Learning AI - The Neuron

Artificial Neuron: atomic building block of every deep learning model

Biological Inspiration:

- Inputs (dendrites) receive electrical signals
- Signals are summed and processed
- If the signal passes a threshold, the neuron fires
- Output is sent down the axon to other neurons

Above mimicked by numbers in artificial neurons

The Math

A neuron takes:

- Inputs: $x_1, x_2, \ldots, x_n$
- Weights: $w_1, w_2, \ldots, w_n$ (how important each input is)
- Bias: b (a constant offset)

It computes:

$$z = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

Or in vector form:

$$z = w \cdot x + b \leftarrow \text{linear algebra "dot product"}$$

Then passes through an activation function $f(z)$ to produce output:

$$y = f(z)$$

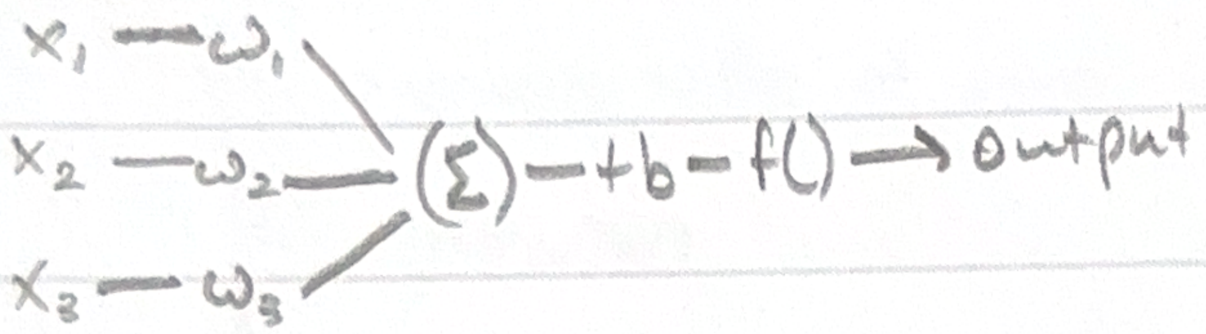
Activation Functions

These functions introduce non-linearity, letting the network learn complex patterns. W/o them, stacking neurons would still be equivalent to a single linear transformation. (NO matter # layers)

Common activation functions:

- Sigmoid: $f(z) = \frac{1}{1+e^{-z}}$ (Outputs range b/w 0 & 1)
- ReLU (Rectified Linear Unit): $f(z) = \max(0, z)$ (Fast, good for deep nets)
- Tanh: $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ (Outputs b/w -1 & 1)

Visualizing a neuron:



Steps:

1. Multiply inputs by weights
2. Add them up (plus bias)
3. Apply activation
4. Output a number

Python Implementation in documents/Learn AI/neuron.ipynb

The neuron's output/activation is the signal that it passes on

How Neurons form a Network

- A layer is just a bunch of neurons running in parallel, each w/ its own weights/bias
- Layers stack to form networks - the output of one layer is the input to the next
- In "deep learning", the "deep" means many layers

e.g. 1. Input layer: raw features (pixels, words, numbers)

2. Hidden layers: transformations that learn features
3. Output layer: final prediction (class, val, next word, etc)