Intro to NN

What are the issues with Classic ML Algos?

- Require us to do manual feature engineering to be able to create complex boundaries
- They might not always work well with big datasets, and we are living in a big-data regime
- Works poorly for sparse data and unstructured data (image/ text/ speech data)

Clearly, all these algos have some limitations which call for a more powerful ML model that can work in the conditions mentioned above. This brings us to Neural Networks (NN).

NN models power pretty much every minute of your digital life

- Online Ads (Google, YouTube)
- Data compression: Done using Autoencoders
- Image enhancement: Eg Magic Eraser
- Gmail: Eg Autocomplete, smart reply

Where did the inspiration for a neuron come from?

NN is loosely inspired by the biological neurons found in the human brain.

In the brain, there exist biological neurons that are connected to each other, forming a network

In simple terms, we understand that

- Neuron takes input(s)
- Perform some computations.
- Ultimately, it fires/passes the output to further neurons, for further processing.

How does an Artificial neuron do its computations?

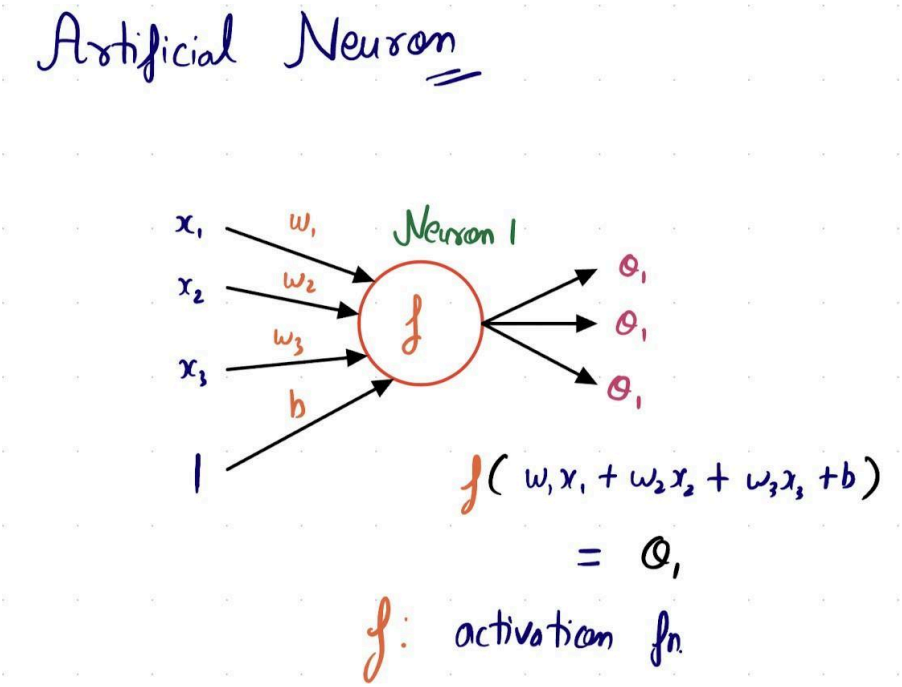
Consider an artificial neuron.

- It receives input features: x_1, x_2, x_3
- Every input has a weight associated with it: w_1, w_2, w_3
    - These weights are multiplied by the input values, thereby telling us how important a given input is to the neuron.
- Inputs are processed by taking the weighted sum.
- Also, a bias term is added.
    - The net input becomes: $w_1x_1 + w_2x_2 + w_3x_3 + b = z$ (let)
- There is a function, called activation function f, which is associated with a neuron
    - The neuron applies this function on the net input value: $f(z) = f(w_1x_1 + w_2x_2 + w_3x_3 + b)$
- The result of this function becomes the output $o_1 = f(z)$

- This output is then forwarded to other neurons.

This flow of computations is known as **Forward Propagation.**

Notice that we are going from left to right during Forward Propagation

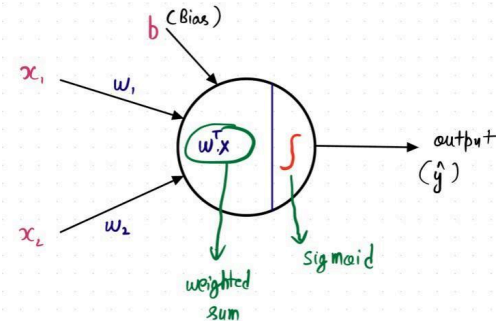


What happens if we put the sigmoid function as the activation function?

We get a Logistic Regression model, as the output of this becomes: $o_1 = \text{sigmoid}(w_1x_1 + w_2x_2 + w_3x_3 + b)$

This neuron, where the activation is a sigmoid function is called a Logistic Regression Unit (LRU)

We can diagrammatically represent a neuron with 2 inputs as:



What if the model looked the same, but the activation function is different? Would it still be called the same NN?

If we replace the activation function to a hinge loss function, we get a Linear SVM model. Forward propagation here would look like:-

- $z = w_1x_1 + w_2x_2 + \ldots + w_dx_d + b$
- $\hat{y} = f_{hinge}(z)$

Note:

- A single neuron is able to represent such powerful models, just imagine what will happen if we use multiple neurons.
- Those models would be able to represent really complex relations.

A Brief History of Artificial Neural Networks

Perceptron

- Perceptron is the very first NN-based model. It was designed by Rosenblatt in 1957
- It is different from LRU because it used a step function for activation, which is:

$$f_{perceptron}(x_i, w, b) = \begin{cases} 1, & \text{if } w^T x_i + b > 0. \\ 0, & \text{otherwise.} \end{cases}$$

Challenges:

- As we tried to increase the complexity of NN, they performed poorly.
- There was no optimal way of training the NN.
- This put the whole area in deep freeze for a decade or so.
- A breakthrough came in 1986 when **backpropagation** was introduced by Geoff Hinton and his team.
- This also brought up a lot of hype for artificial intelligence, But, all of that hype died down by 1990s
- There were two main bottlenecks:-
    - We didn't have computational power
    - Nor did we have enough data to train
    - Backprop was failing for NNs with more depth.
- This time period is called **AI winter,** where the funding for AI dried up by 1995
- Meanwhile, Geoff Hinton continued his research and finally came up with a solution to this problem in 2006.
- Also, the discovery of using ReLu and Leaky ReLu as activation functions was another breakthrough.

How does NN fare against classical ML (based on training data)?