Neural Networks and Optimization I

Dr. Parlett-Pelleriti

Basic Neural Network Components

- Nodes
- Weights
- Biases
- Activation Functions
- Loss
- Universal Function Approximation

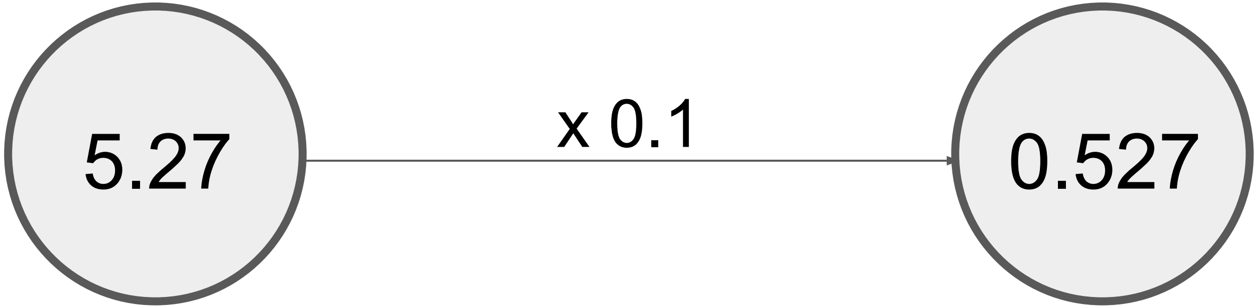
Architecture

Nodes

Nodes Hold Values

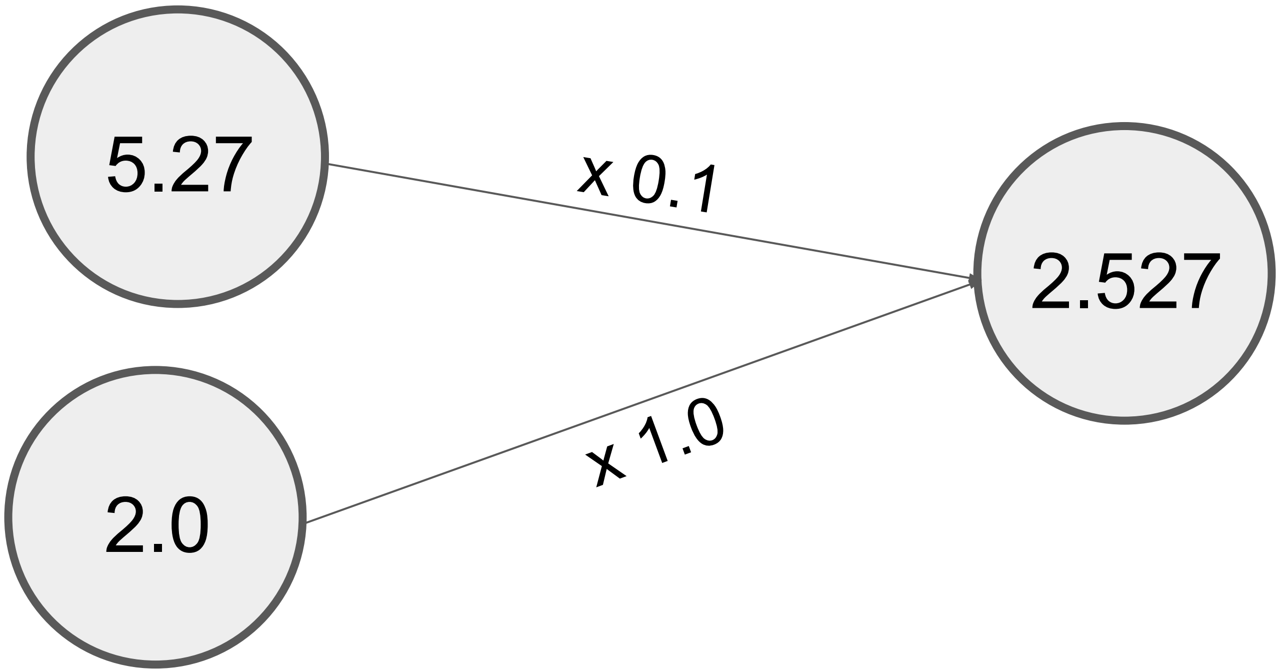
Weights

Weights multiply the number in a previous node and add it to the next node



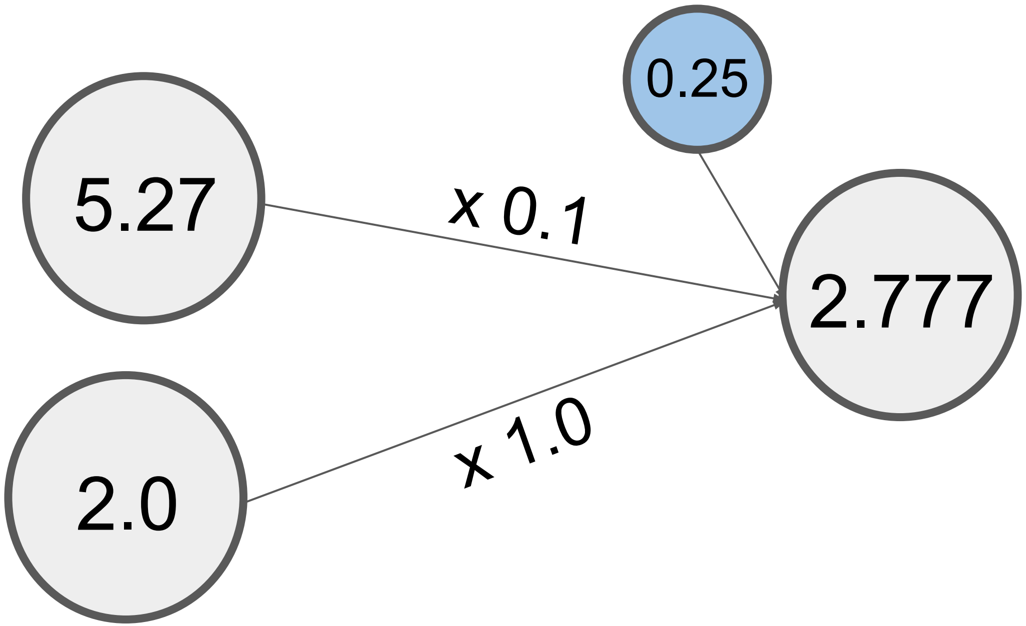
Weights

We can have multiple weights feeding into one node



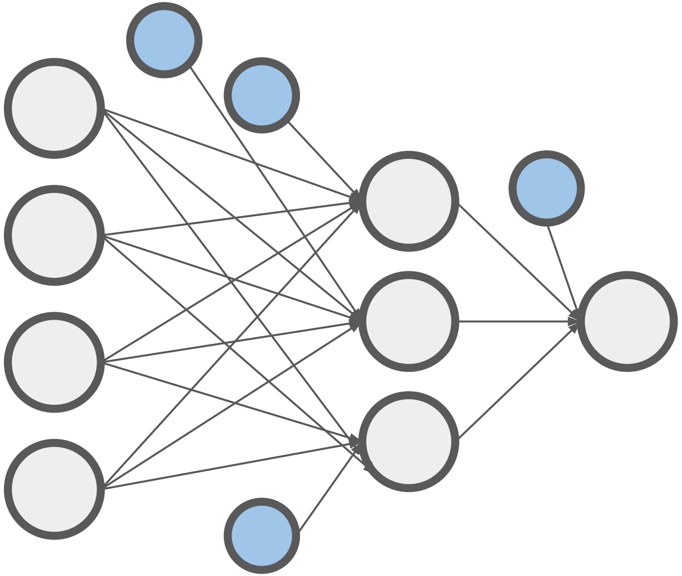
Biases

Biases move the value of a node up (for positive values) or down (for negative values) no matter what the weights and previous nodes’ values were



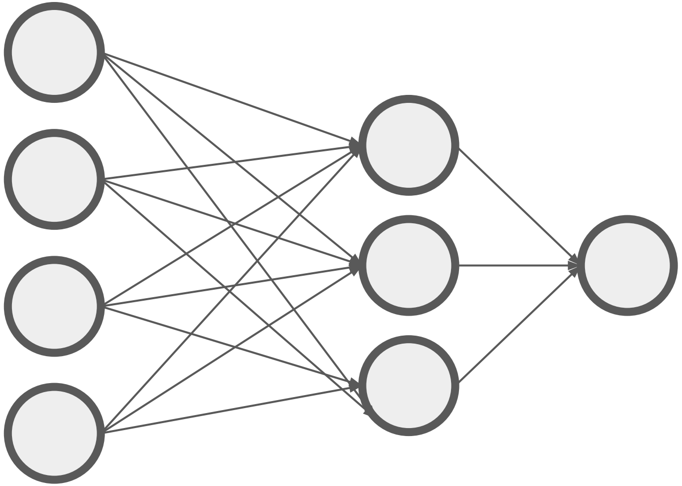
Biases

Together, nodes, weights, and biases make up the core structure of a neural network



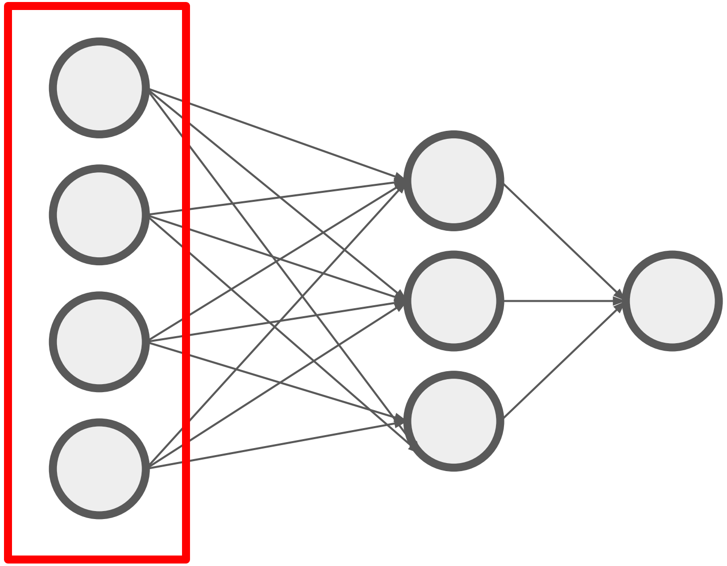
Biases

Together, nodes, weights, and biases make up the core structure of a neural network **(usually we don't show the biases, but they're there)**



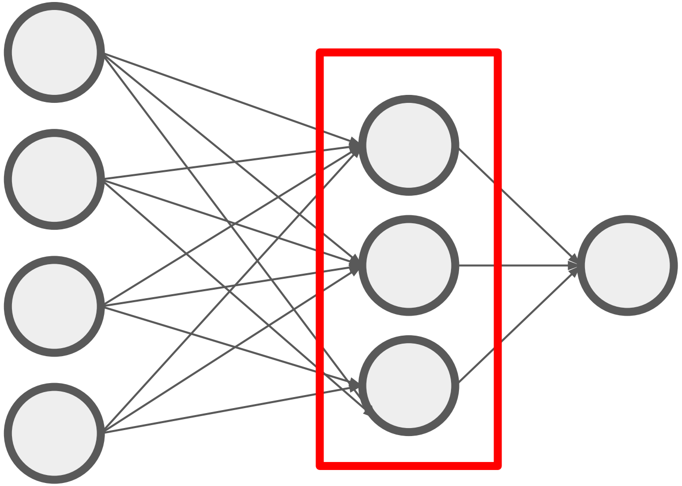
Layers

Nodes at the same level of depth are a **layer**



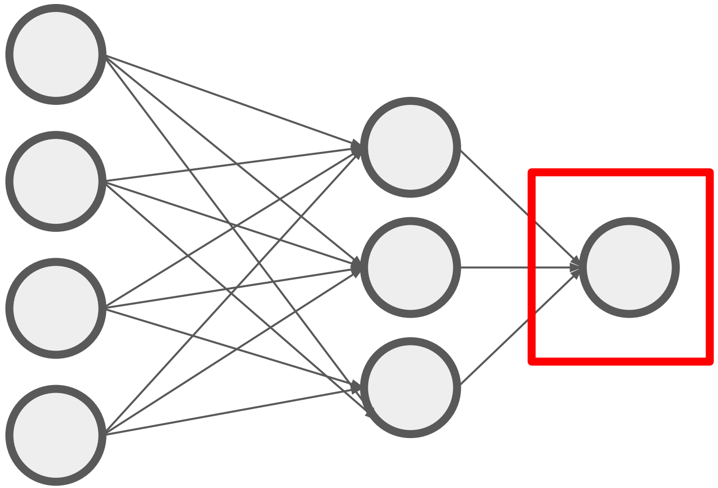
Layers

Nodes at the same level of depth are a **layer**



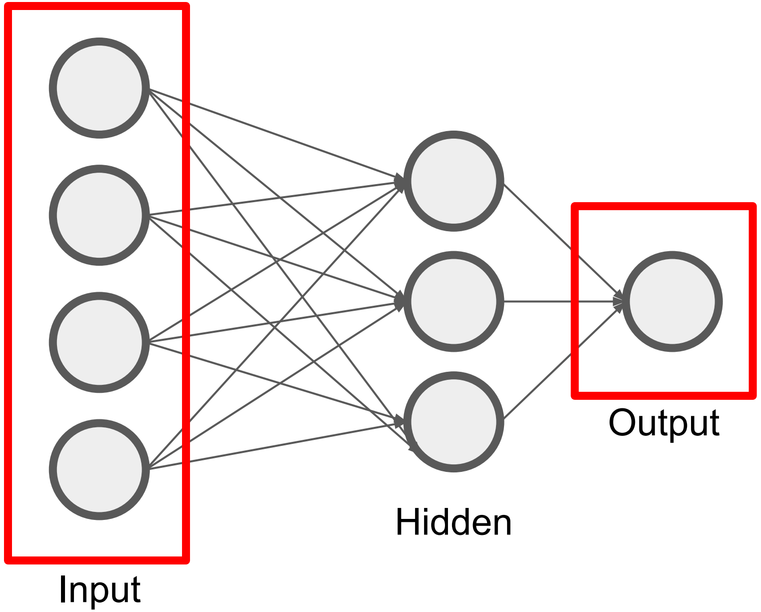
Layers

Nodes at the same level of depth are a **layer**



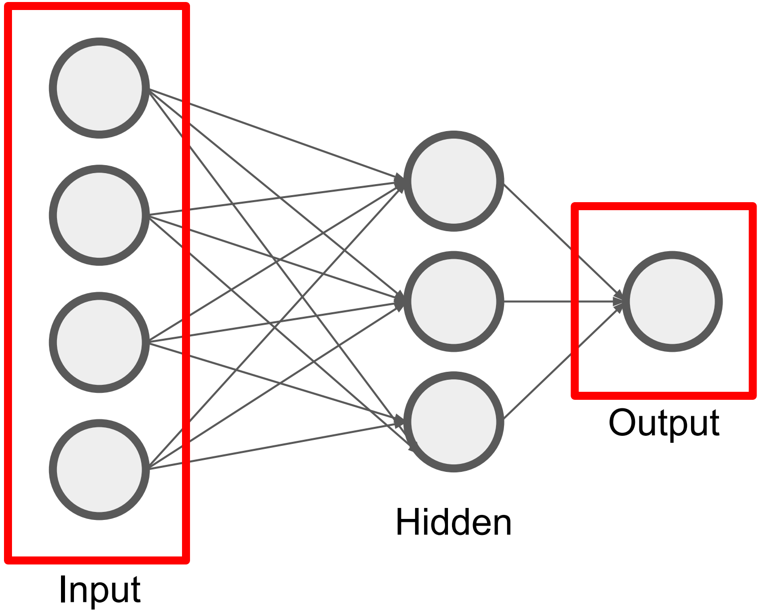
Layers

Two layers every NN must have are in **input layer** (what data is going in) and an **output layer** (what prediction is being made)



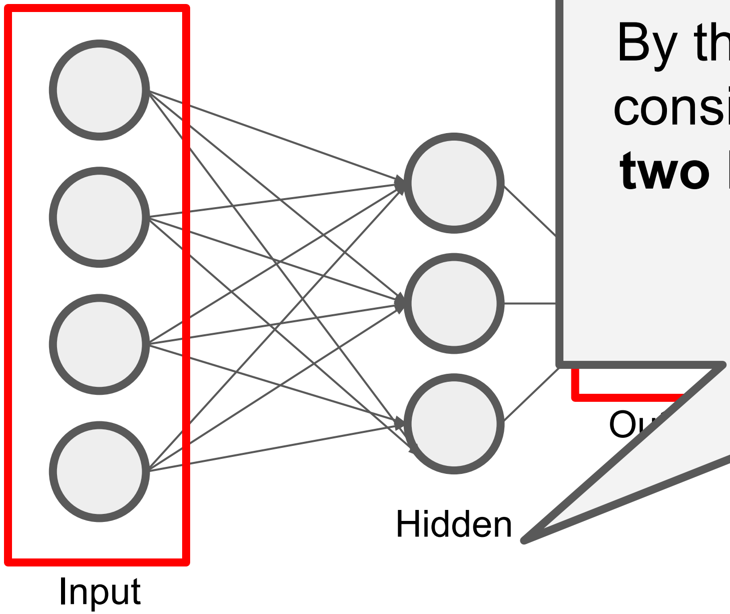
Layers

Any layer in between is a **hidden layer**.



Layers

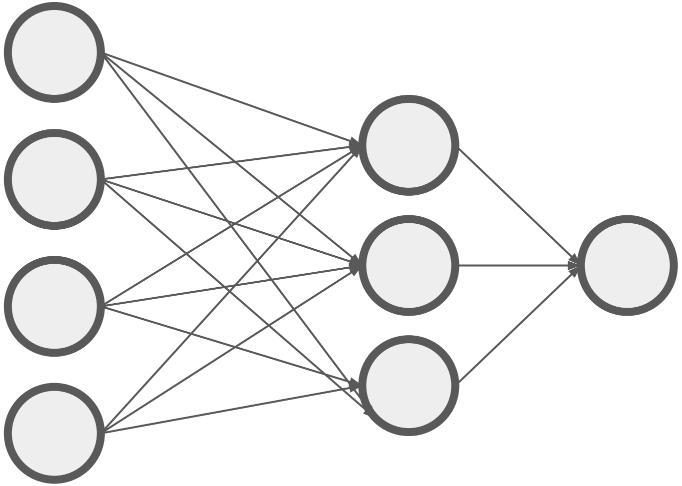
Any layer in between is a **hidden layer**.



By the way, a neural network is often considered "**deep**" if it has more than **two hidden layers**…but that's just a rule of thumb

Math Notation

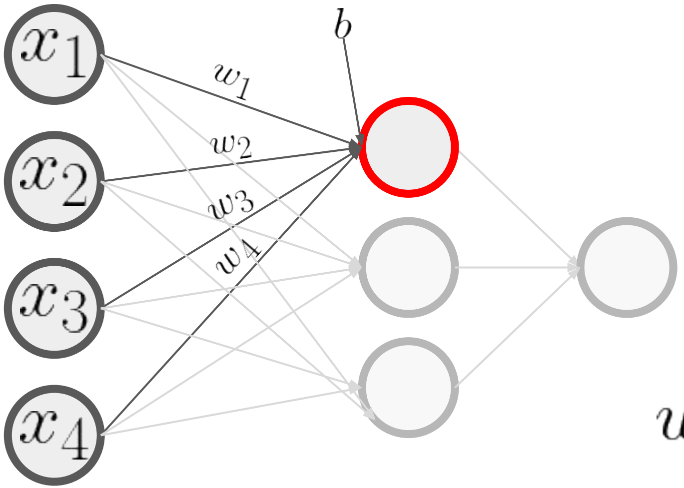
The value of a node is a **linear combination** of all the nodes in the previous layer that are connected to it.



$$\mathbf{w}^T\mathbf{x} + b$$
$$\mathbf{w} \cdot \mathbf{x} + b$$

Math Notation

The value of a node is a **linear combination** of all the nodes in the previous layer that are connected to it.



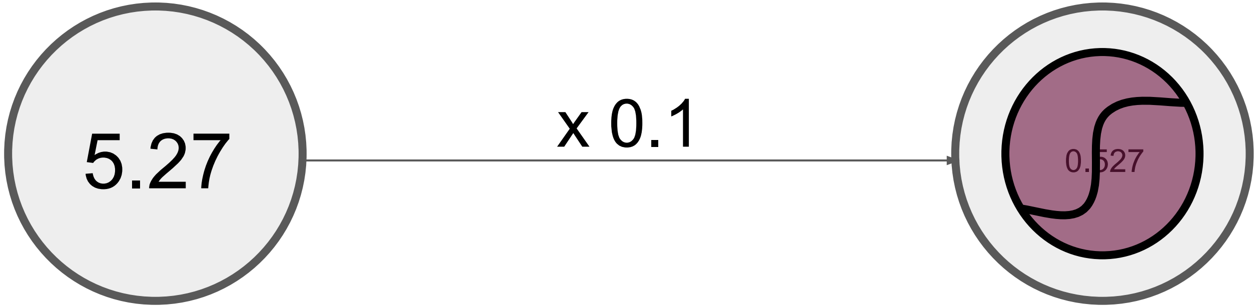
$$\mathbf{w}^T\mathbf{x} + b$$

$$\mathbf{w} \cdot \mathbf{x} + b$$

$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

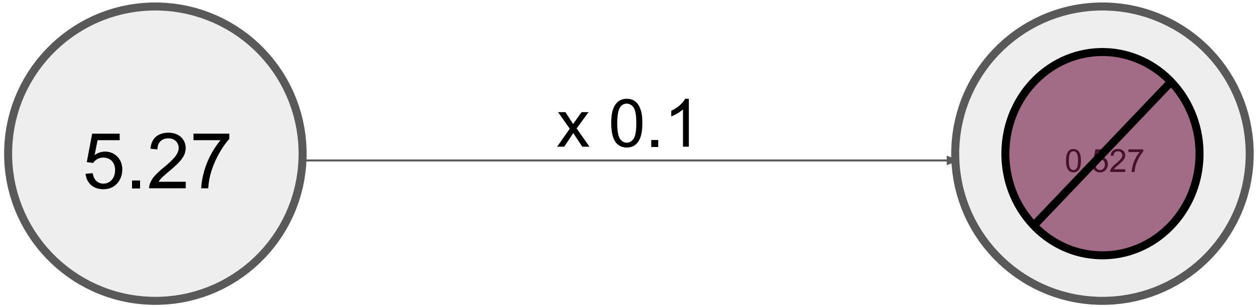
Activation Functions

Now with non-linearity!



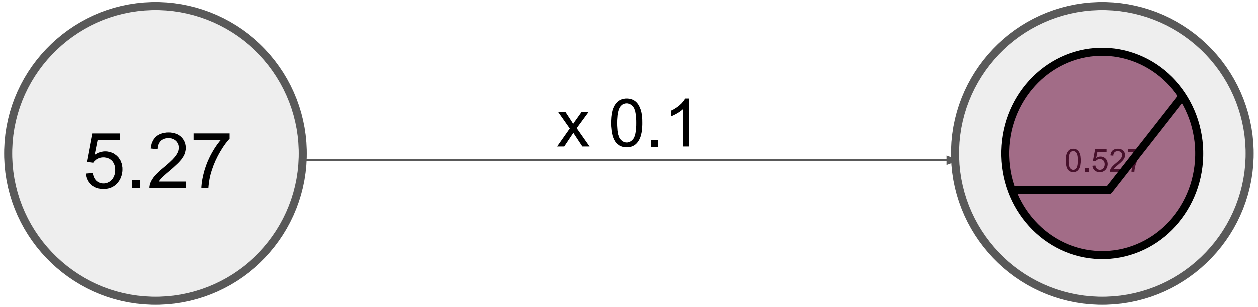
Activation Functions

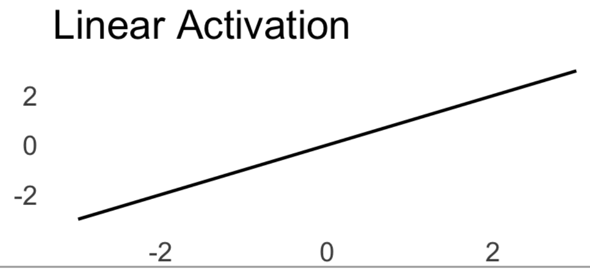
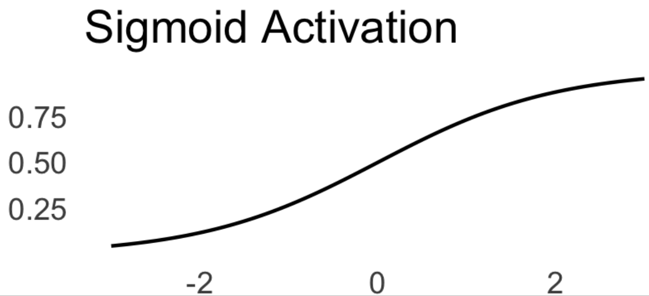
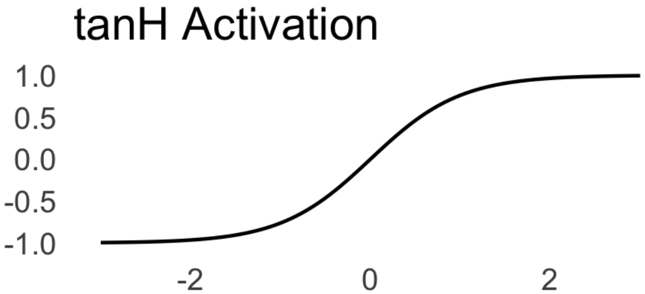
Now with non-linearity!

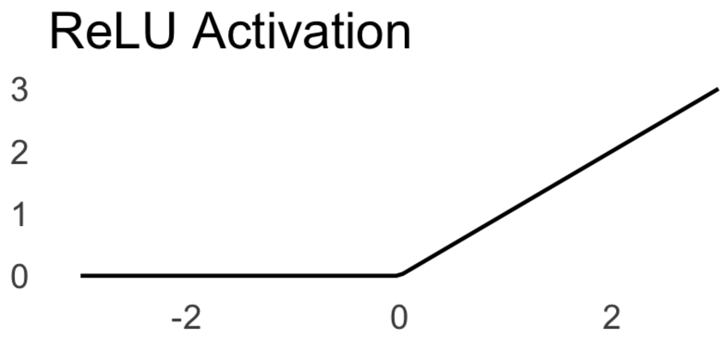
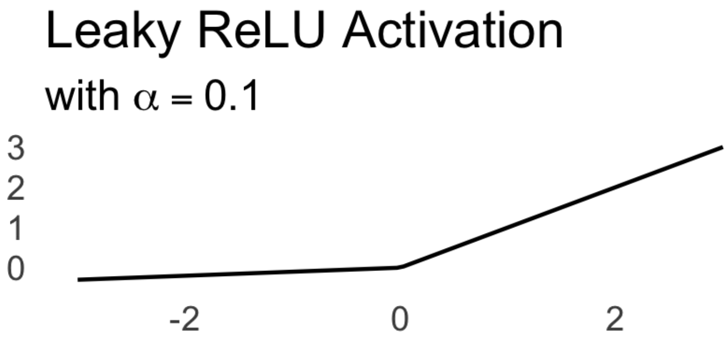


Activation Functions

Now with non-linearity!

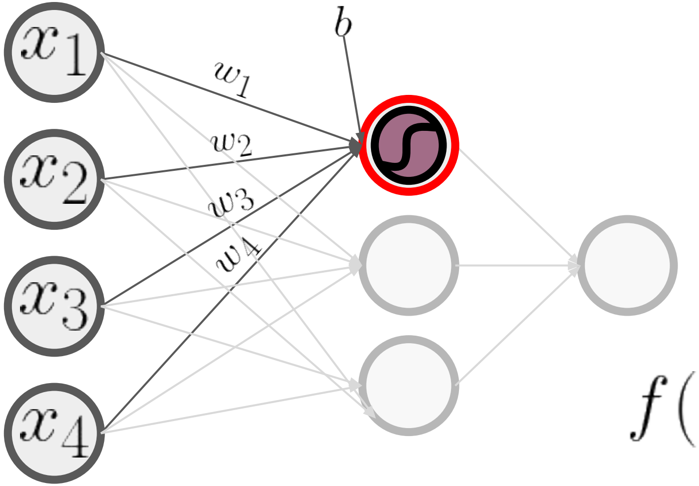


Function	f(x)	Graph
linear	$f(x) = x$	
sigmoid	$f(x) = \dfrac{1}{1 + e^{-x}}$	
tanh	$f(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	

Function	f(x)	Graph
ReLu	$f(x) = max(0, x)$	
Leaky ReLu	$f(x) = max(\alpha * x, x)$	

Math Notation

The value of a node is a **linear combination** of all the nodes in the previous layer that are connected to it.



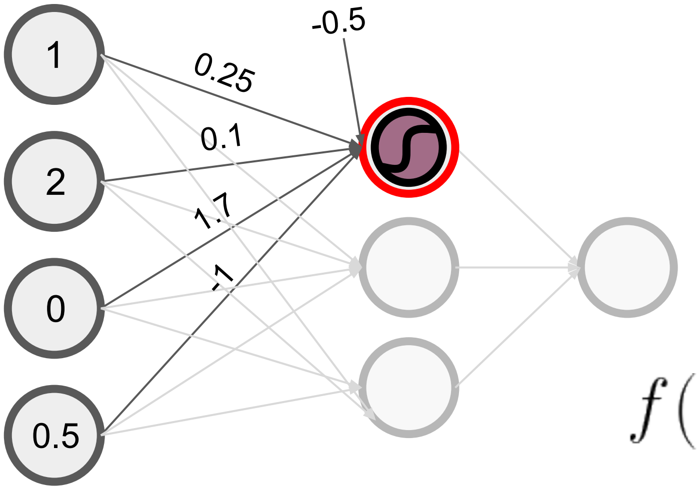
$$f(\mathbf{w}^T\mathbf{x} + b)$$

$$f(\mathbf{w} \cdot \mathbf{x} + b)$$

$$f(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b)$$

Math Notation

The value of a node is a **linear combination** of all the nodes in the previous layer that are connected to it.



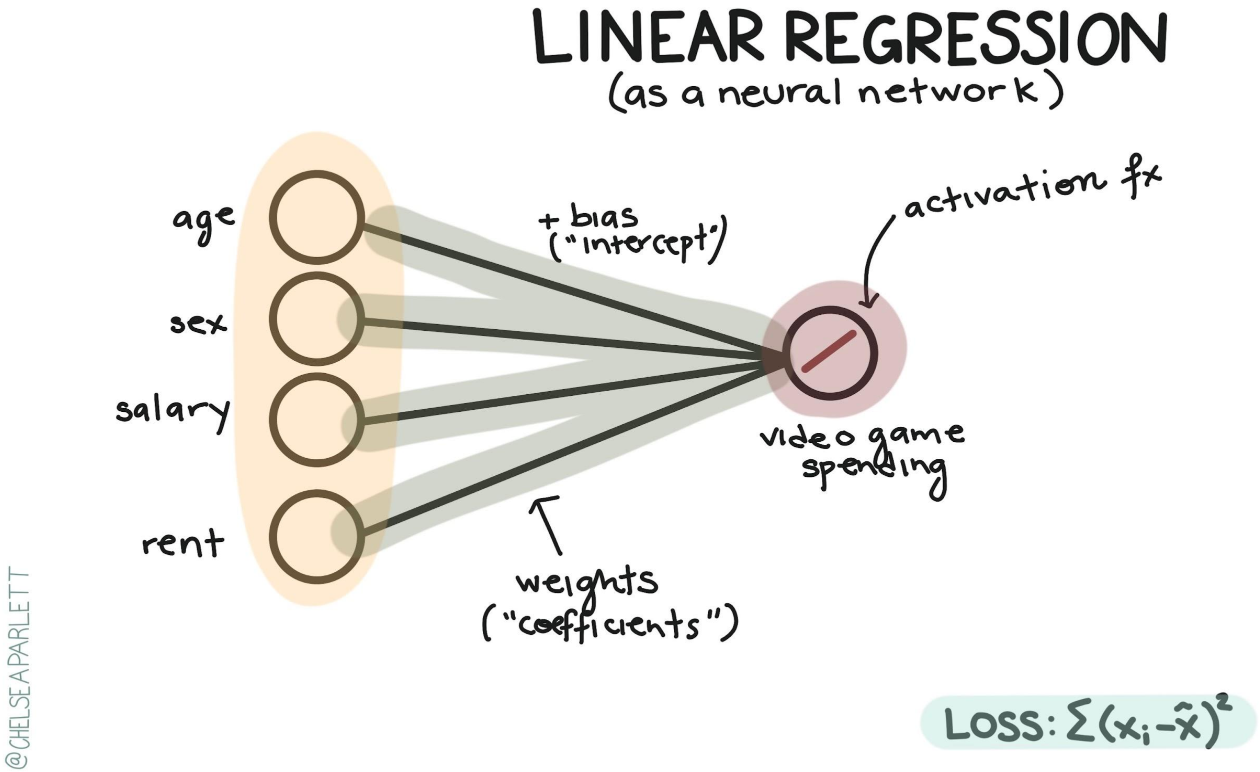
$$f(\mathbf{w}^T\mathbf{x} + b)$$

$$f(\mathbf{w} \cdot \mathbf{x} + b)$$

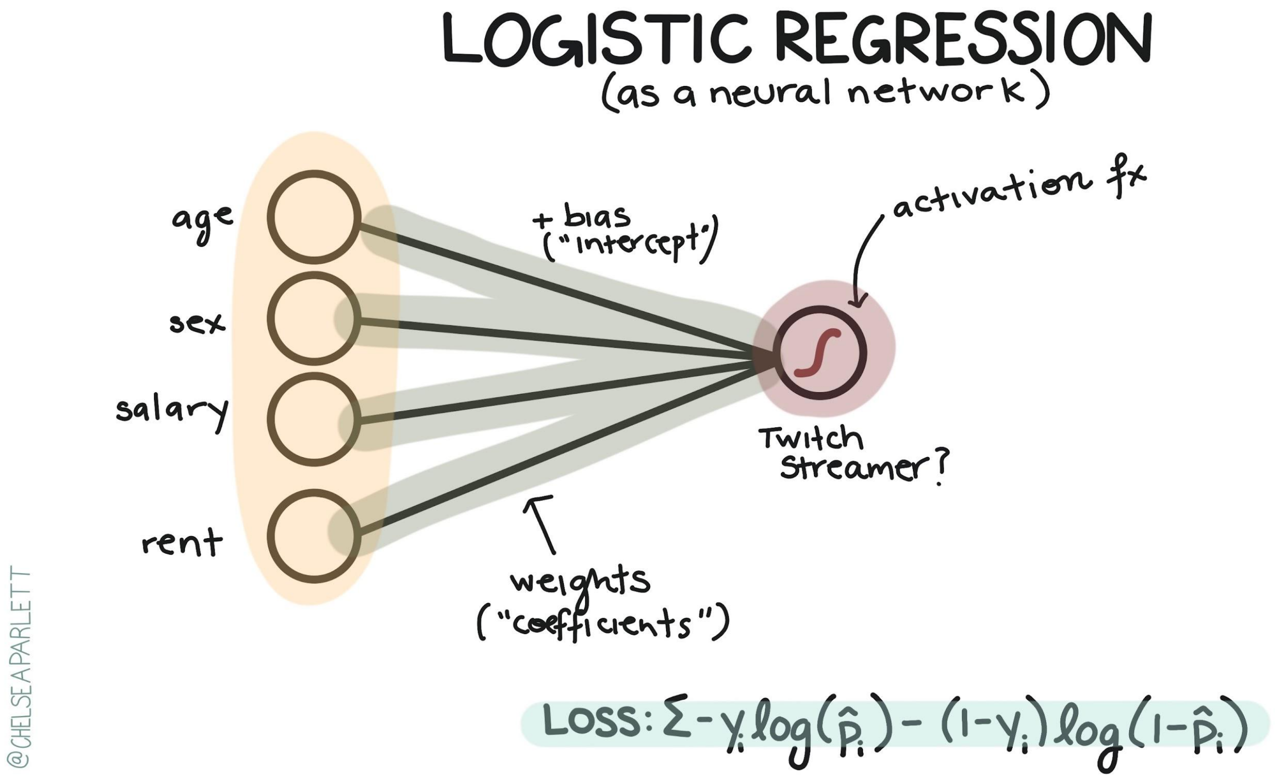
$$f(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b)$$

Familiar Models as Neural Networks

Linear Regression as a NN



Logistic Regression as a NN



Loss Functions

Loss Functions

A **metric** that measures the **performance** of your model where **lower** is better

Common Loss Functions (continuous)

MSE

$$\frac{1}{N}\sum_{i=1}^{N}(\text{actual} - \text{predicted})^2$$

MAE

$$\frac{1}{N}\sum_{i=1}^{N}|\text{actual} - \text{predicted}|$$

Common Loss Functions (categorical)

Log Loss/ Binary Cross Entropy

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)$$

Hinge Loss

$$\sum_{i=1}^{N} max(0, 1 - t_i \cdot y_i)$$

Universal Function Approximation