McGill initiative in Computational Medicine

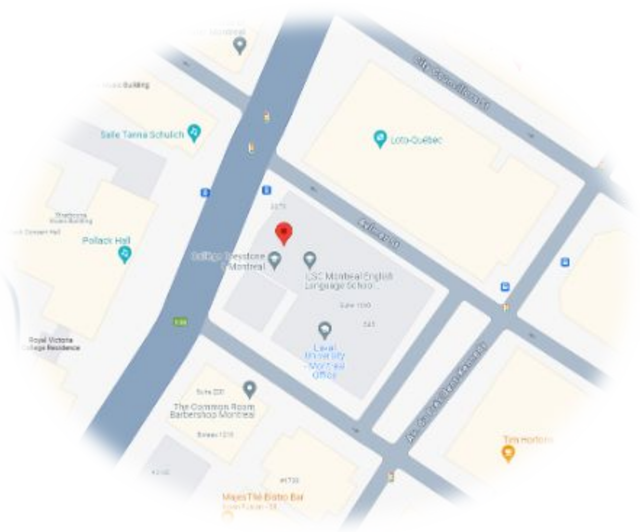
Fundamentals of Machine Learning

Multi Layer Perceptrons

Workshop Lead: Tugce Gurbuz

Jul 24th, 2024

<u>Mission statement:</u> deliver quality workshops designed to help biomedical researchers develop the skills they need to succeed.



Location: 550 Sherbrooke Street, Montreal, Quebec

Scan the QR code to sign up for our **mailing list**

Contact: workshop-micm@mcgill.ca

McGill initiative in
Computational Medicine

McGill initiative in Computational Medicine

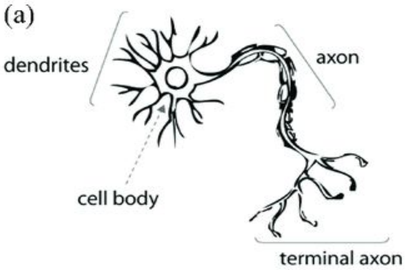
Summer 2024 Workshop Series

Workshop	Date	Lead/Facilitator	Location	Registration
How to think in Code	July 3 10AM-1PM	Thomas Zheng	Education Room 133	Open
Intro to UNIX and HPC	July 11 9AM-1pm	Georgi Mehri	Education Room 133	Open
Intro to Git & GitHub	July 12 1PM-5PM	Adrien Osakwe	Education Room 133	Open
Intro to Python (Part 1)	July 16 9AM-1PM	Benjamin Rudski	Education Room 133	Open
Intermediate Python (Part 2)	July 18 9AM-1PM	Benjamin Rudski	Education Room 133	Open
Fundamentals of Machine Learning	July 24 9AM-1PM	Tugce Gurbuz	Education Room 133	Open
Intro to Matlab	August 7 9AM-1PM	Meghana Munipalle	Education Room 133	TBA
Intro to R (Part 1)	August 12 9AM-1PM	TBA	Education Room 133	TBA
Intermediate R (Part 2)	August 14 1PM-5PM	Gerardo Martinez	Education Room 133	TBA
Intro to Bayesian Inference in R	August 16 1PM-5PM	Adrien Osakwe	Education Room 133	TBA
Proteogenomics	August 19 1PM-5PM	Thomas Zheng	Education Room 133	TBA

https://www.mcgill.ca/micm/training/workshops-series

Perceptron (1958)

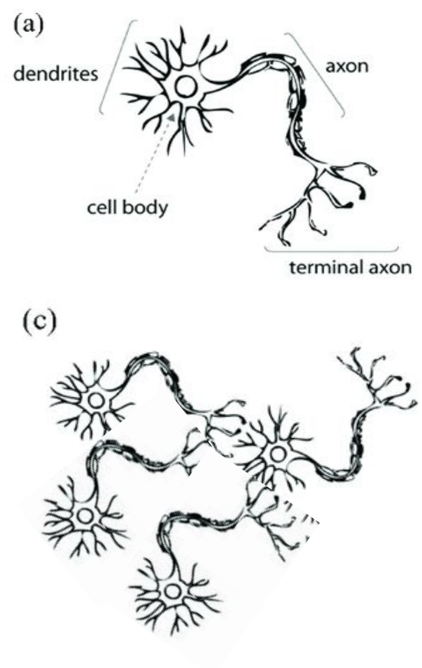
Source



1: fire
0: not fire

Perceptron (1958)

Source

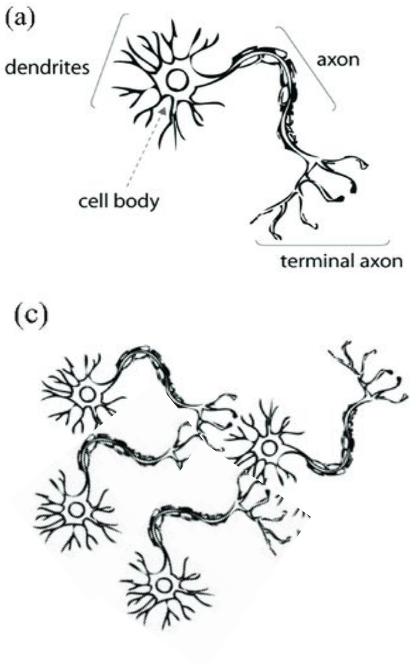


1: fire
0: not fire

Excitatory and inhibitory connections

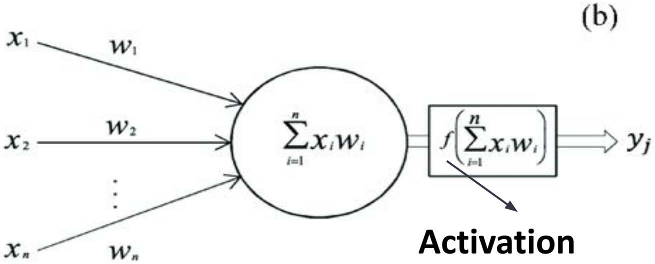
Perceptron (1958)

[Source]



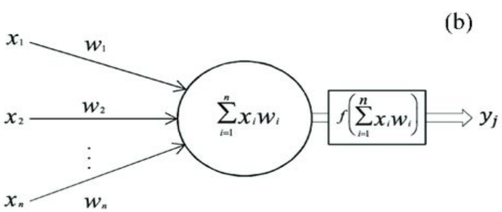
1: fire
0: not fire

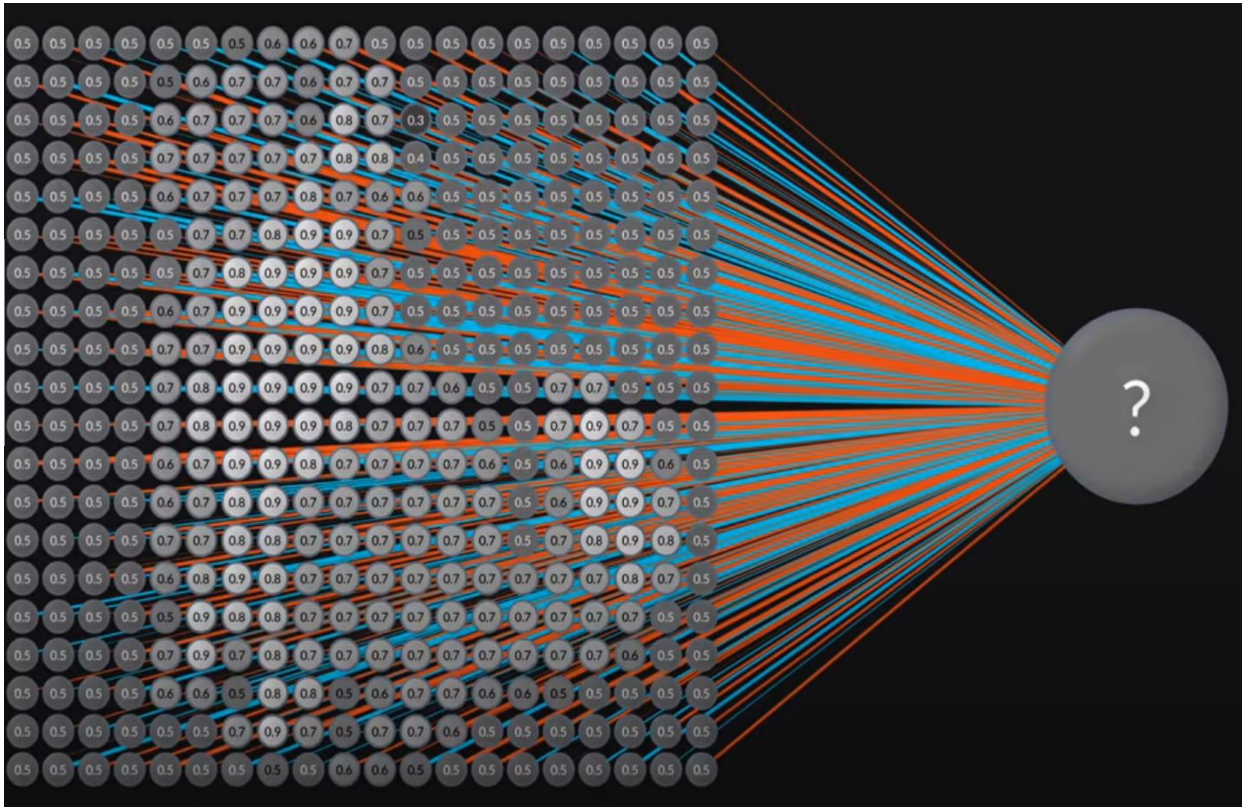
Excitatory and inhibitory connections



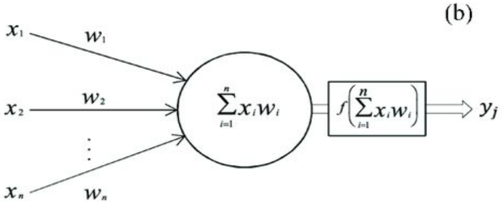
Activation function

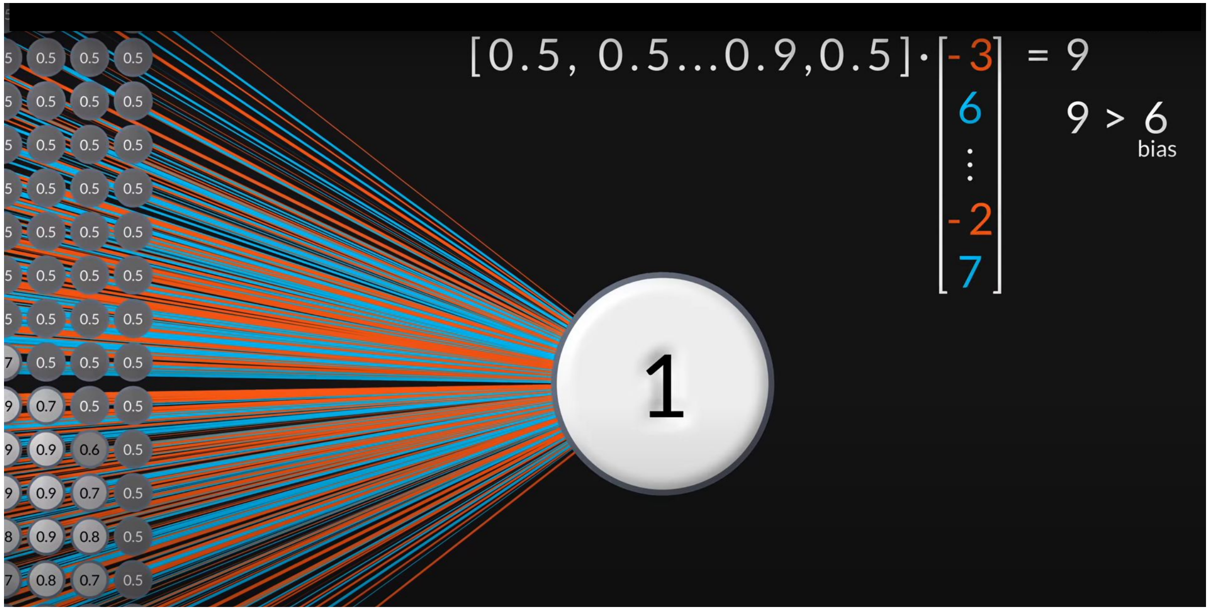
How does perceptron distinguish between 2 images?



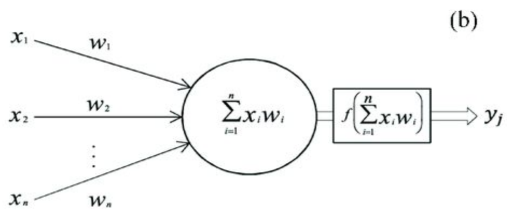


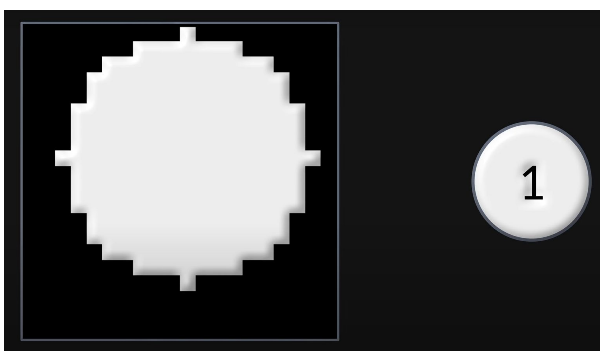
How does perceptron distinguish between 2 images?





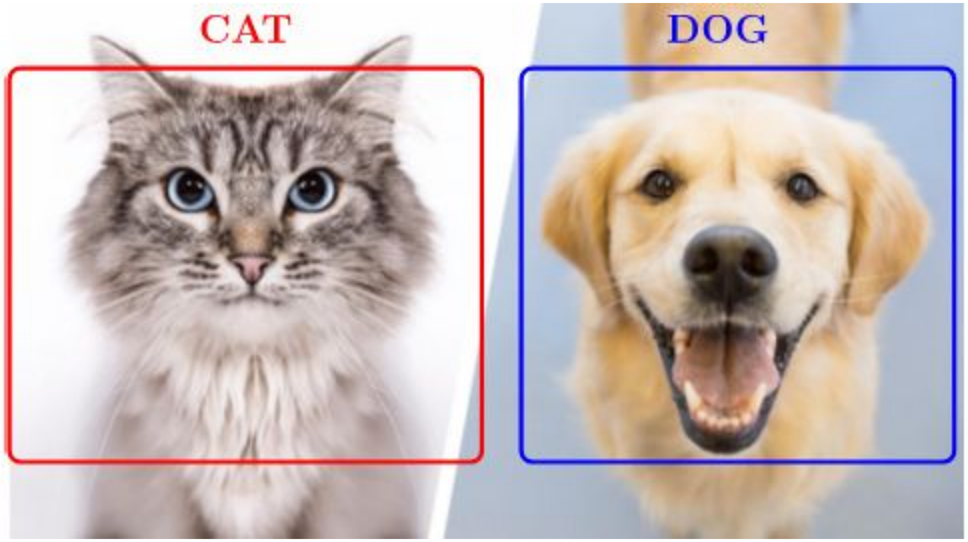
How does perceptron distinguish between 2 images?



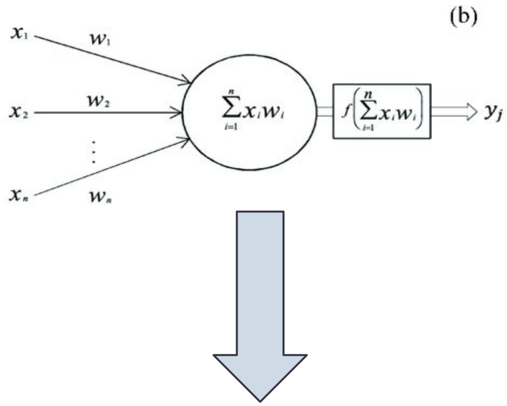


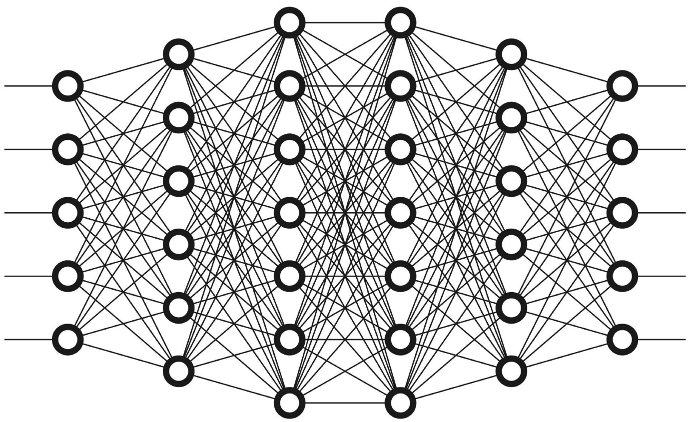
McGill initiative in
Computational Medicine

How does perceptron distinguish between 2 images?

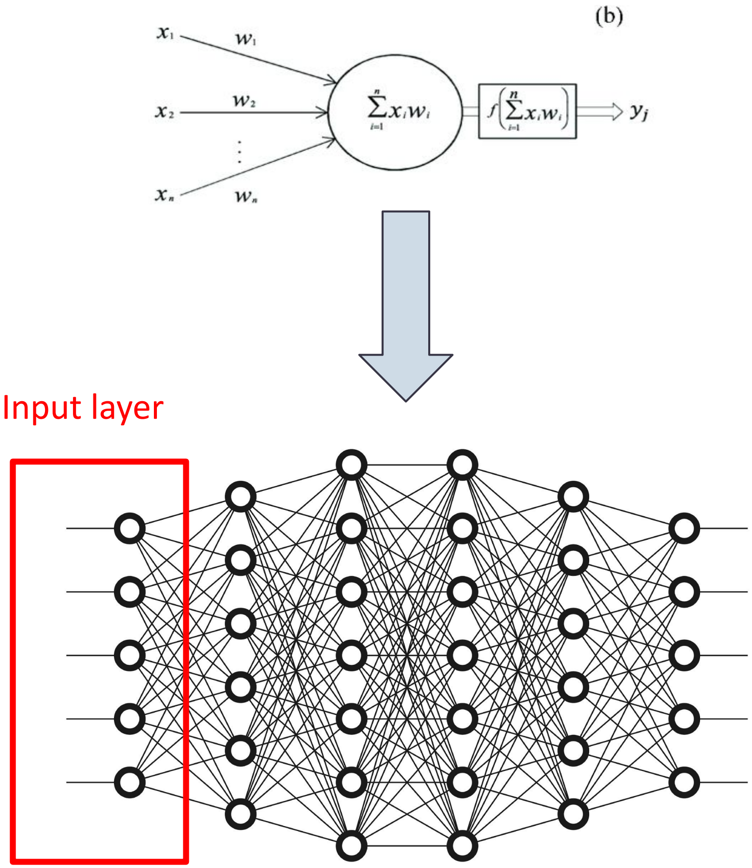


Multi Layer Perceptron



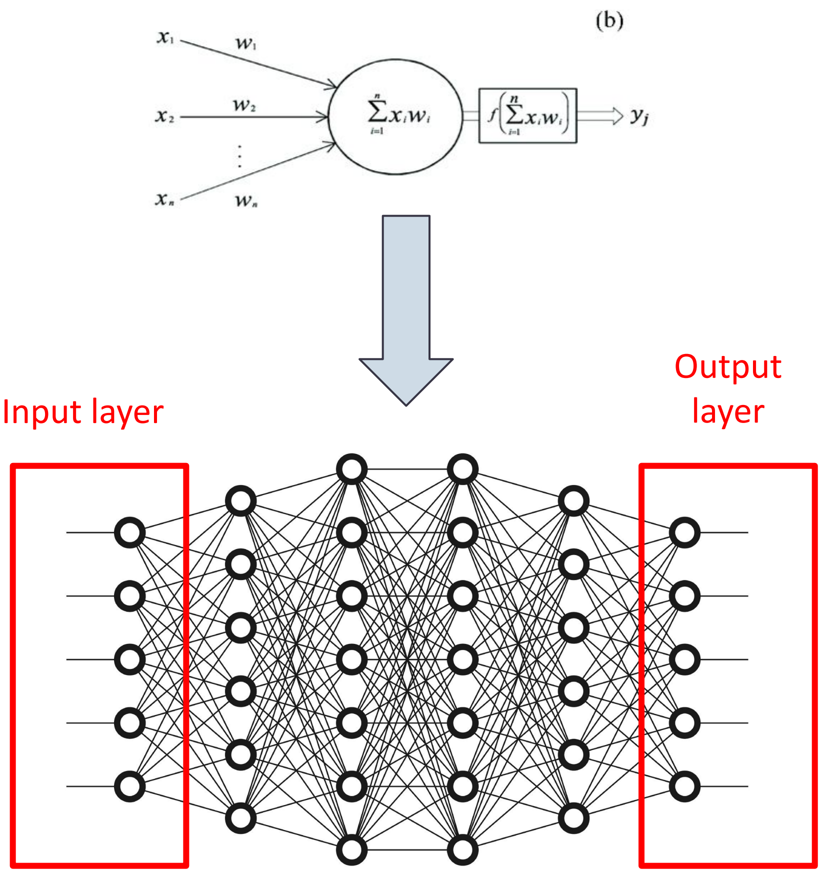


Multi Layer Perceptron

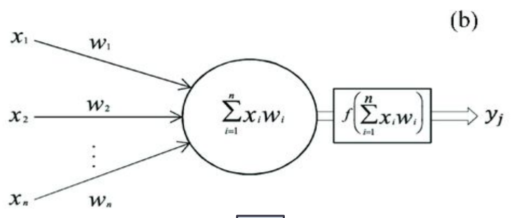


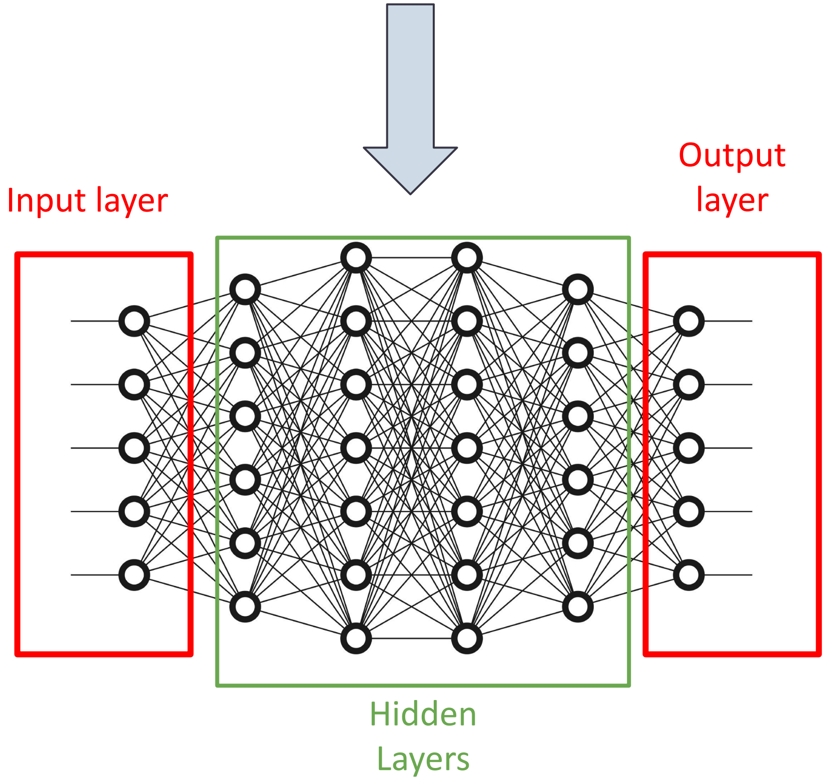
McGill initiative in
Computational Medicine

Multi Layer Perceptron

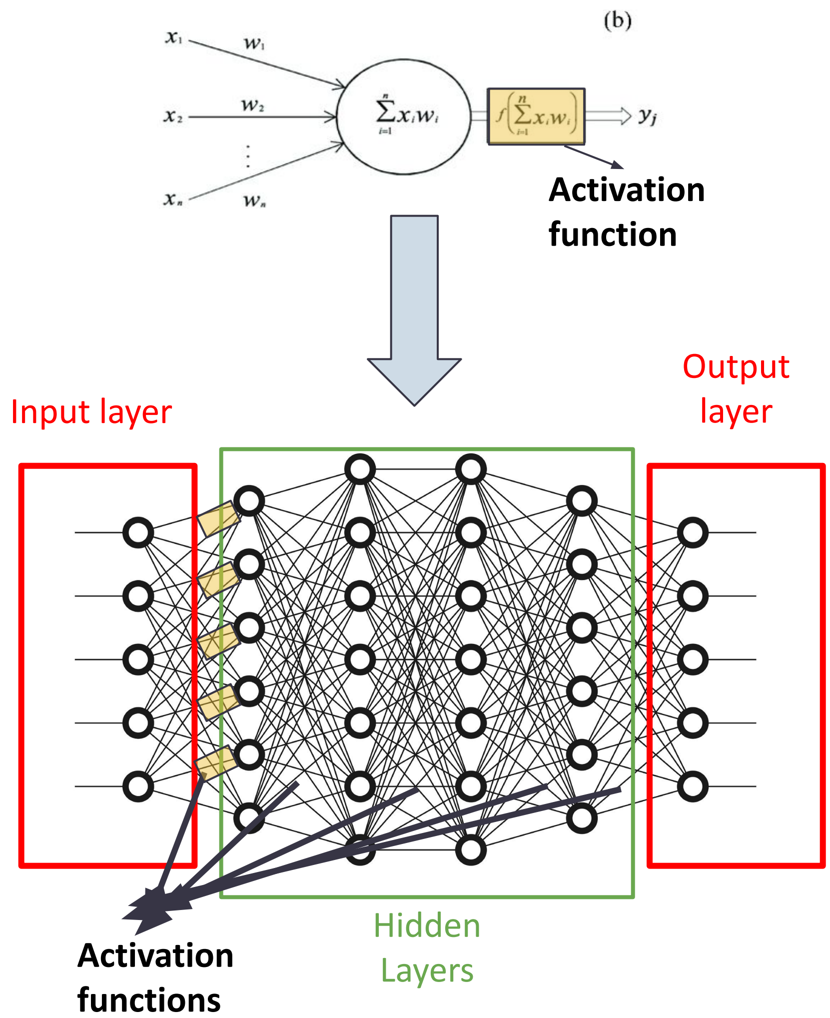


Multi Layer Perceptron





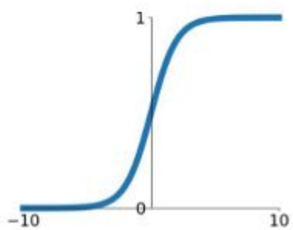
Multi Layer Perceptron



Activation Functions

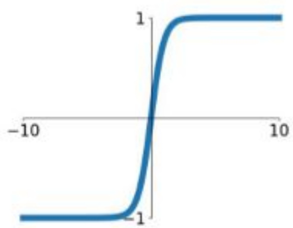
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



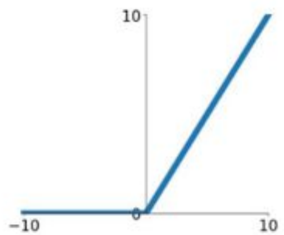
tanh

$\tanh(x)$



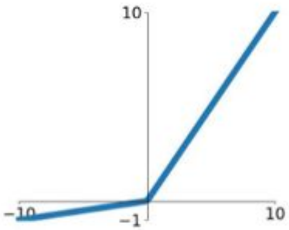
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

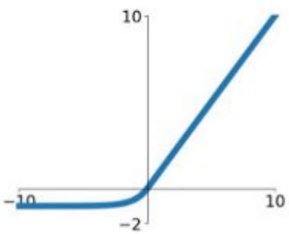


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

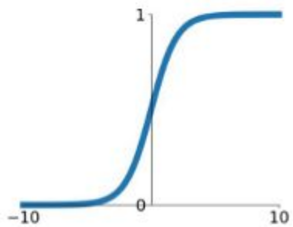
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



Activation Functions

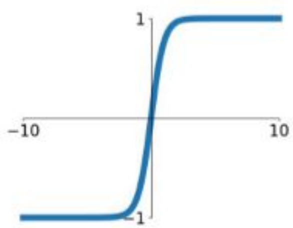
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



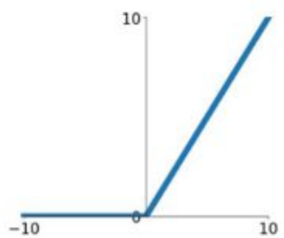
tanh

$\tanh(x)$



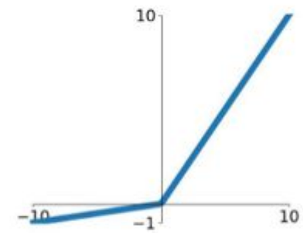
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

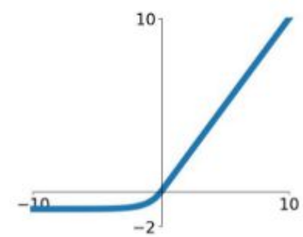


Maxout

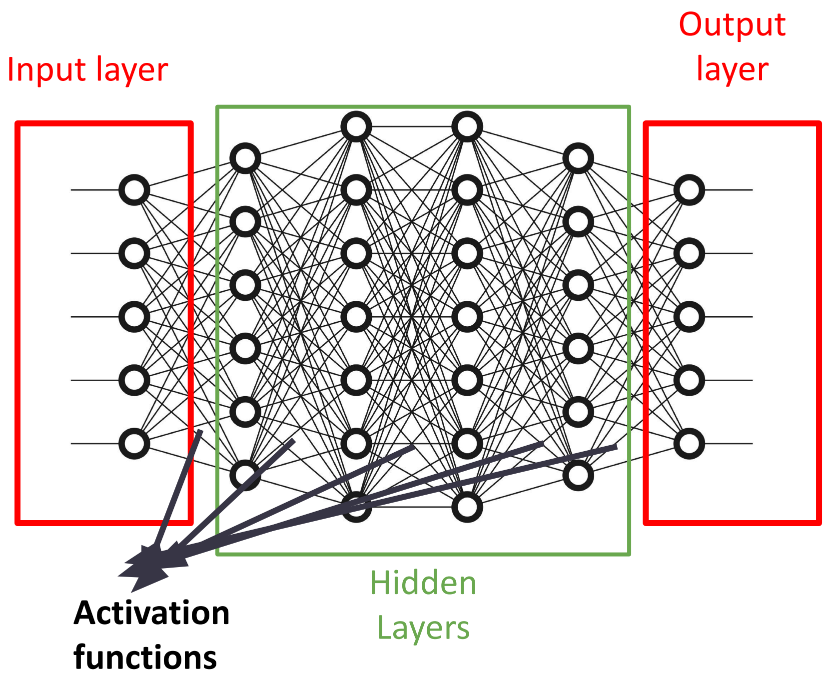
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Let’s implement our first ML model!

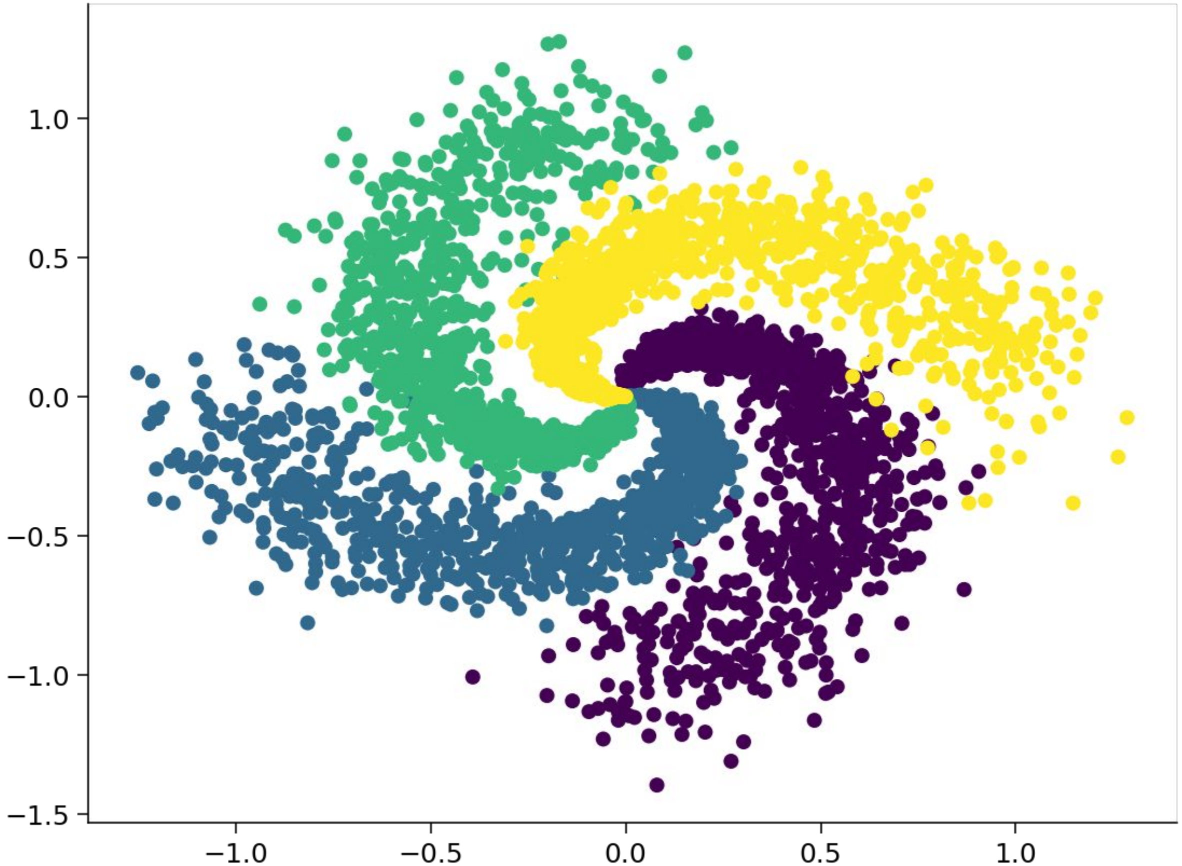


Let’s go to our first tutorial and implement our first ML model!!

How do we teach MLP to classify?

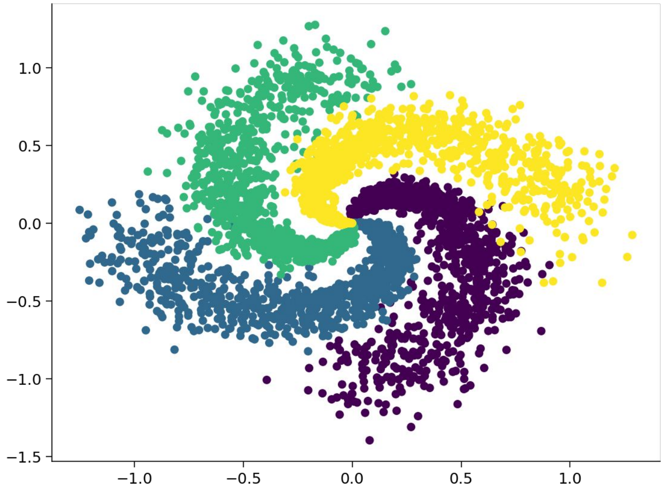
How do we teach MLP to classify?

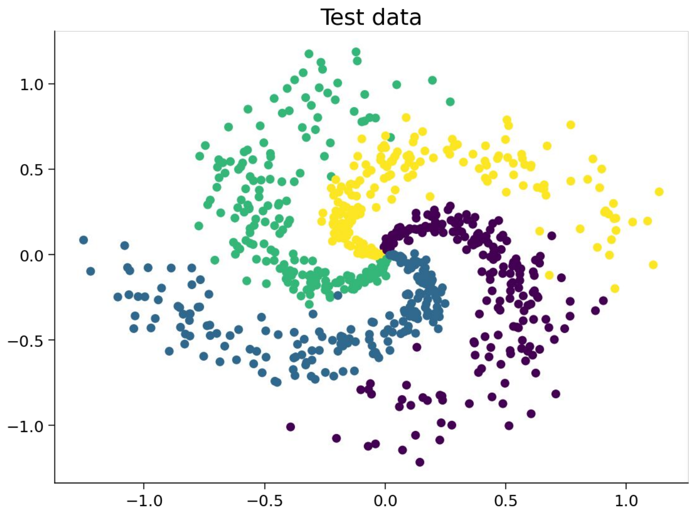
Dataset



How do we teach MLP to classify?

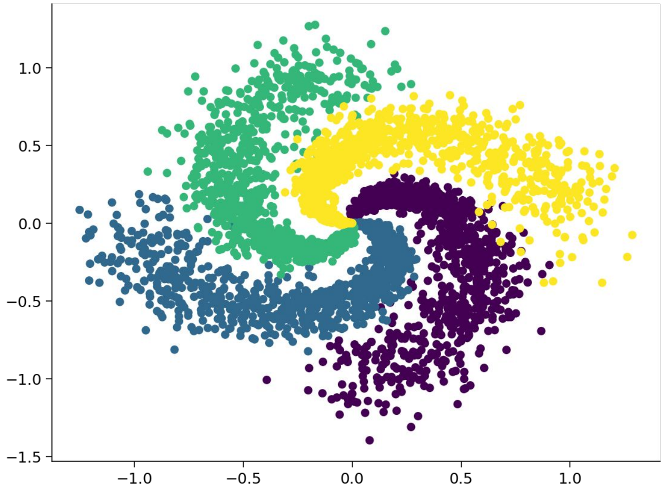
Dataset

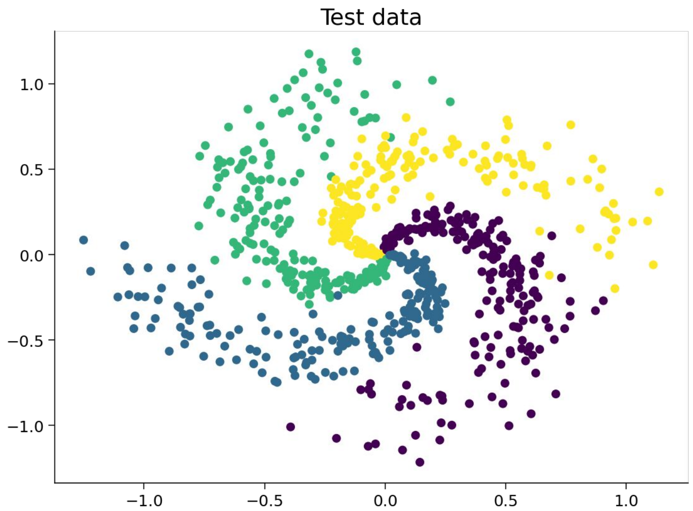




How do we teach MLP to classify?

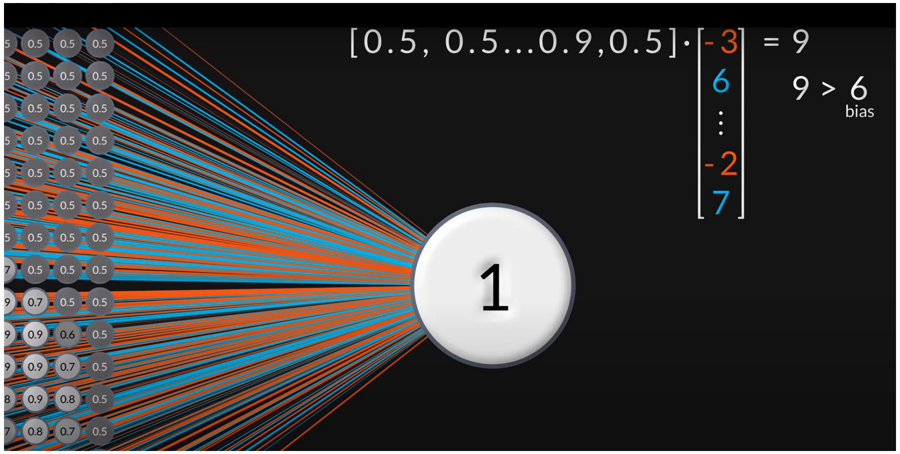
Dataset

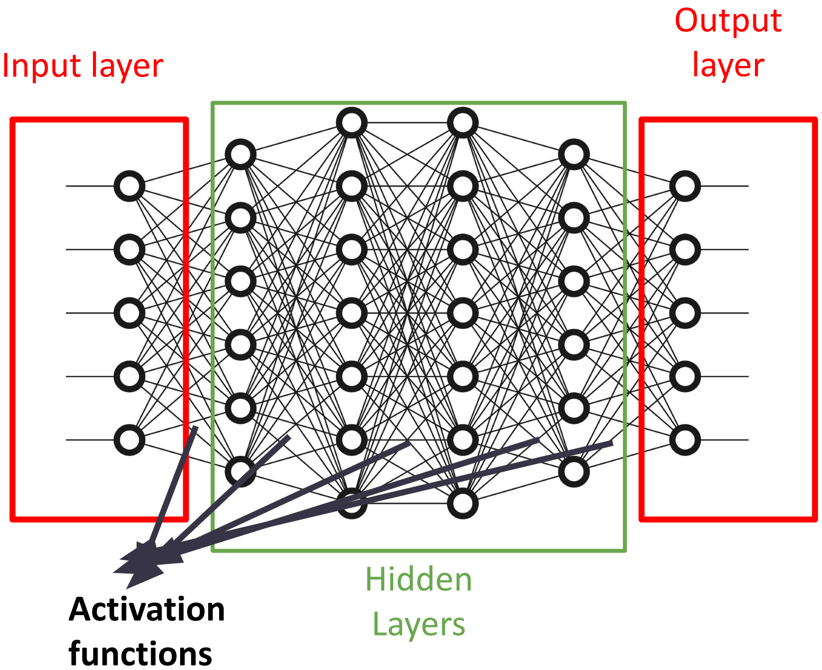




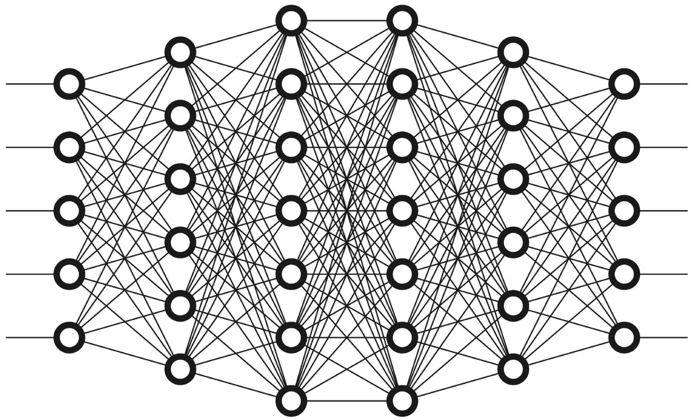
Let’s prepare data for our model!

How do we teach MLP to classify?





How do we teach MLP to classify?
Learning good weights



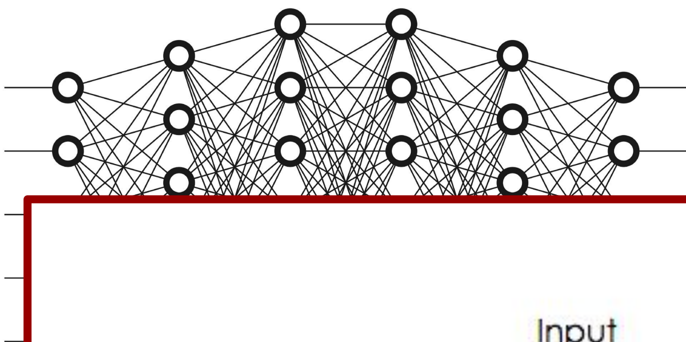
Learning Recipe:
1-Initialize weights
2-Run a forward pass and make a prediction
3-Calculate loss to evaluate how good/bad your prediction was
4-Adjust your weights in a way so that loss will be minimized

Popular weight initialization methods: Xavier/Glorot Initialization , Normalized Xavier/Glorot Initialization, He Weight Initialization, Random Weight Initialization

Appendix: Andrew Ng's Lecture on Weight Initialization

How do we teach MLP to classify?

Learning good weights

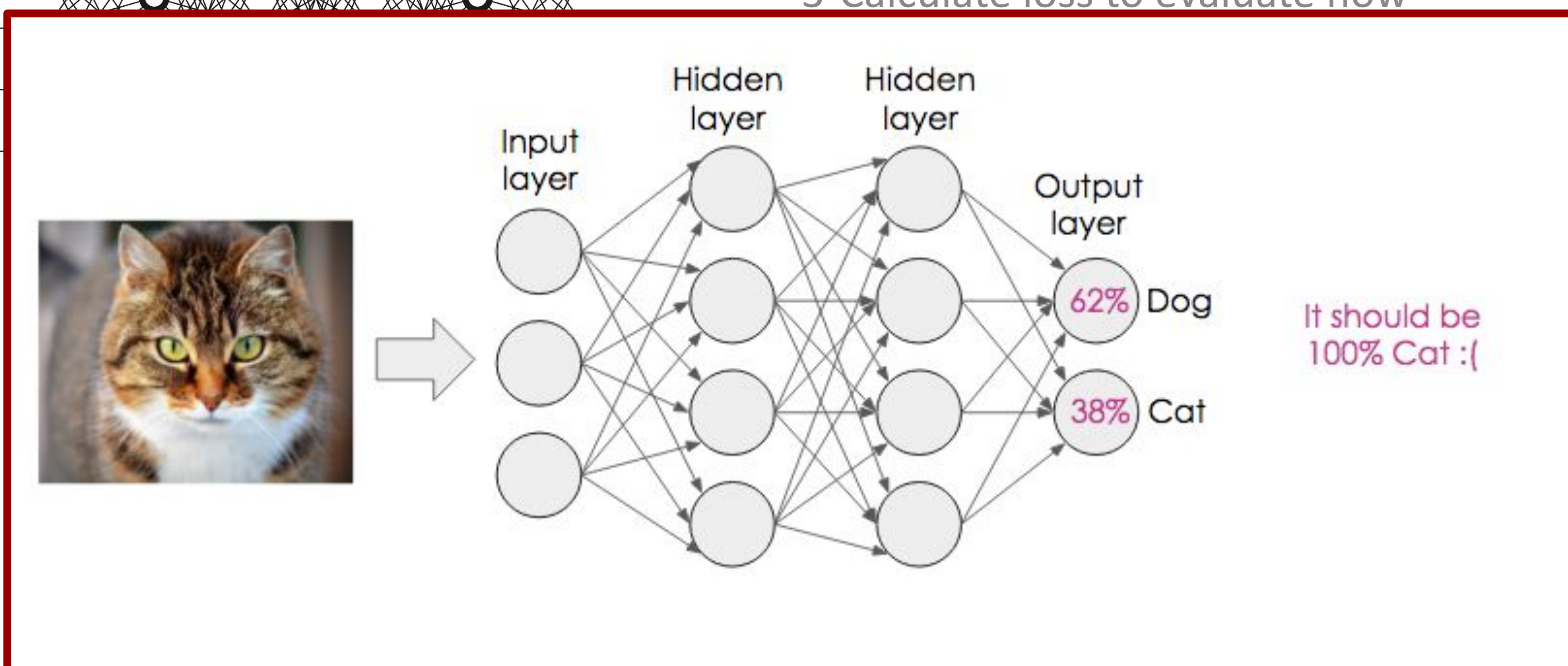


Learning Recipe:

1-Initialize weights

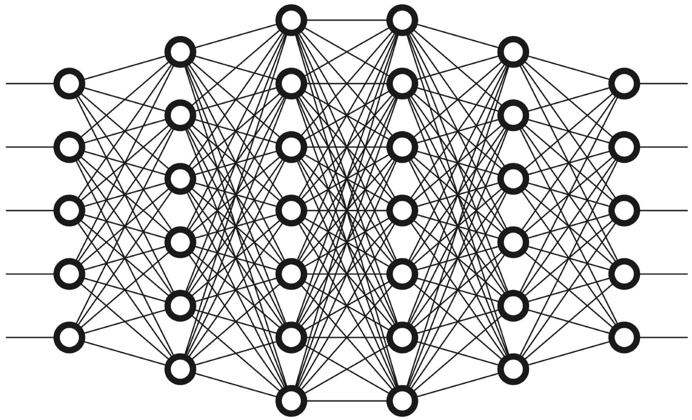
2-Run a forward pass and make a prediction

3-Calculate loss to evaluate how



How do we teach MLP to classify?

Learning good weights



Learning Recipe:

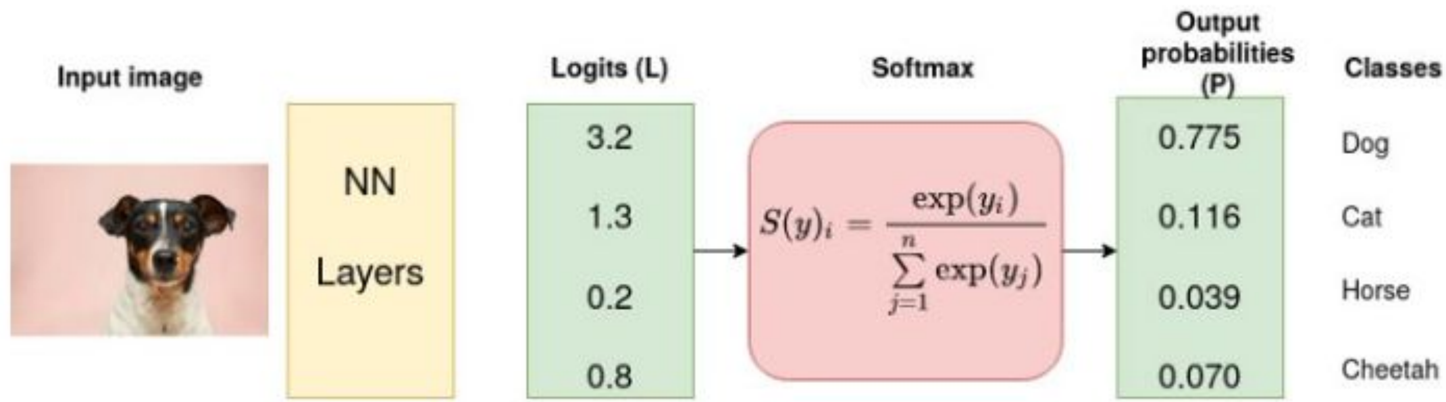
1-Initialize weights

2-Run a forward pass and make a prediction

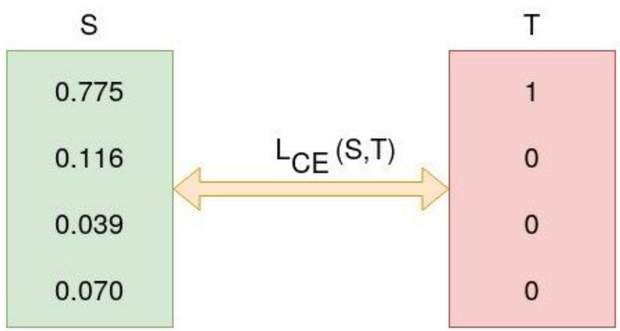
3-Calculate loss to evaluate how good/bad your prediction was

4-Adjust your weights in a way so that loss will be minimized

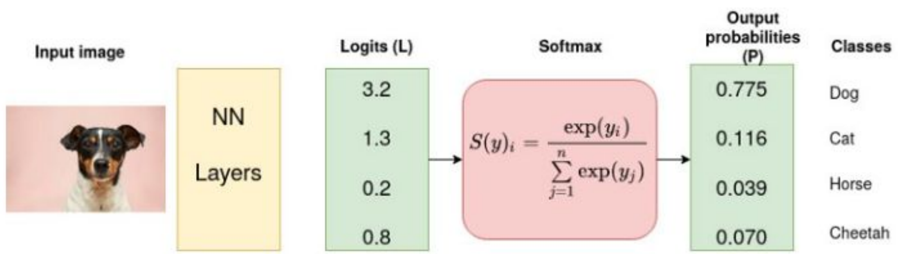
Classifying categorical variables -> Cross-entropy Loss Function



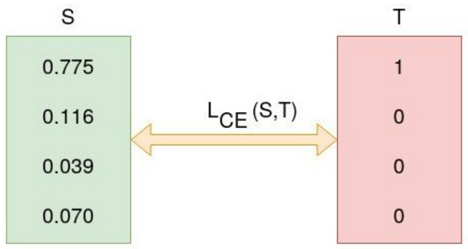
Input image source: Photo by Victor Grabarczyk on Unsplash . Diagram by author.



Cross Entropy (L) (Source: Author).



Input image source: Photo by Victor Grabarczyk on Unsplash . Diagram by author.



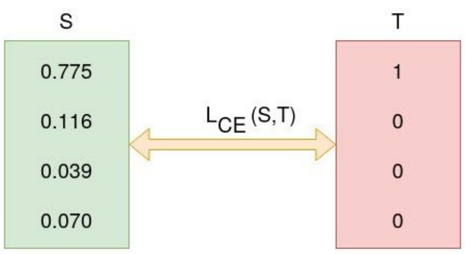
Cross Entropy (L) (Source: Author).

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

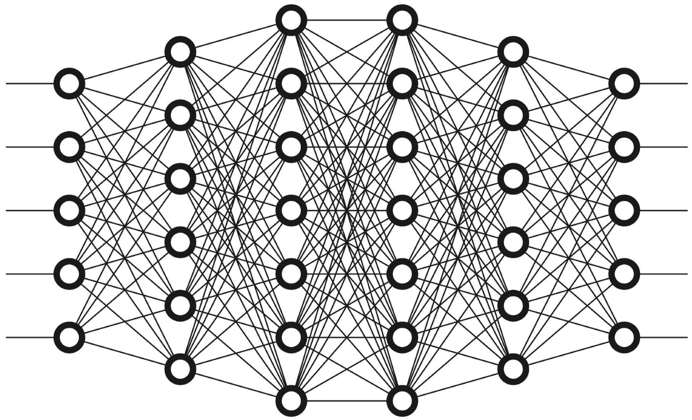


Cross Entropy (L) (Source: Author).

$$\begin{aligned} L_{CE} &= -\sum_{i=1} T_i \log(S_i) \\ &= -[1\log_2(0.775) + 0\log_2(0.126) + 0\log_2(0.039) + 0\log_2(0.070)] \\ &= -\log_2(0.775) \\ &= 0.3677 \end{aligned}$$

How do we teach MLP to classify?

Learning good weights



Learning Recipe:

1-Initialize weights

2-Run a forward pass and make a prediction

3-Calculate loss to evaluate how good/bad your prediction was

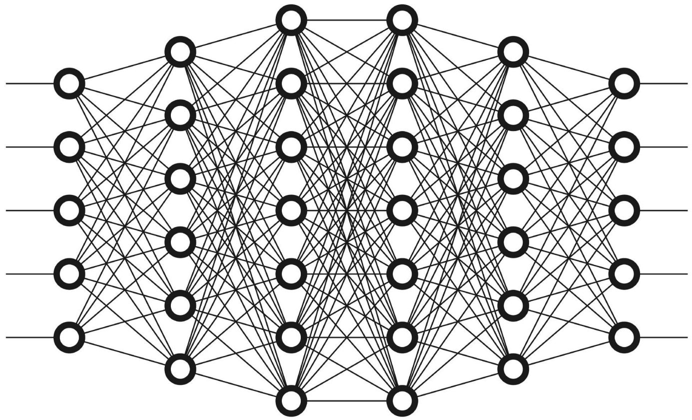
4-Adjust your weights in a way so that loss will be minimized

Optimization! -> More details in the next lecture

We will use Adam Optimizer for this tutorial

How do we teach MLP to classify?

Learning good weights



Learning Recipe:
1-Initialize weights
2-Run a forward pass and make a prediction
3-Calculate loss to evaluate how good/bad your prediction was
4-Adjust your weights in a way so that loss will be minimized

Let's train our MLP model!!