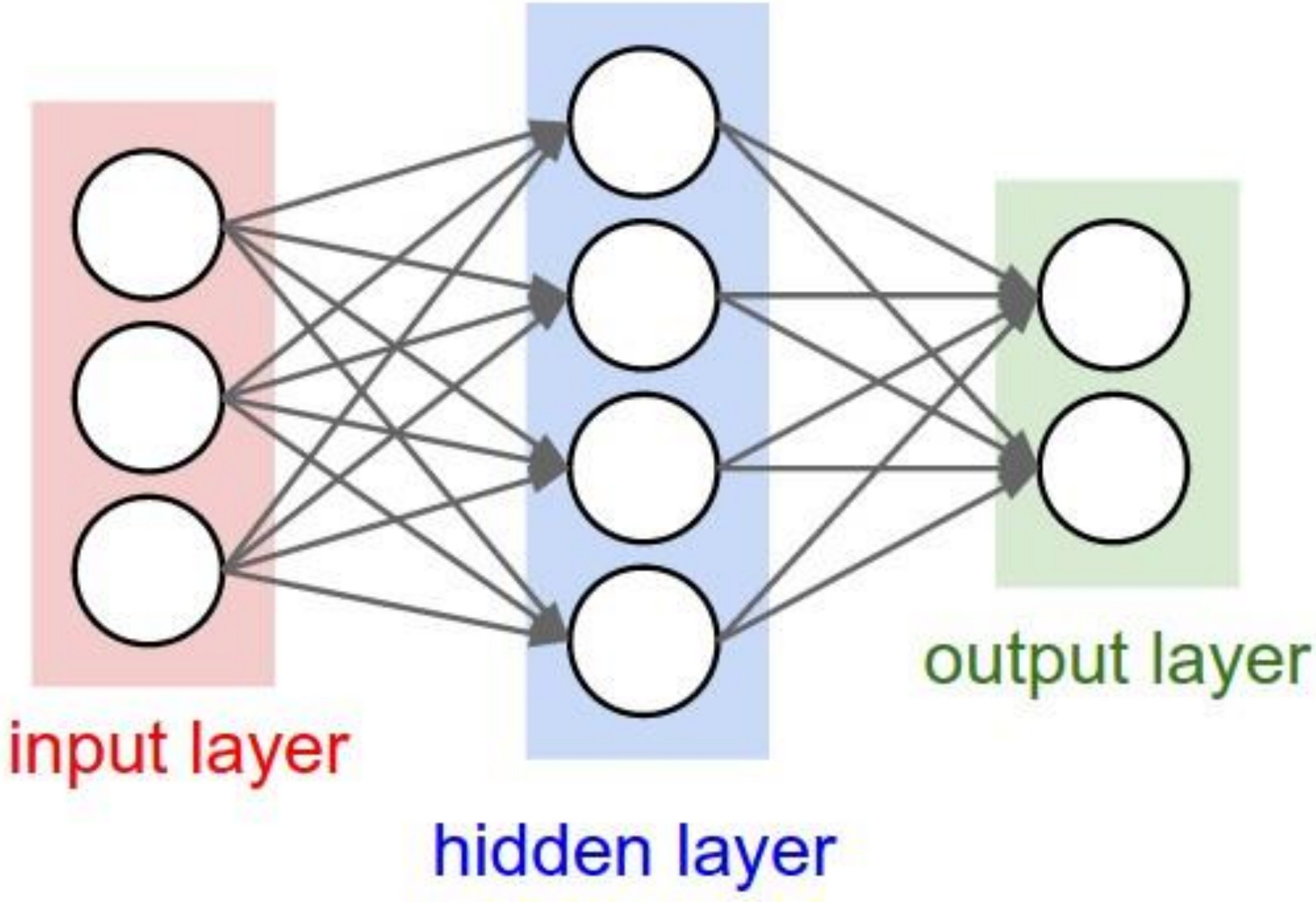


Input Process Output



Input Process Output

A Neural Network

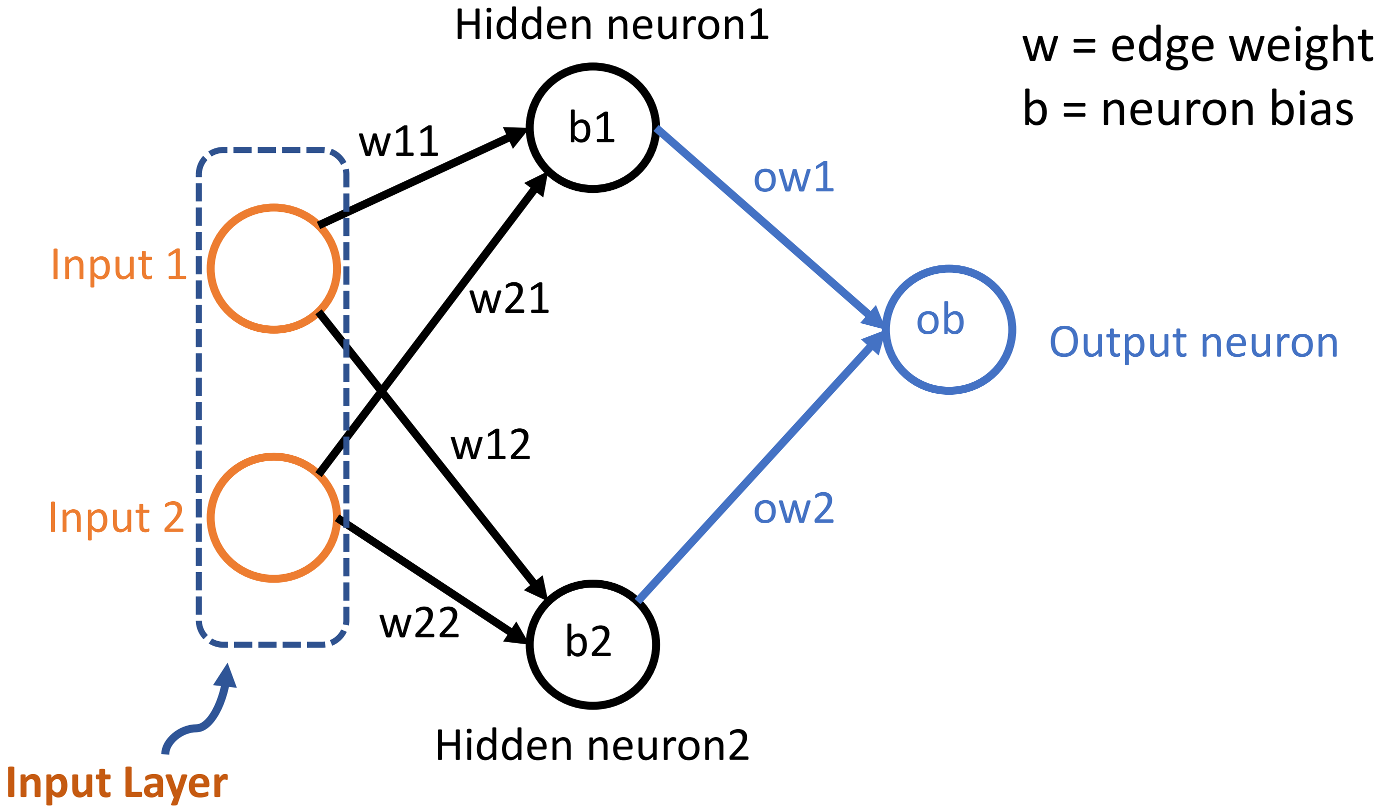


House Price Prediction Data
: *Regression Problem*

Problem Statement : To predict House price based on number of bedrooms and floor level

No. of Bedrooms	Floor level	Price (in 1000 of $)
3	6	300
4	7	400
2	8	350
5	4	375

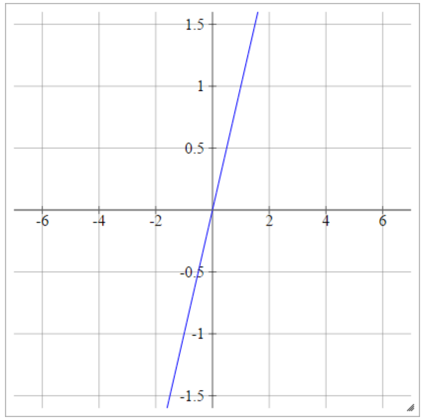
Regression Model



Hidden Layer Activation
= Linear

Output Layer Activation
= Linear

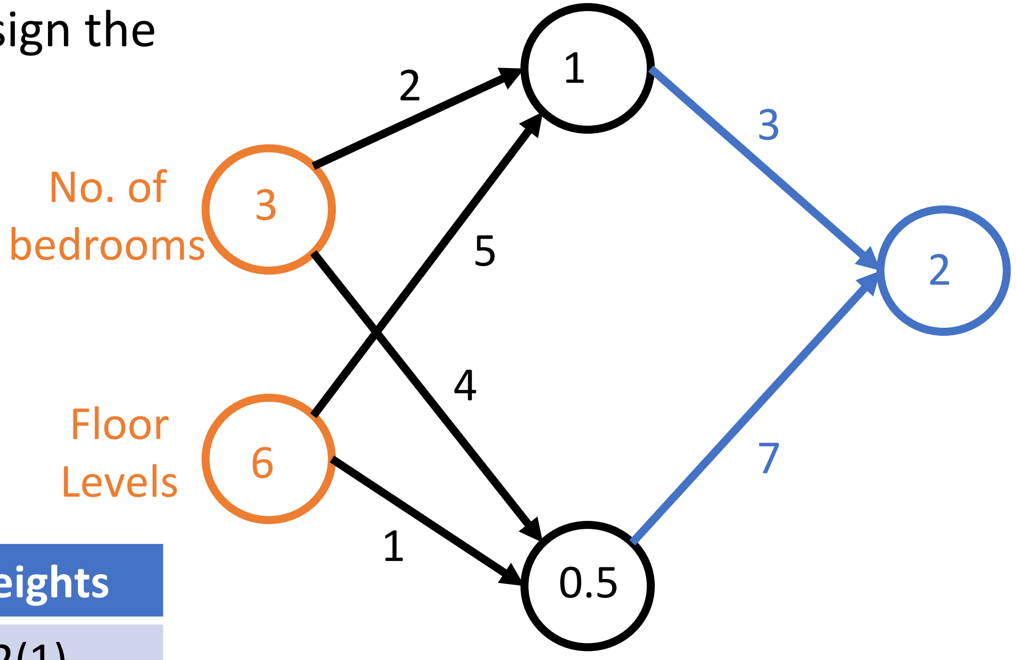
Linear Equation :
Y = m*x + b



Iteration 1

- Randomly assign the weights
- Randomly assign the bias

$Y1 = 3*2+6*5+1$
$= 37$



Output = $37*3+18.5*7+2$
$=\underline{242.5}$

$Y2 = 3*4+6*1+0.5$
$=18.5$

Input	Weights
3	2(1)
3	4(2)
6	5(1)
6	1(2)
Y1	3
Y2	7

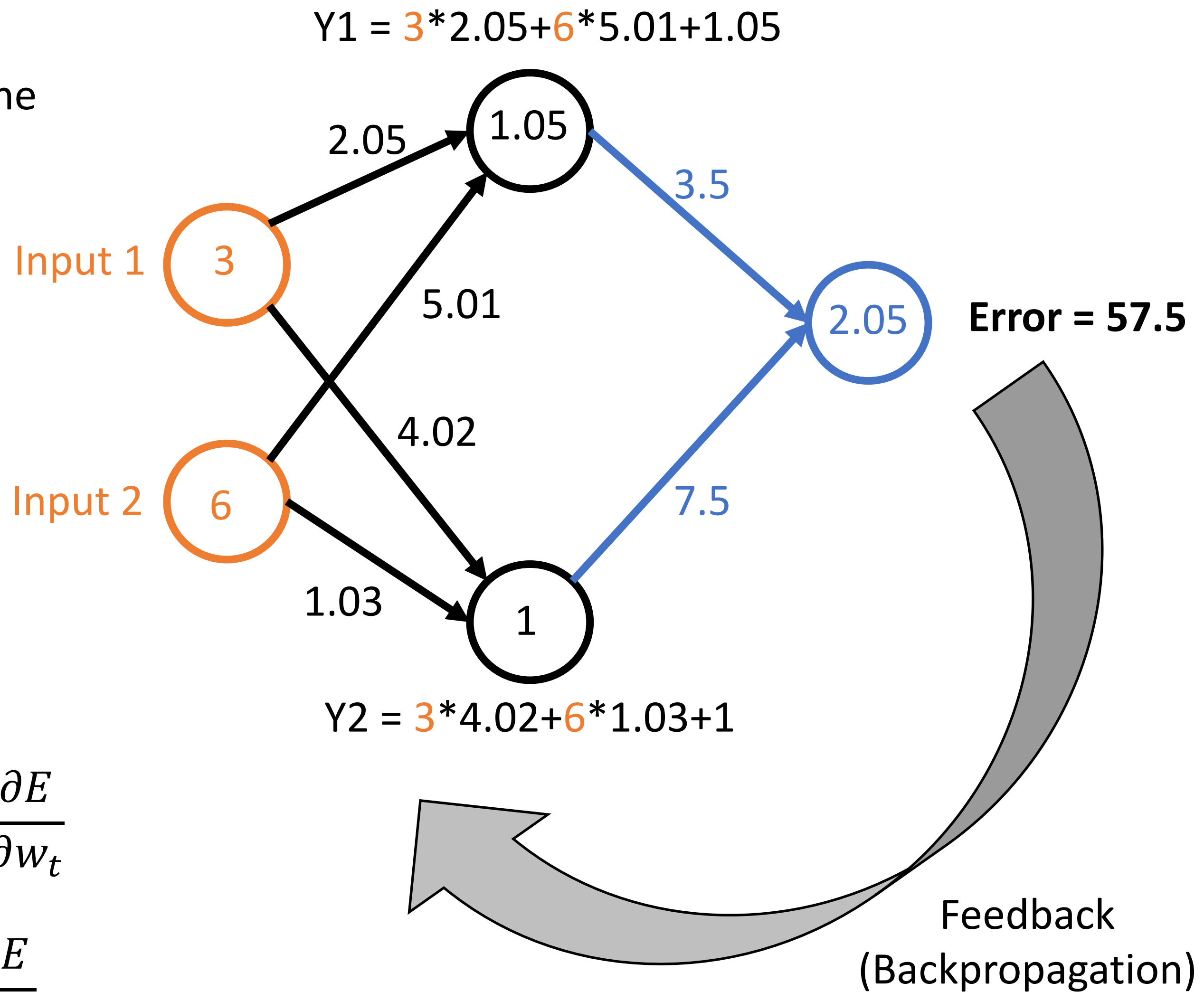
Forward Pass

Error = Actual – Predicted
$=300-242.5$
$=57.5$

Neuron	Bias
Hidden 1	1
Hidden 2	0.5
Output	2

Iteration 2

Weights and Biases updated for all the neurons based on the error



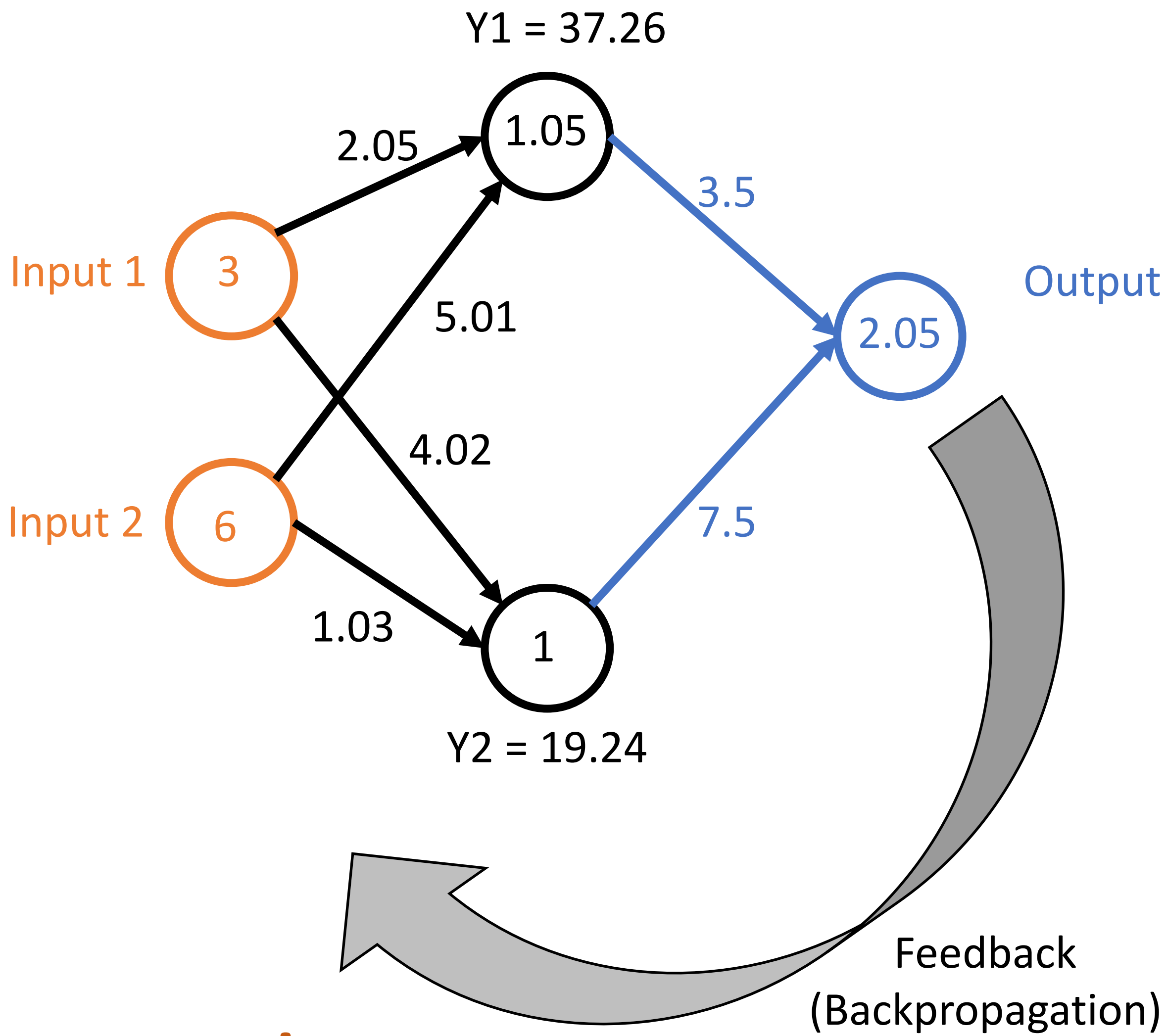
Input	Iteration 1 Weights	Iteration 2 Weights
3	2(1)	2.05(1)
3	4(2)	4.02(2)
6	5(1)	5.01(1)
6	1(2)	1.03(2)
Y1	3	3.5
Y2	7	7.5

Neuron	Iteration 1 Bias	Iteration 2 Bias
Hidden 1	1	1.05
Hidden 2	0.5	1
Output	2	2.05

$$w_{t+1} = w_t + \frac{\partial E}{\partial w_t}$$

$$b_{t+1} = b_t + \frac{\partial E}{\partial b_t}$$

Iteration 2



Output = 37.26*3.5+19.24*7.5+2.05
=276.76
The updated output

Error = 300 – 276.76
=23.24

And the process continues …

Considering the whole dataset

We considered only one data point for simplicity

No. of Bedrooms	Floor level	Price (in 1000 of $)
3	6	300
4	7	400
2	8	350
5	4	375

Considering the whole dataset

Actually the whole dataset will be considered for the iterations

No. of Bedrooms	Floor level	Price (in 1000 of $)
3	6	300
4	7	400
2	8	350
5	4	375

$$Error = \sum_{i=1}^{n}(Actual \ - Predicted)$$

Where *n = number of rows of data*

Image Classification



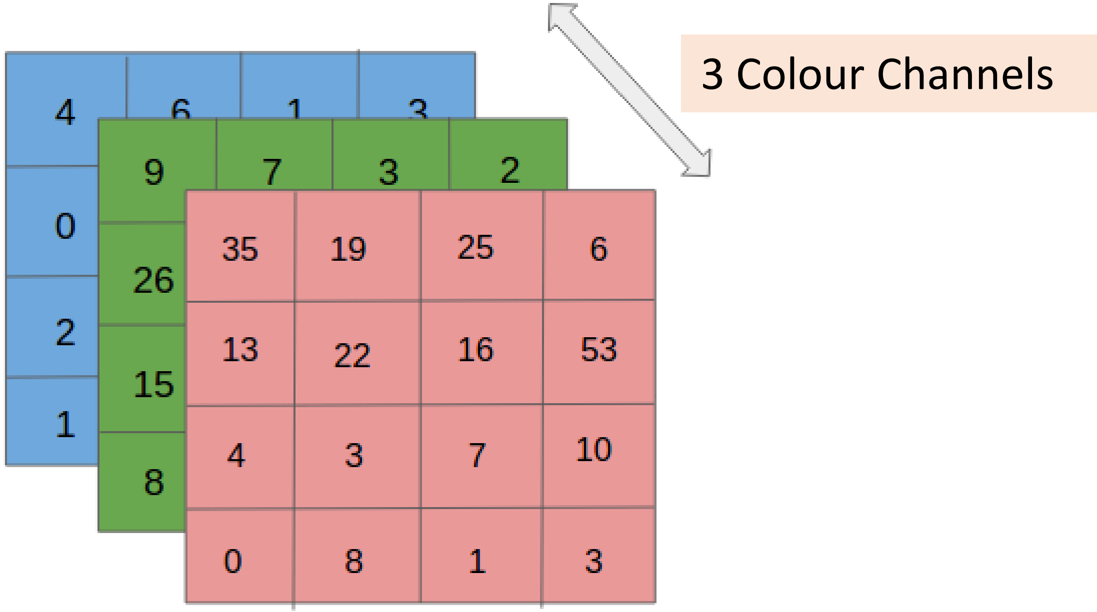

Labels
cat
dog
cat
dog




Labels
cat
cat
dog
dog

Processing the Data and how the Computer reads it

Original Image



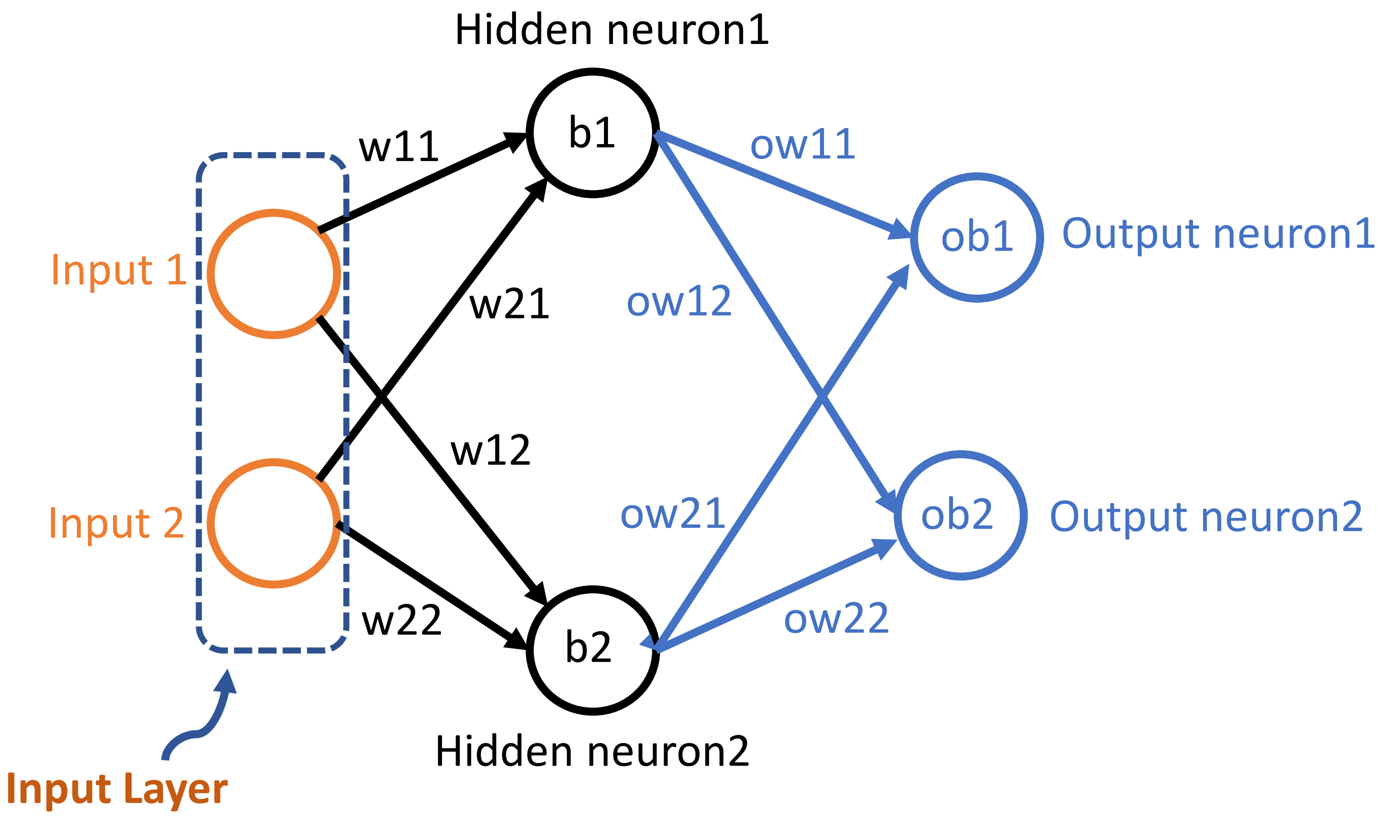
Converted to numbers for the computer algorithm to understand

5	6	42
7	10	55

5	6	42	7	10	55

Flattening the matrix

Classification Model



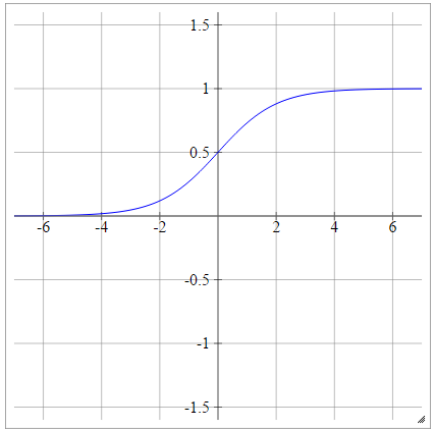
Hidden Layer
Activation = Sigmoid

Output Layer
Activation = Sigmoid

Sigmoid/Logistic Function :

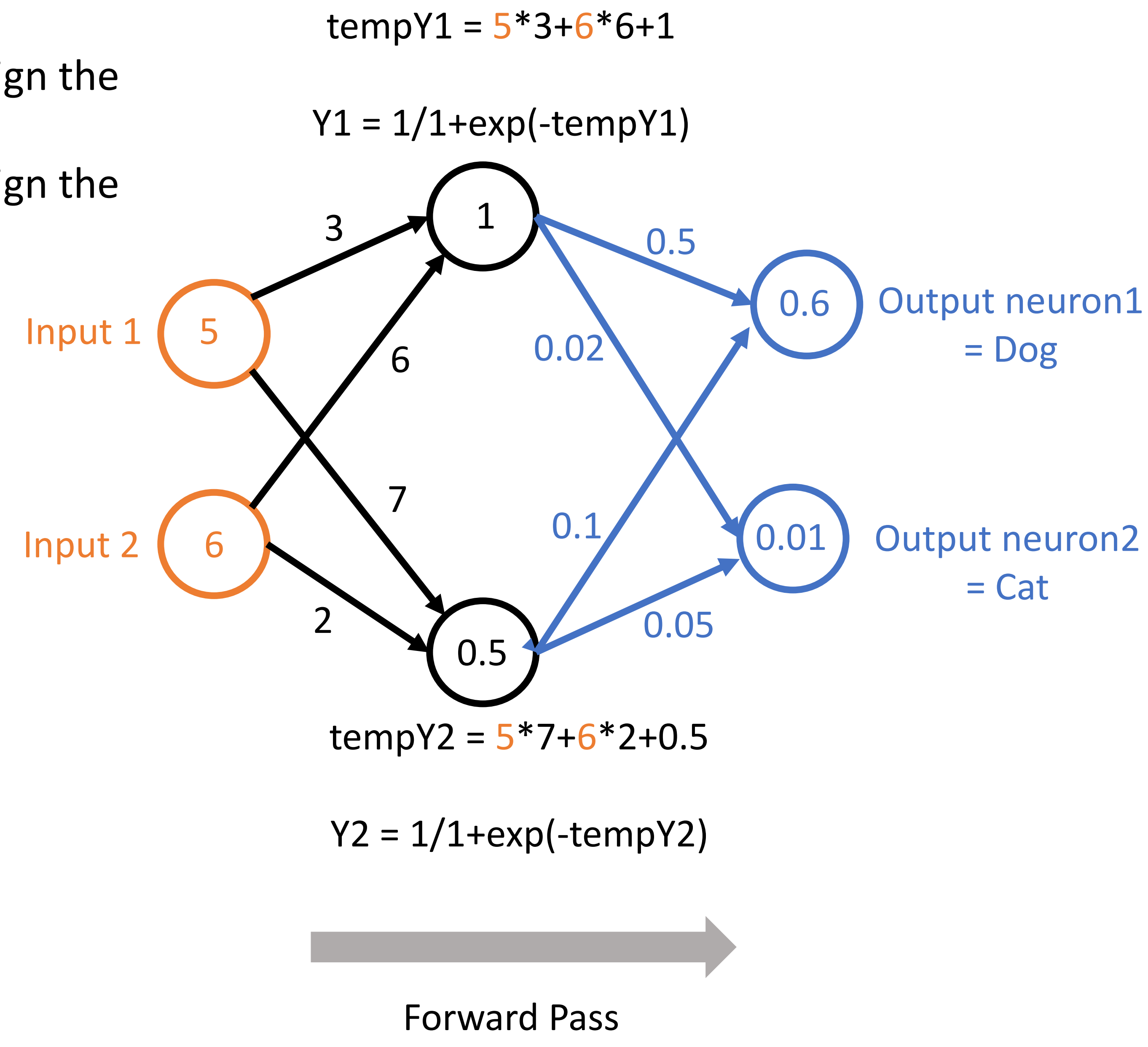
Y = 1 / (1 + exp(-z))

Where
z = m*x + b



Iteration 1

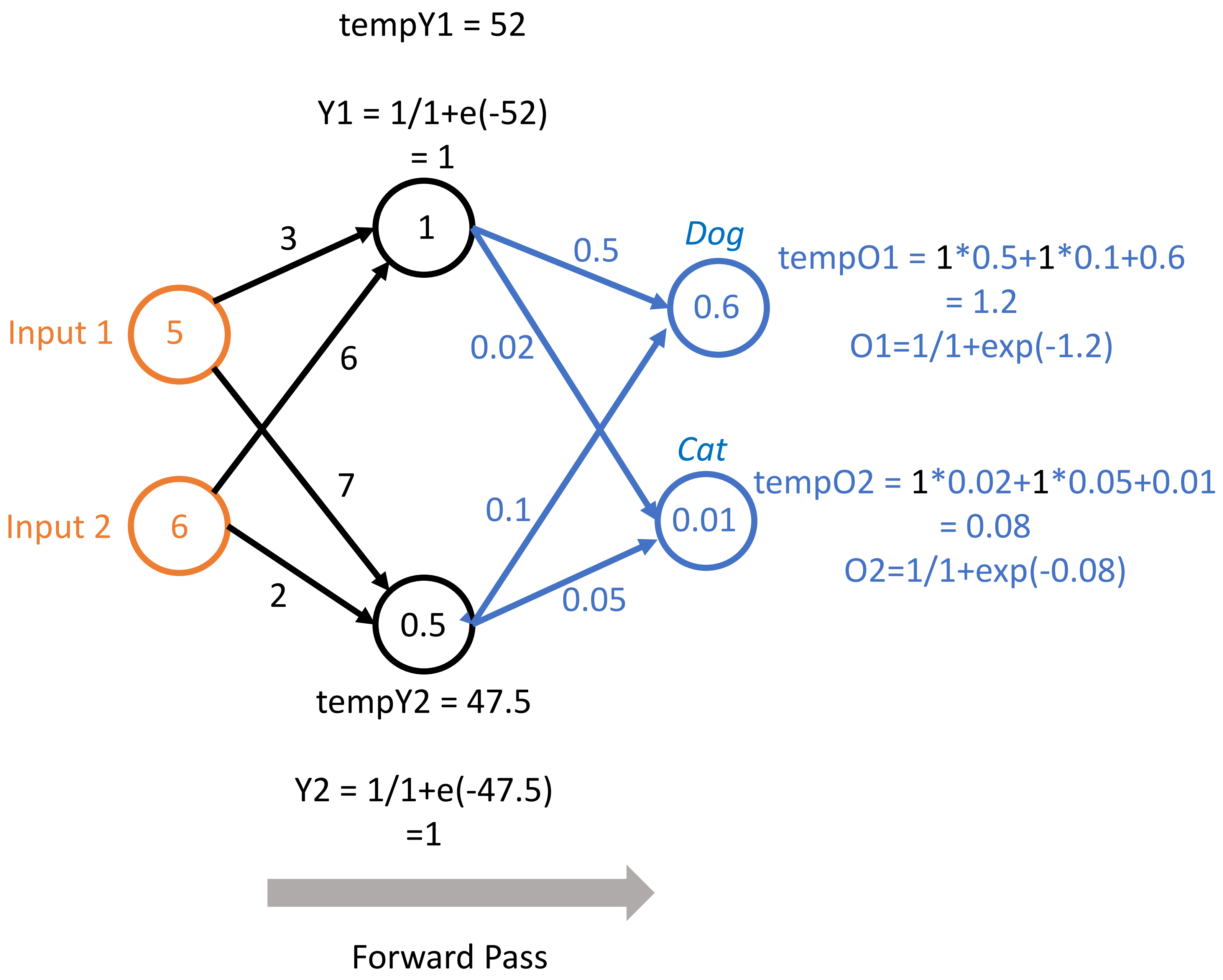
- Randomly assign the weights
- Randomly assign the bias



Input	Weights
2	3(1)
2	7(2)
5	6(1)
5	2(2)
Y1	0.5(1)
Y1	0.02(2)
Y2	0.1(1)
Y2	0.05(2)

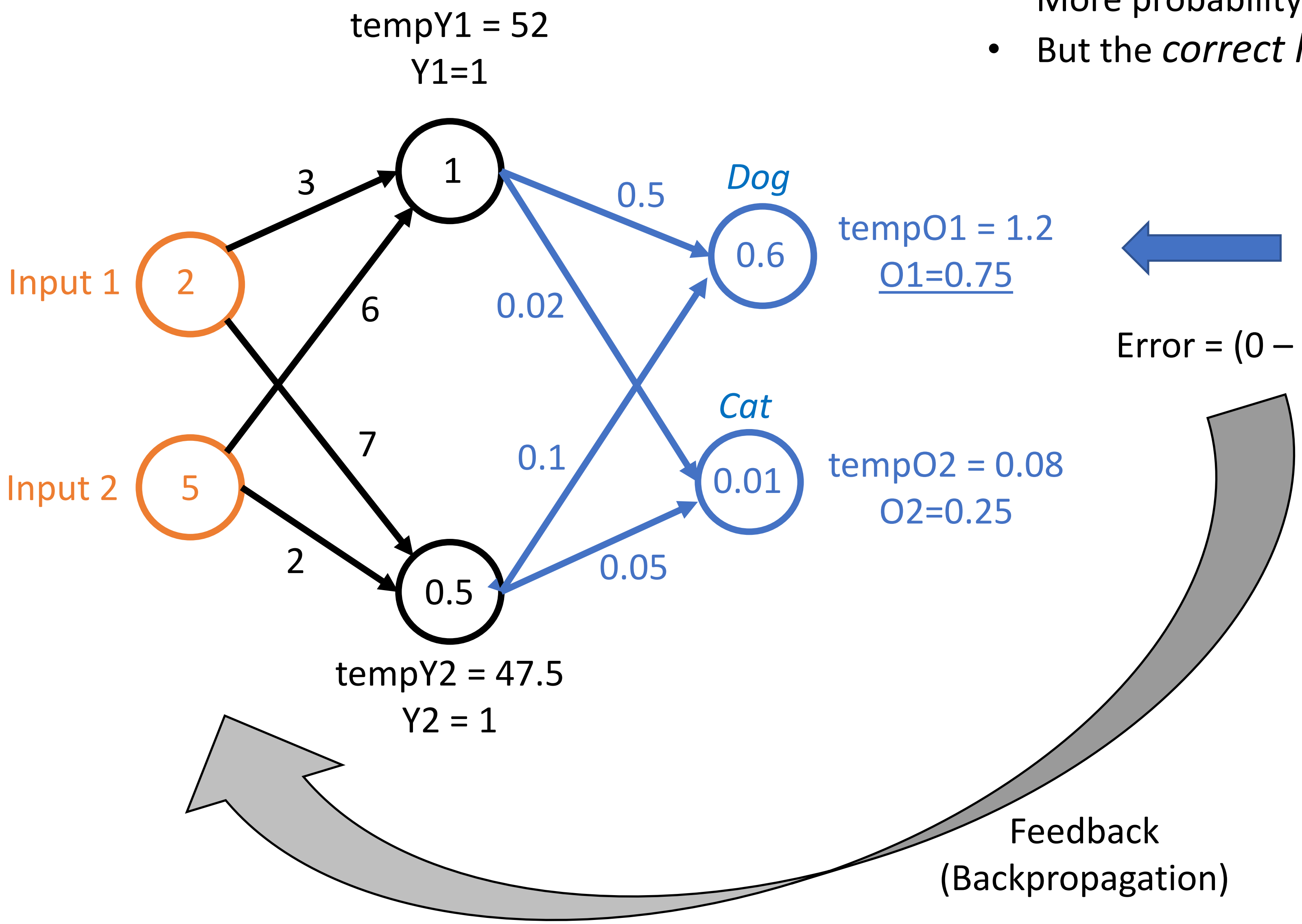
Neuron	Bias
Hidden 1	1
Hidden 2	0.5
Output 1	0.6
Output 2	0.01

Iteration 1


tempY1 = 52
Y1 = 1/1+e(-52)
= 1
Input 1
5
Input 2
6
3
6
7
2
1
0.5
0.5
0.02
0.1
0.05
Dog
0.6
Cat
0.01
tempO1 = 1*0.5+1*0.1+0.6
= 1.2
O1=1/1+exp(-1.2)
tempO2 = 1*0.02+1*0.05+0.01
= 0.08
O2=1/1+exp(-0.08)
tempY2 = 47.5
Y2 = 1/1+e(-47.5)
=1
Forward Pass

Iteration 1

- More Probability to Output 1 = More probability to Dog
- But the *correct label was Cat*



Error = (0 – 0.75)^2 + (1-0.25)^2

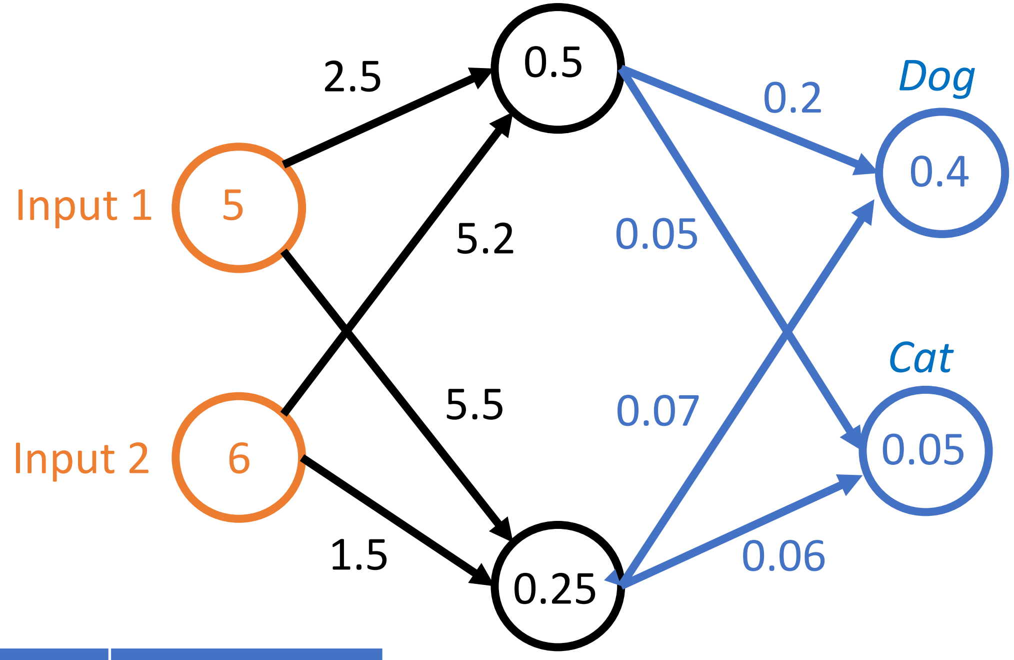
= 1.14

$$w_{t+1} = w_t + \frac{\partial E}{\partial w_t}$$

$$b_{t+1} = b_t + \frac{\partial E}{\partial b_t}$$

Iteration 2

Weights and Biases change according to the error



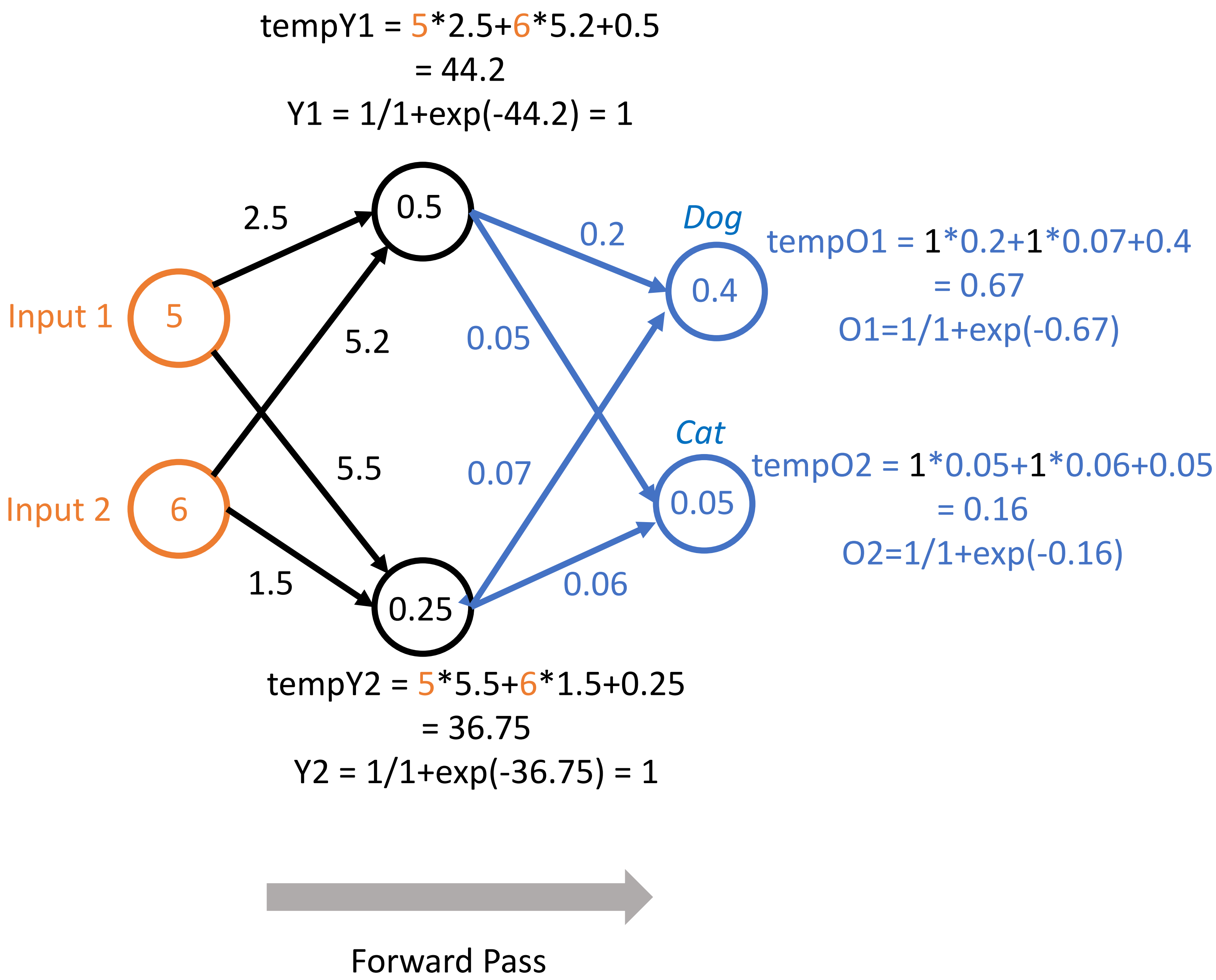
Input	Iteration 1 Weights	Iteration 2 Weights
2	3(1)	2.5(1)
2	7(2)	5.5(2)
5	6(1)	5.2(1)
5	2(2)	1.5(2)
Y1	0.5(1)	0.2(1)
Y1	0.02(2)	0.05(2)
Y2	0.1(1)	0.07(1)
Y2	0.05(2)	0.06(2)

Iteration 1
Error = 1.14

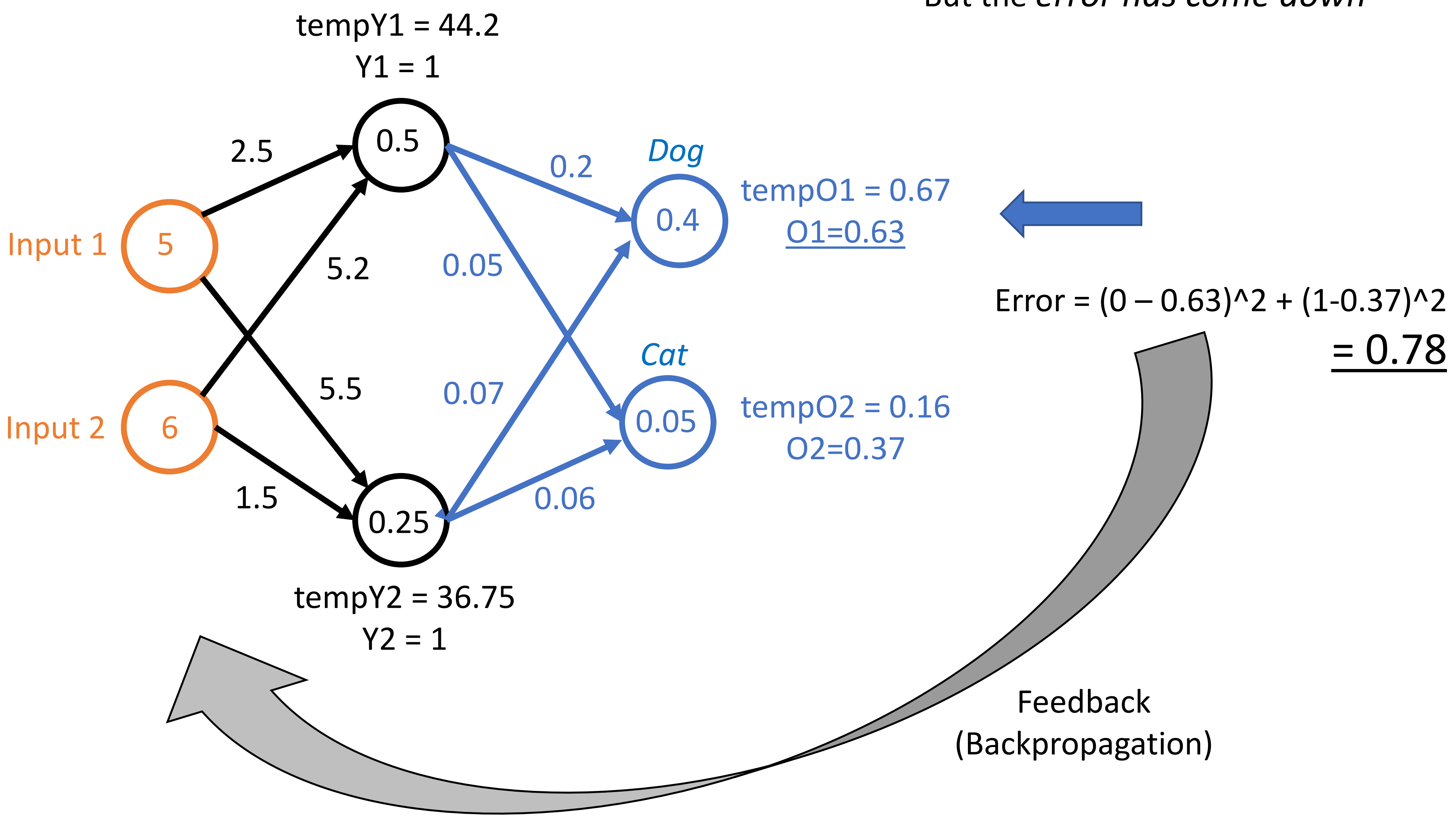
Feedback
(Backpropagation)

Neuron	Iteration 1 Bias	Iteration 2 Bias
Hidden 1	1	0.5
Hidden 2	0.5	0.25
Output 1	0.6	0.4
Output 2	0.01	0.05

Iteration 2

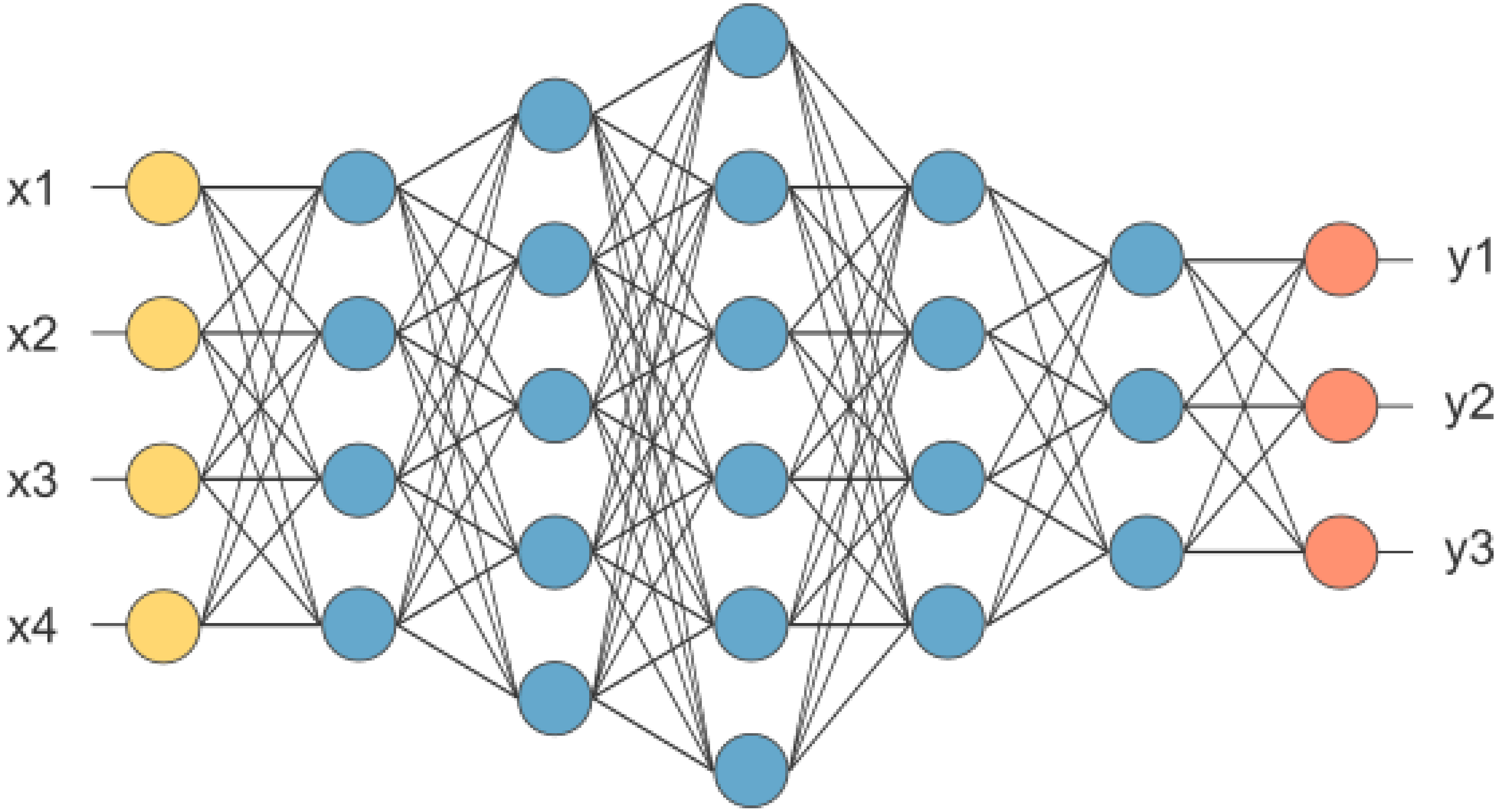

tempY1 = 5*2.5+6*5.2+0.5
= 44.2
Y1 = 1/1+exp(-44.2) = 1
2.5
0.5
5.2
5.5
1.5
0.25
Input 1
5
Input 2
6
0.2
0.05
0.07
0.06
Dog
0.4
Cat
0.05
tempO1 = 1*0.2+1*0.07+0.4
= 0.67
O1=1/1+exp(-0.67)
tempO2 = 1*0.05+1*0.06+0.05
= 0.16
O2=1/1+exp(-0.16)
tempY2 = 5*5.5+6*1.5+0.25
= 36.75
Y2 = 1/1+exp(-36.75) = 1
Forward Pass

- Still more probability to Dog
- But the *error has come down*



And the process continues …

A DEEP Neural Network a.k.a. DEEP LEARNING



Feature Representation by Deep Nets

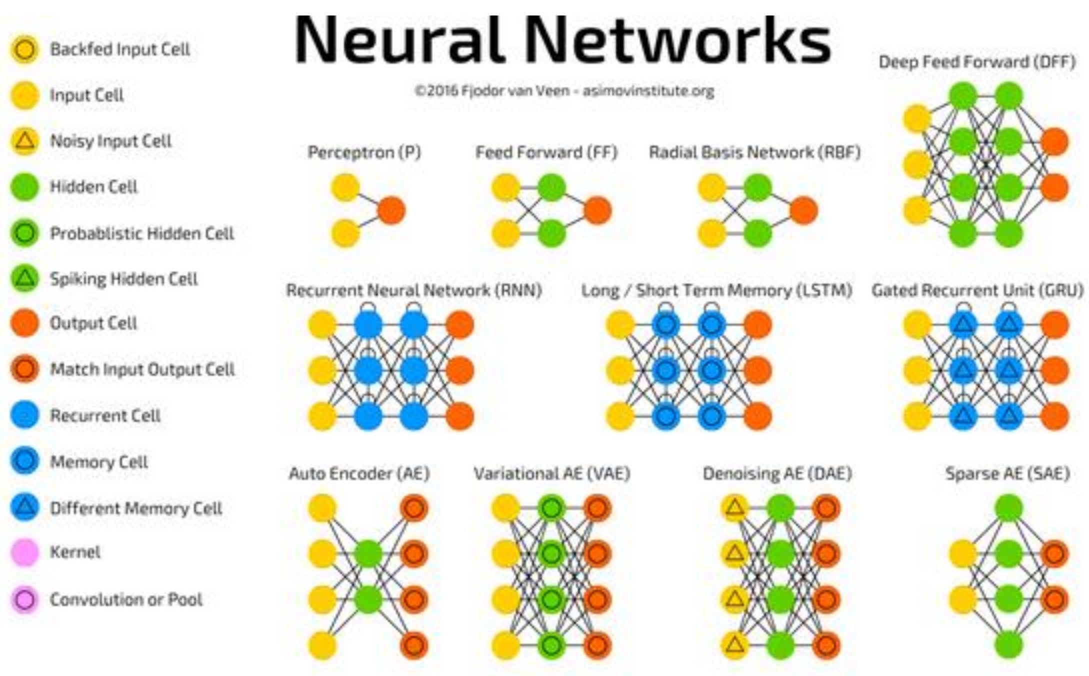
Simple to Complex representation by the different layers

First Layer Representation   Second Layer Representation   Third Layer Representation

What made Deep Learning possible ...

Lots of data

Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs)



Lots of new and better algorithms

The Limitations of Deep Learning

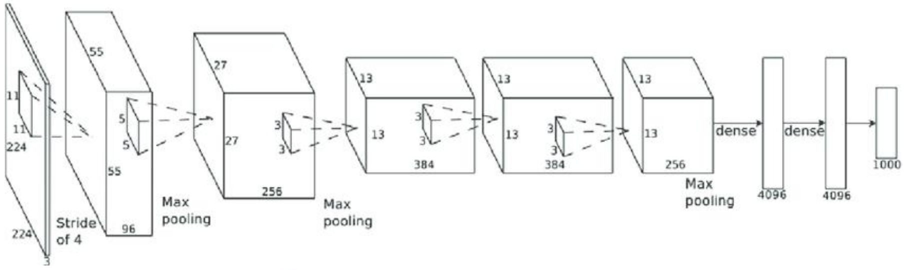
- Need for **large, well-labelled** training datasets (for supervised learning)

- For training requires **lots of time and/or money**
  GPUs *(Graphics Processing Unit) provide significant increase in computing power however are not cheap (thousands of $)*

- The **architecture** of the deep net needs to be optimized for each use case

Transfer Learning

Randomly initialized DeepNet



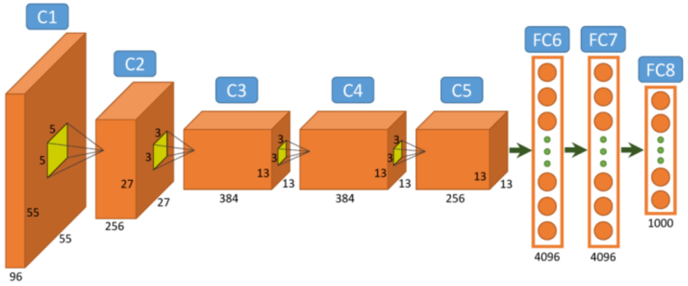
Our data

+

Fine Tuning

Target Labels relevant to our data

This is what we work with



Trained with imagenet

The platform for Deep Learning - TensorFlow

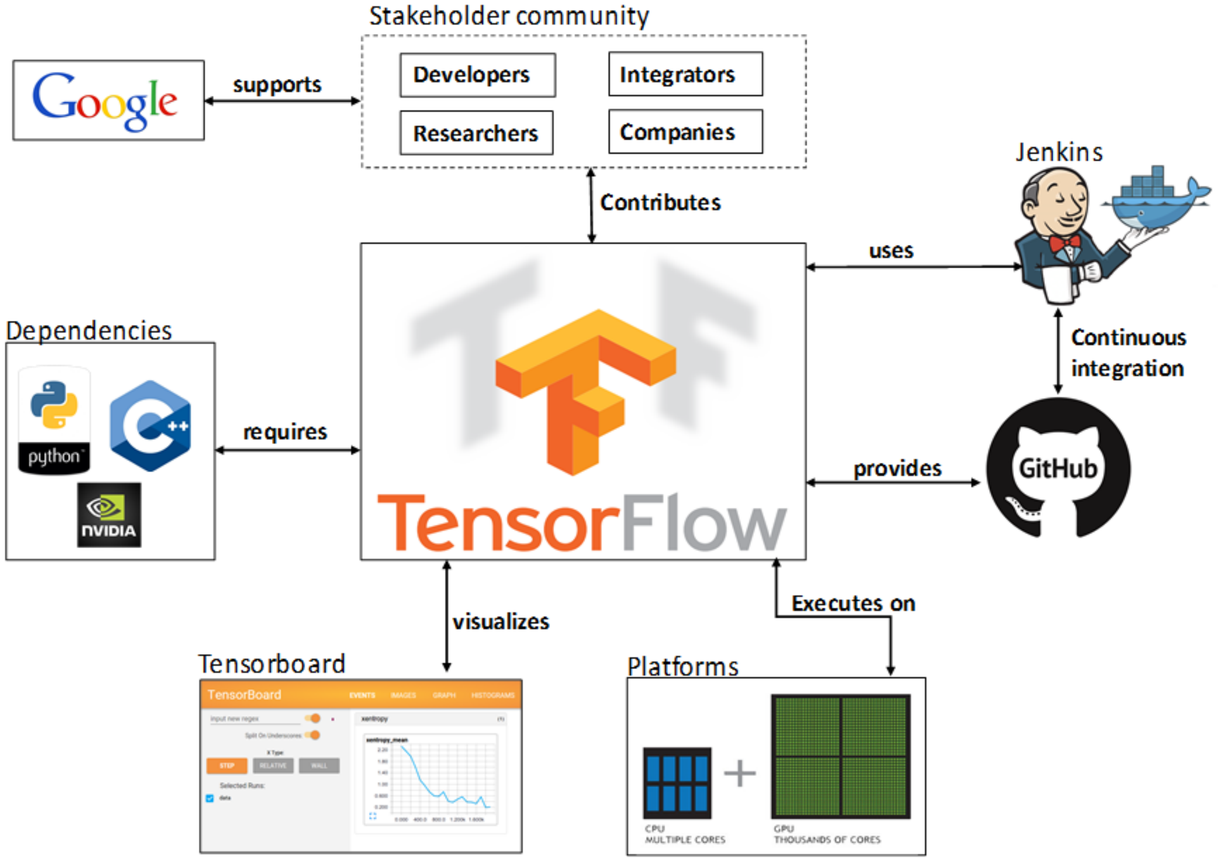
What is Tensor Flow ?

TensorFlow™

Open Source Python artificial intelligence library using data flow graphs to build models

Helps to create a deep neural network architecture

Used in language understanding, image recognition, classification and prediction .



Lets get our hands dirty…