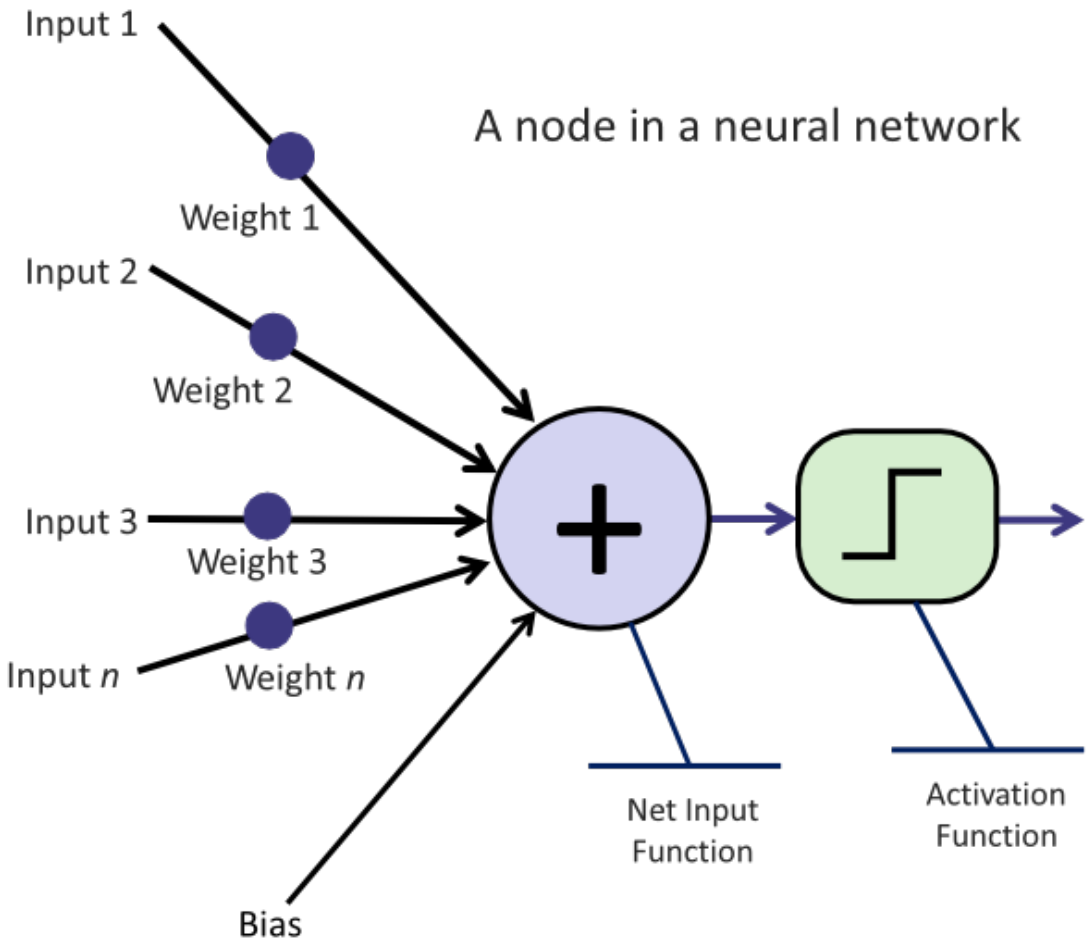
practicum

building ai knowledge

Deep Learning, Opening the Machine

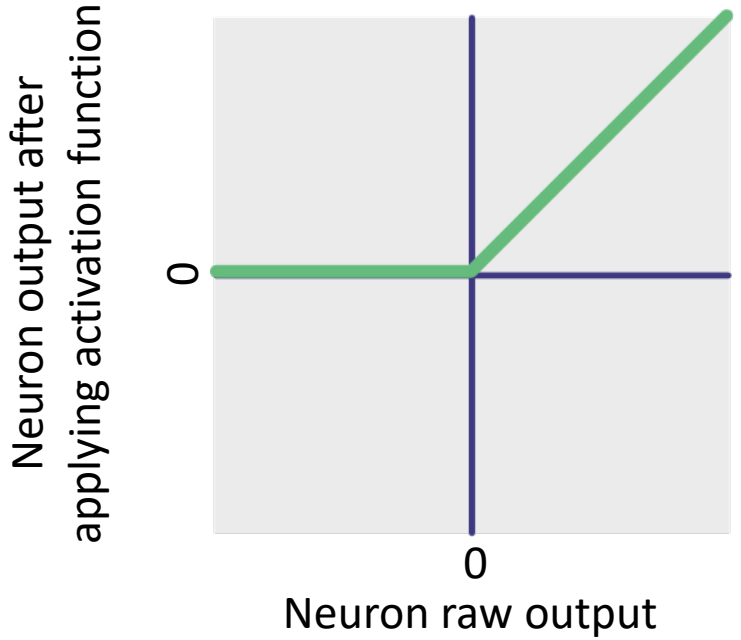
Remember The Node?



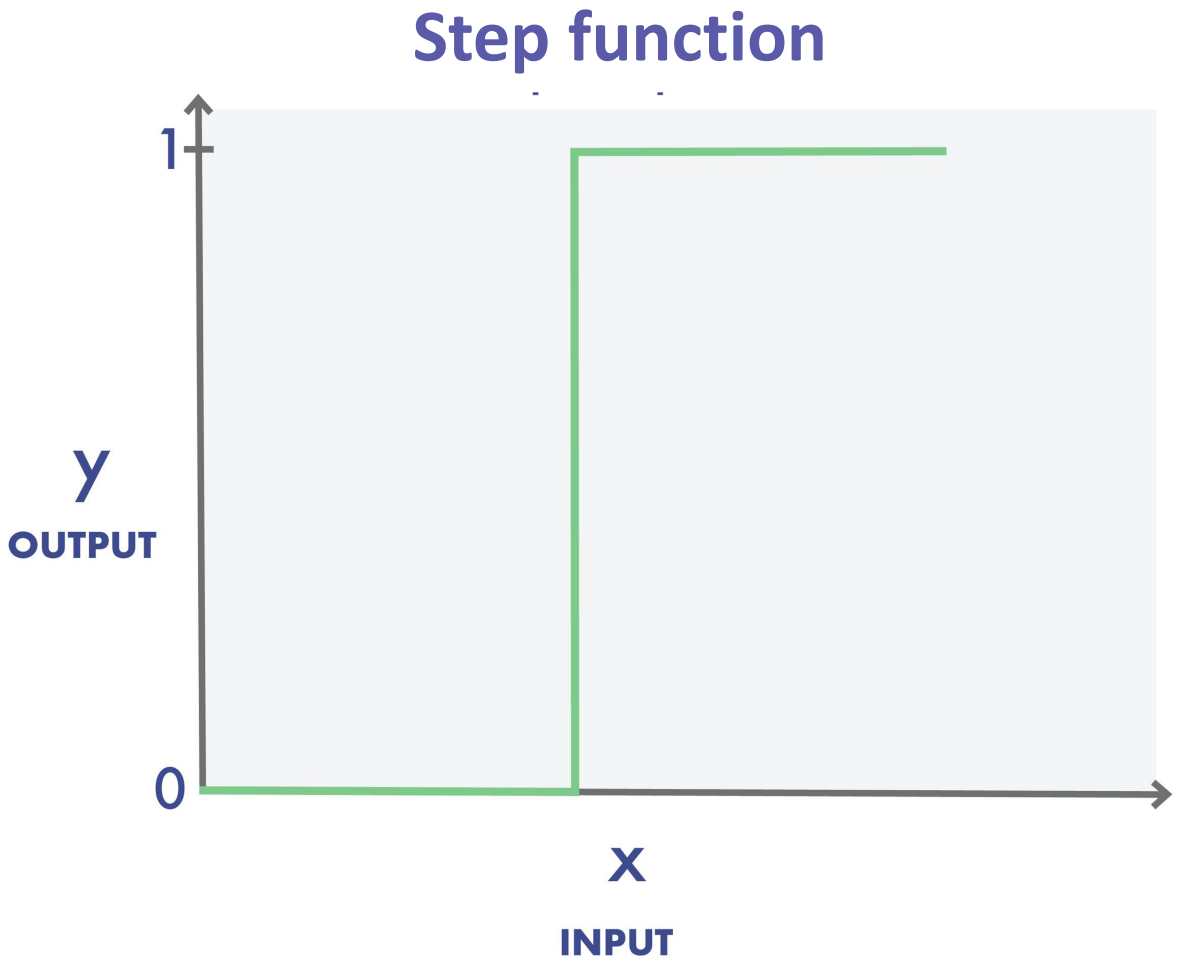
1. Linear Transformation

$$Sum = w_1 \times x_1 + w_2 \times x_2 + \cdots + w_n \times x_n + bias$$

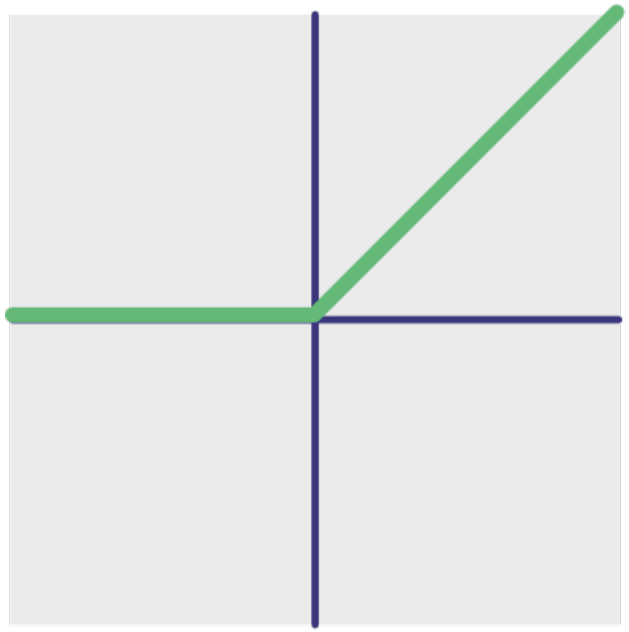
2. Activation Function



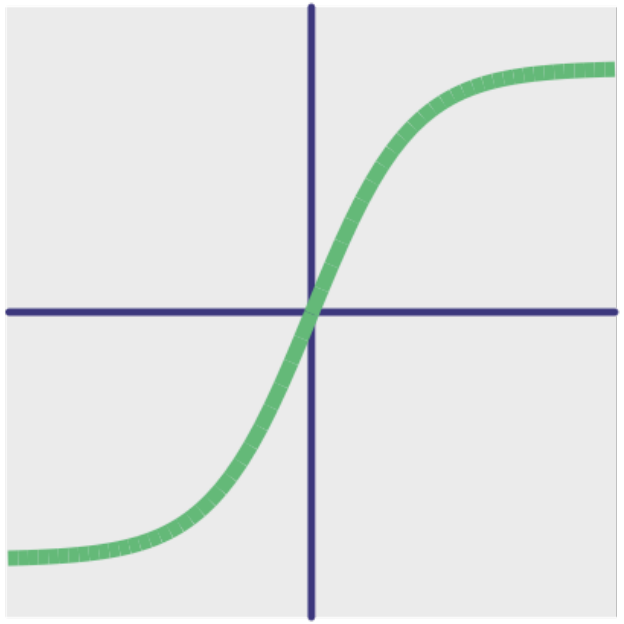
What do Activation Functions do?



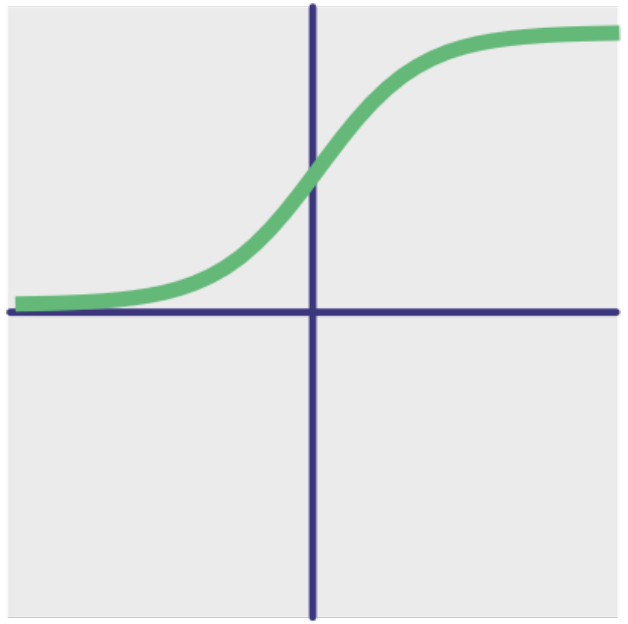
Common Activation Functions



ReLU
(Rectified Linear Unit)



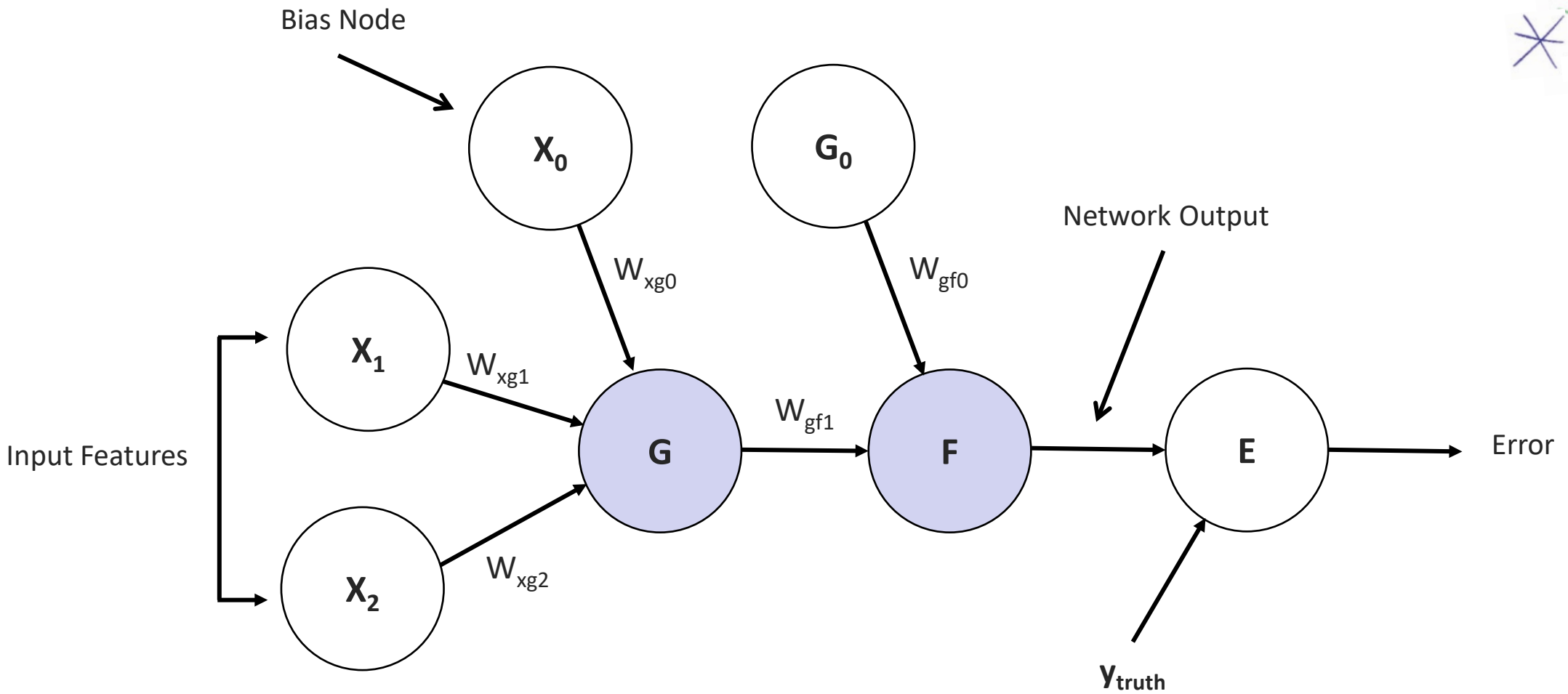
tanh
(Hyperbolic tangent)

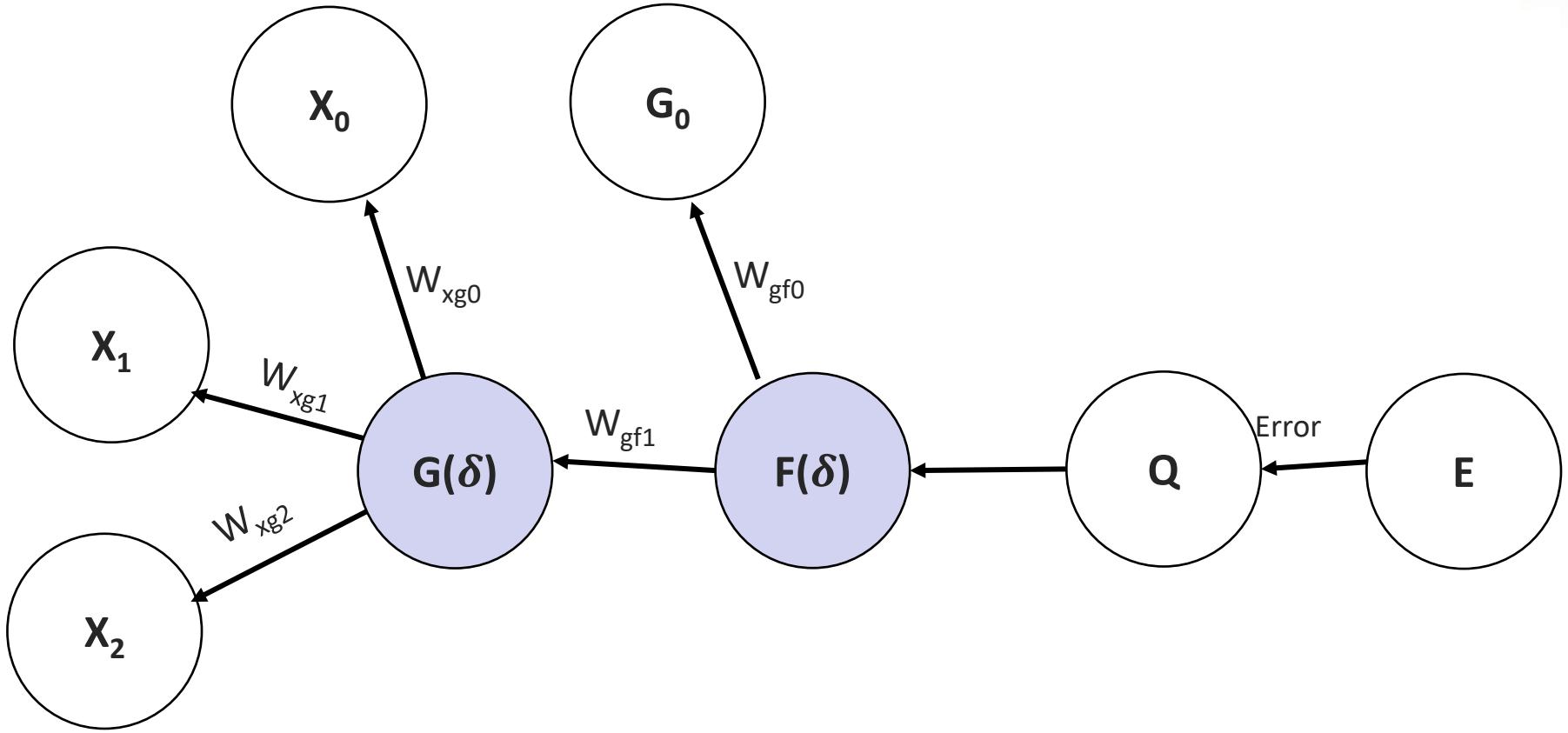


Sigmoid

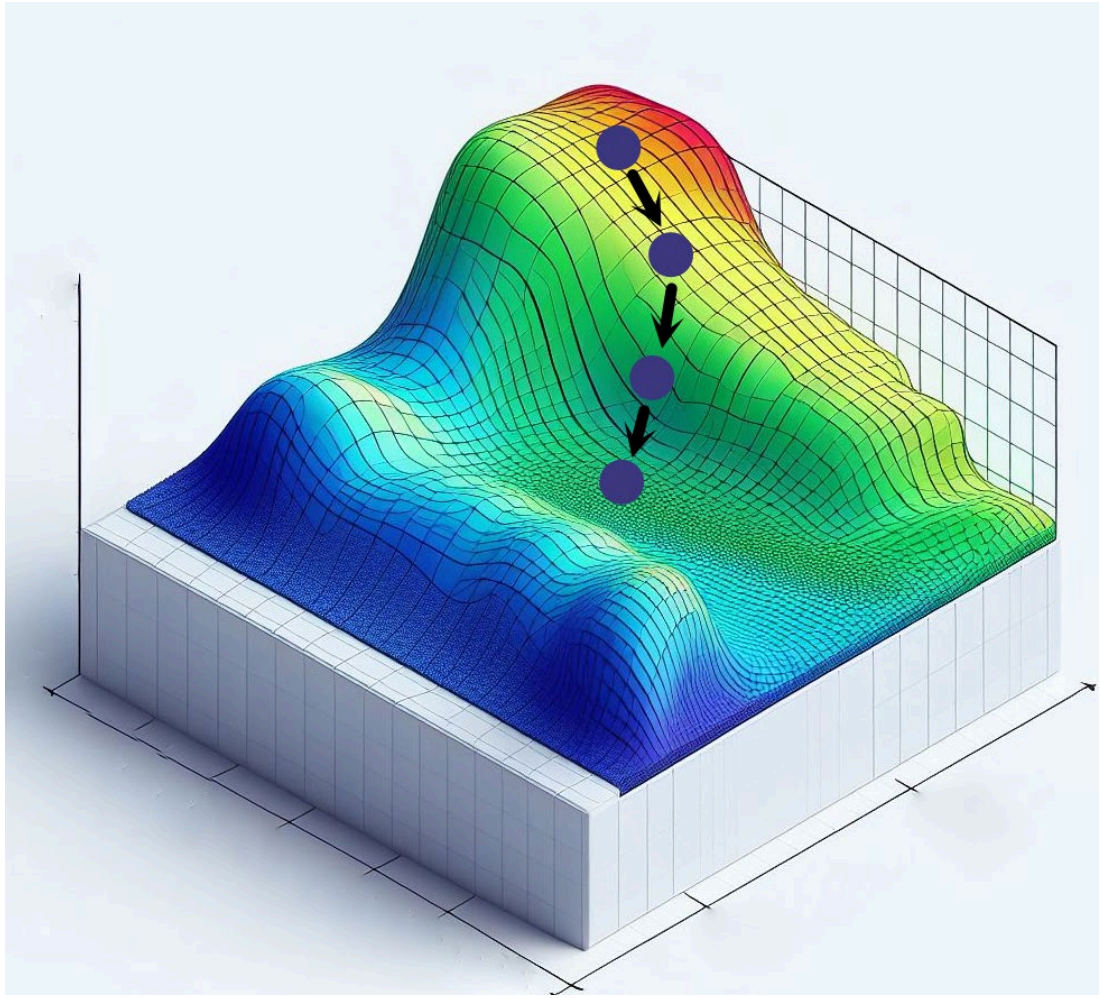
They’re Random (Usually…)

This image was generated using AI tools


Bias Node
X_0
G_0
Network Output
W_{xg0}
W_{gf0}
X_1
W_{xg1}
W_{gf1}
Input Features
G
F
E
Error
X_2
W_{xg2}
y_{truth}
Input Layer
Hidden Layer
Output Layer
Error/Loss Function
Forward Propagation

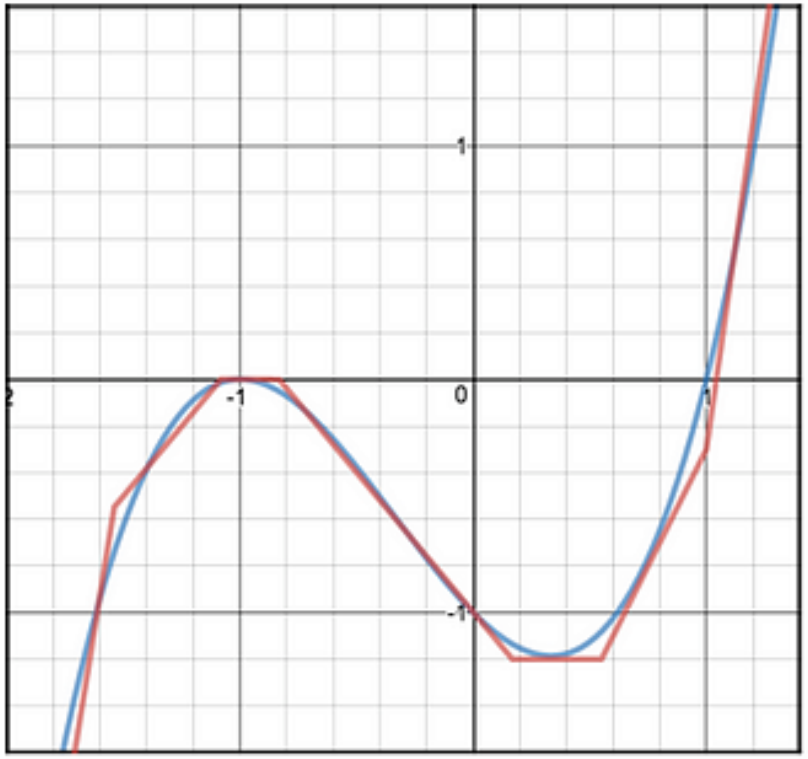

X0
G0
X1
Wxg0
Wgf0
Wxg1
Wgf1
G(δ)
F(δ)
Q
Error
E
Wxg2
X2
Input Layer
Hidden Layer
Output Layer
Optimizer
Error/Loss Function
Back Propagation

The Low Down



This image was generated using AI tools

Universal Approximation!

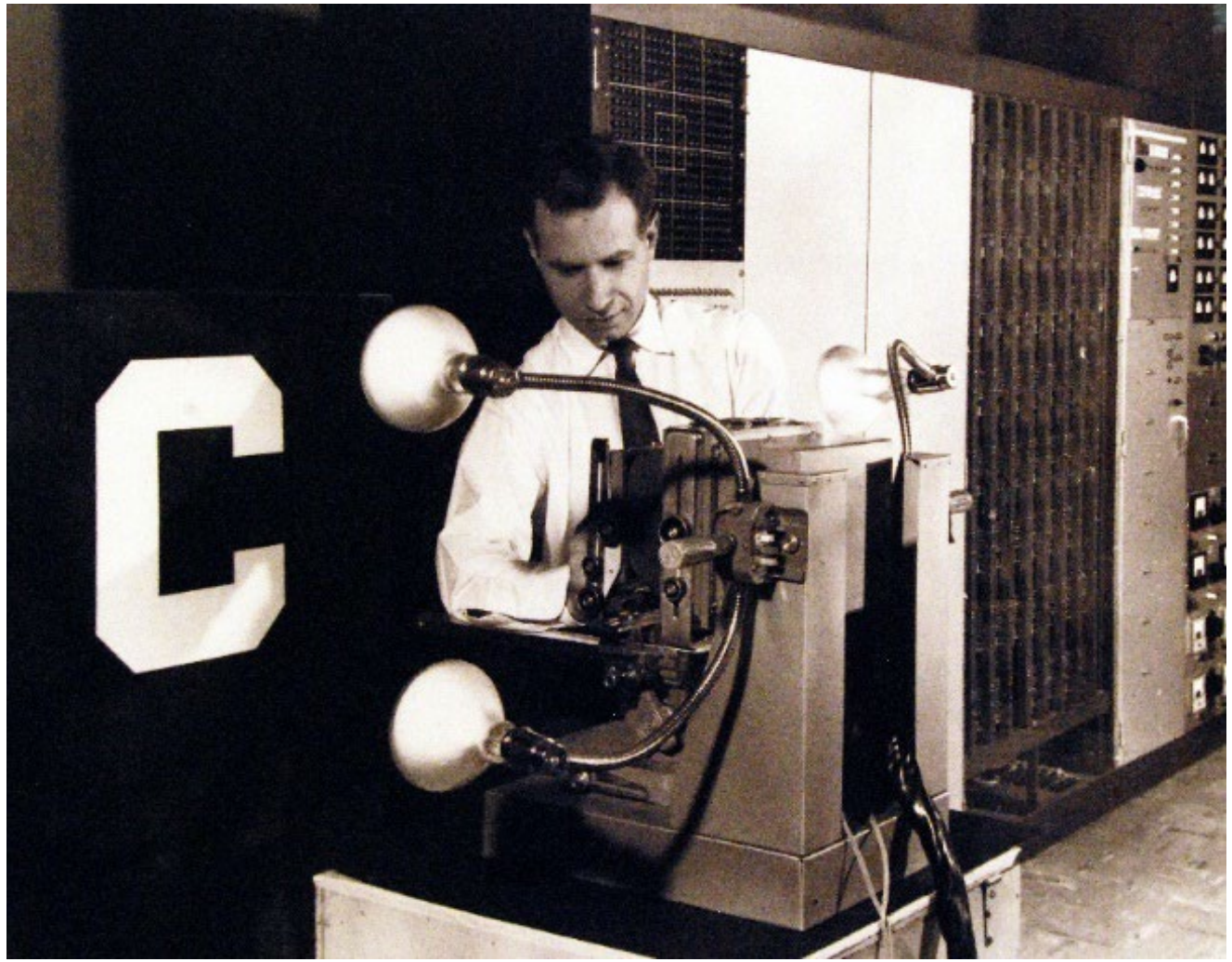


$$n_1(x) = Relu(-5x - 7.7)$$
$$n_2(x) = Relu(-1.2x - 1.3)$$
$$n_3(x) = Relu(1.2x + 1)$$
$$n_4(x) = Relu(1.2x - .2)$$
$$n_5(x) = Relu(2x - 1.1)$$
$$n_6(x) = Relu(5x - 5)$$

$$Z(x) = -n_1(x) - n_2(x) - n_3(x) + n_4(x) + n_5(x) + n_6(x)$$

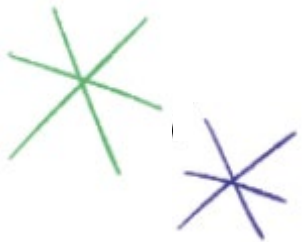
Source: Fortuner, B (2017). *Can neural networks solve any problem?* Medium.com: Towards Data Science

The Lonely Node



Source: Fortuner, B (2017). *Can neural networks solve any problem?* Medium.com: Towards Data Science

Exercise

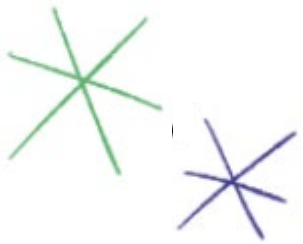


The Perceptron

02_code_a_perceptron.ipynb

This notebook will walk you through building and training your own binary classification model, then using it to make predictions!

Exercise



Look at This

03_mnist_classifier.ipynb

This notebook will walk you through training an image classification model using a full neural network.