

Neural Networks and Optimization I

Dr. Parlett-Pelleriti

Basic Neural Network Components

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

Architecture

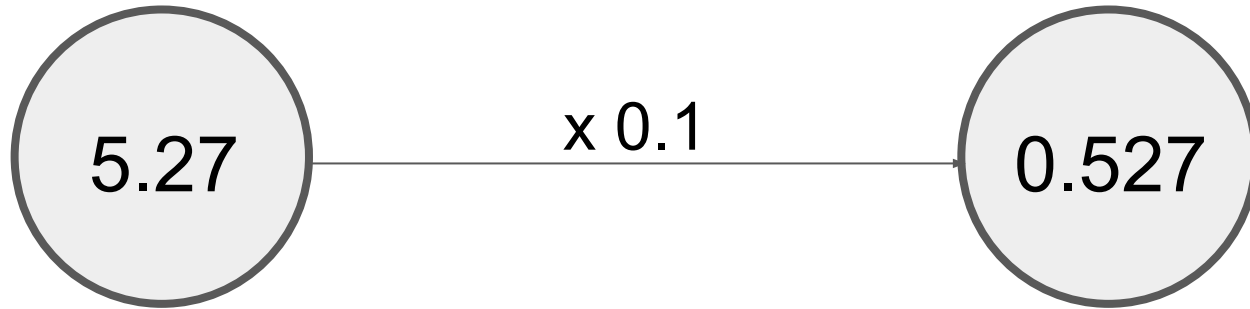
Nodes

Nodes Hold Values



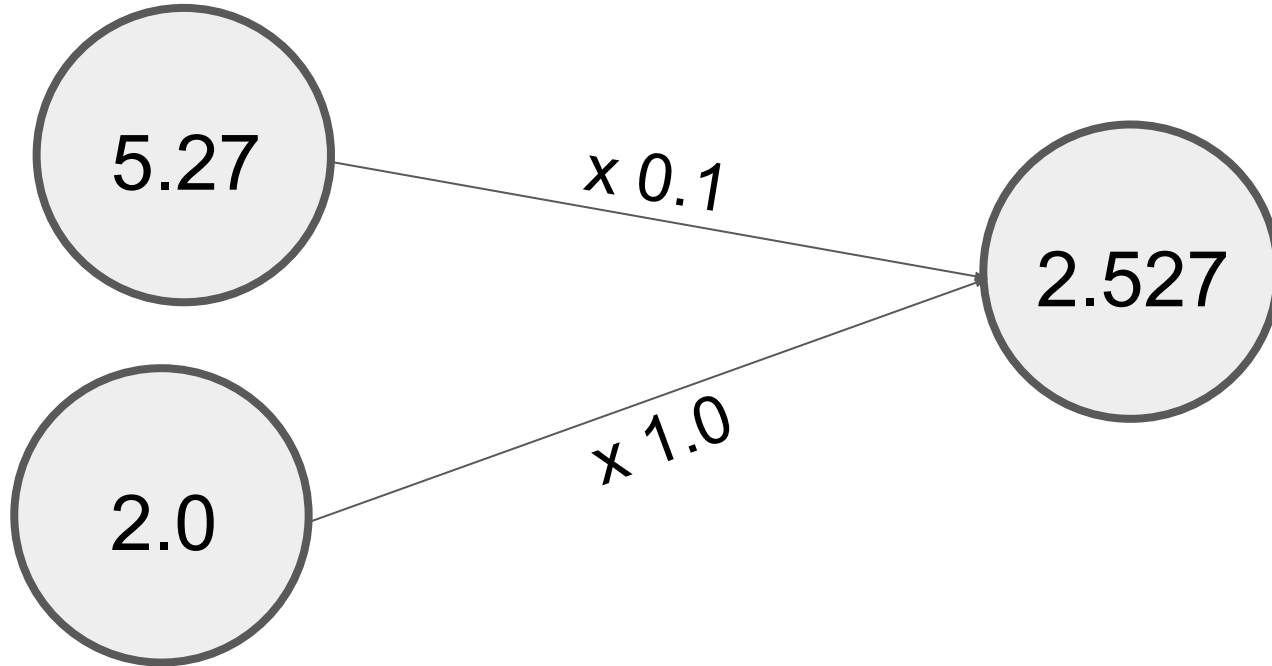
Weights

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



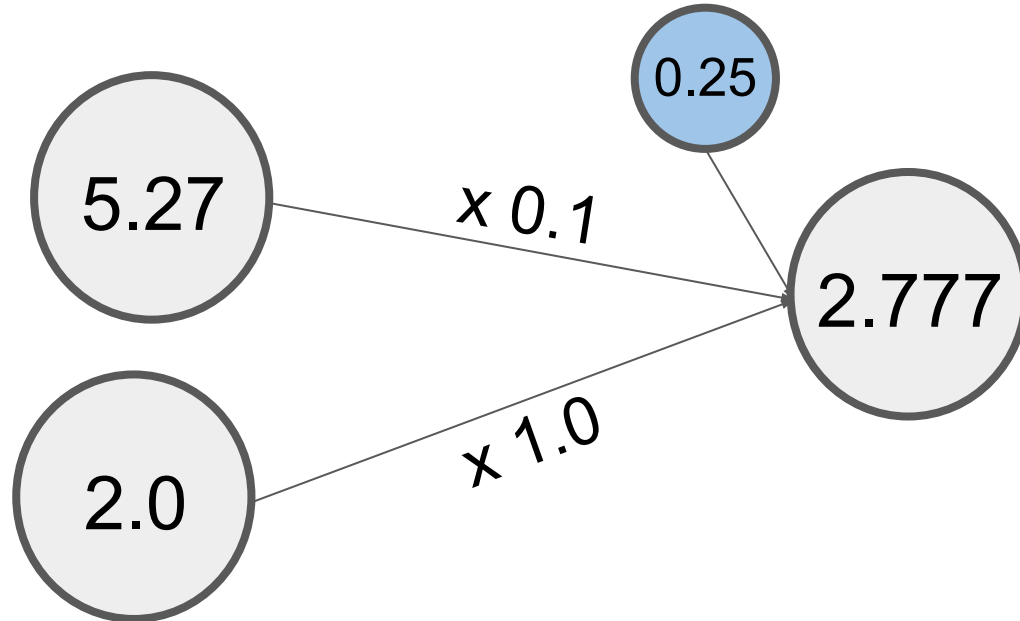
Weights

We can have multiple weights feeding into one node



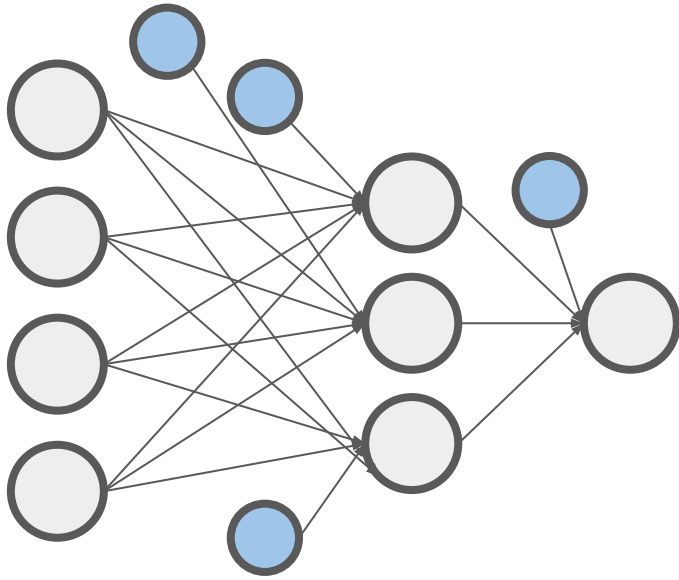
Biases

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



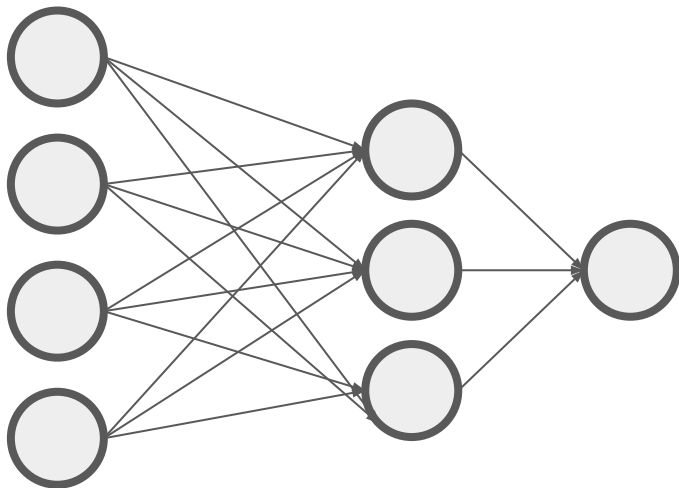
Biases

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



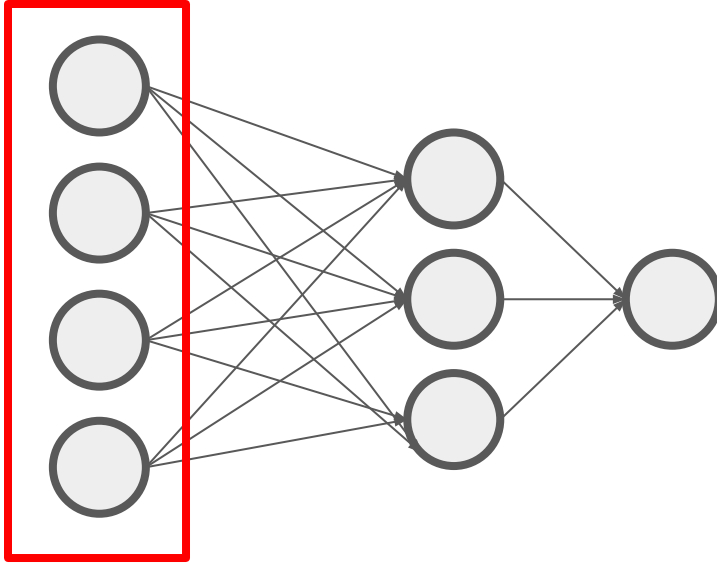
Biases

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)**



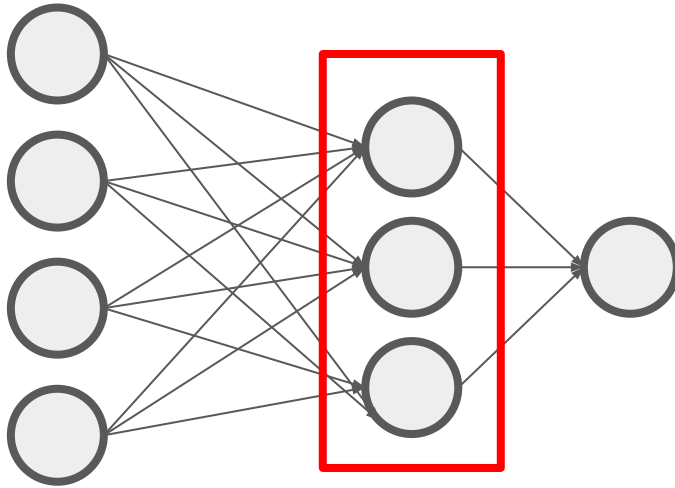
Layers

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



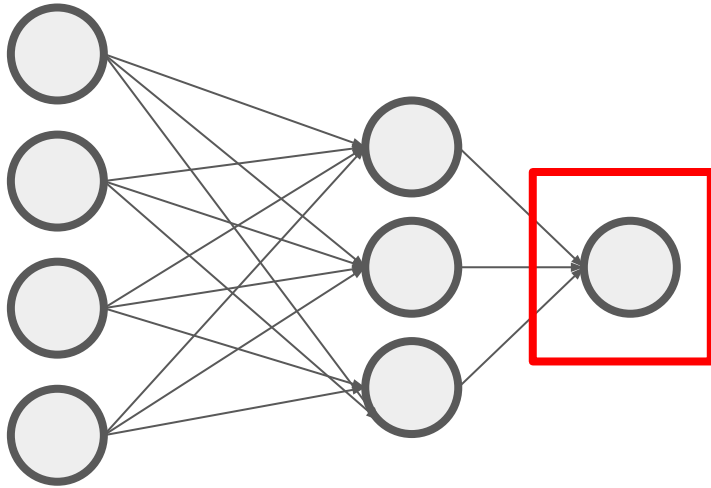
Layers

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



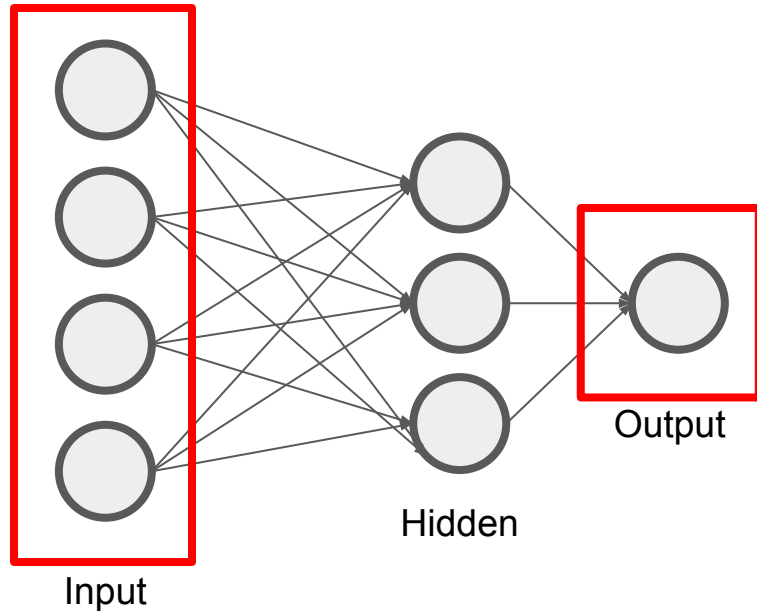
Layers

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



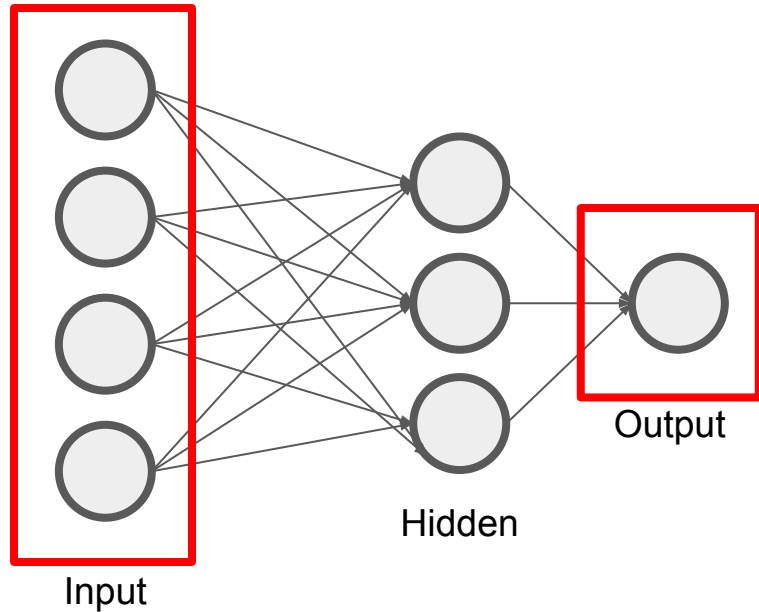
Layers

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



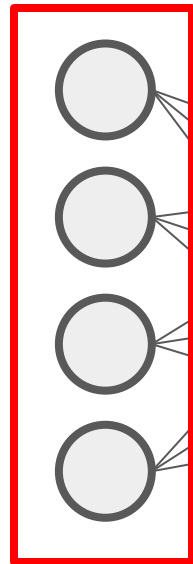
Layers

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



Layers

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



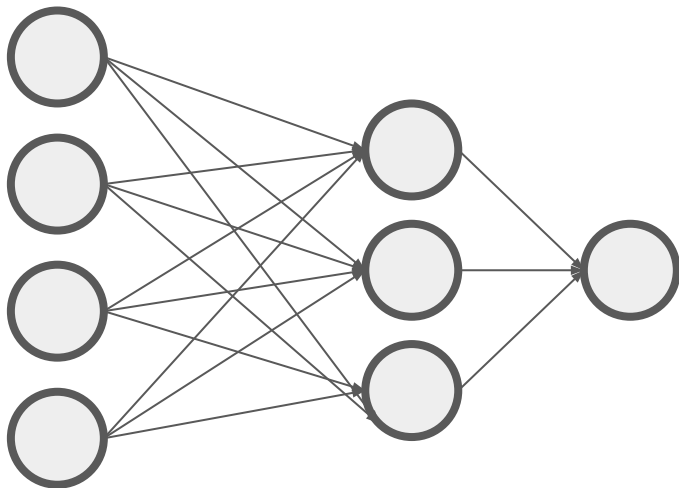
Hidden

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

Output

Math Notation

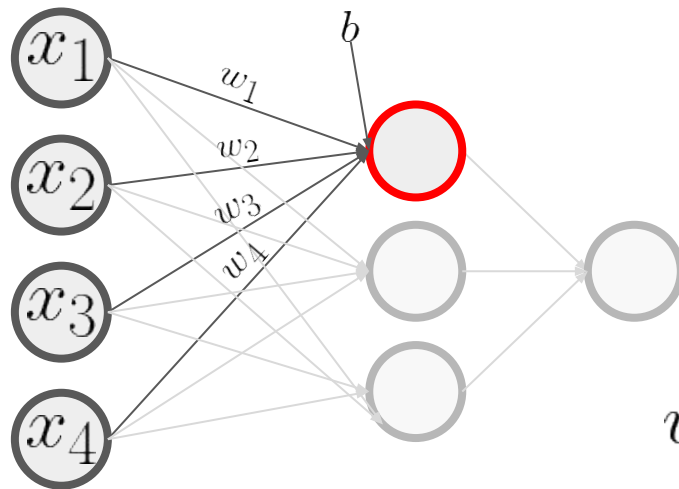
The value of a node is a **linear combination** of all the nodes in the previous layer that are connected to it.



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

Math Notation

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



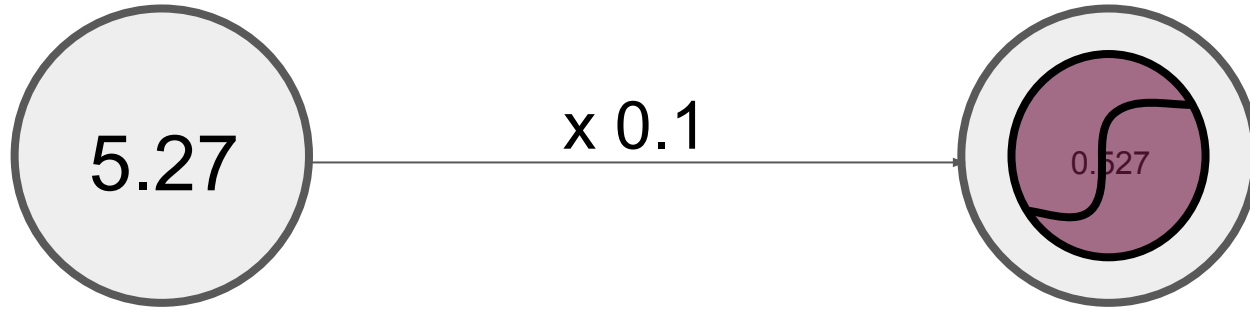
$$\mathbf{w}^T \mathbf{x} + b$$

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

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

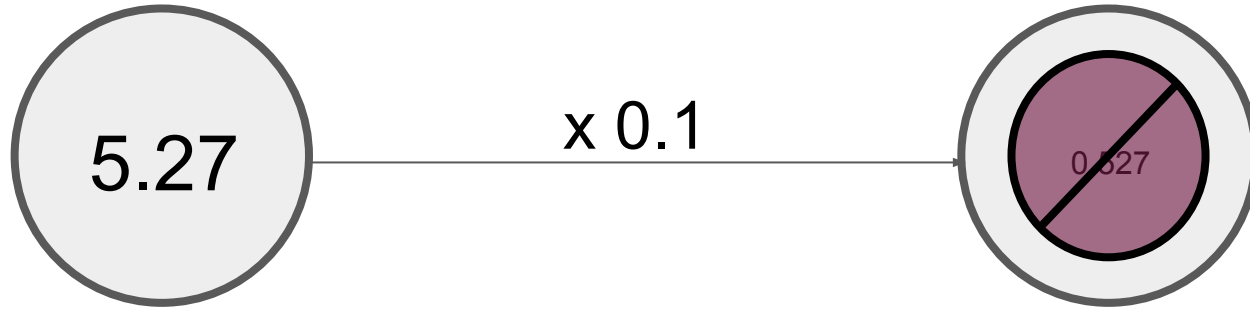
Activation Functions

Now with non-linearity!



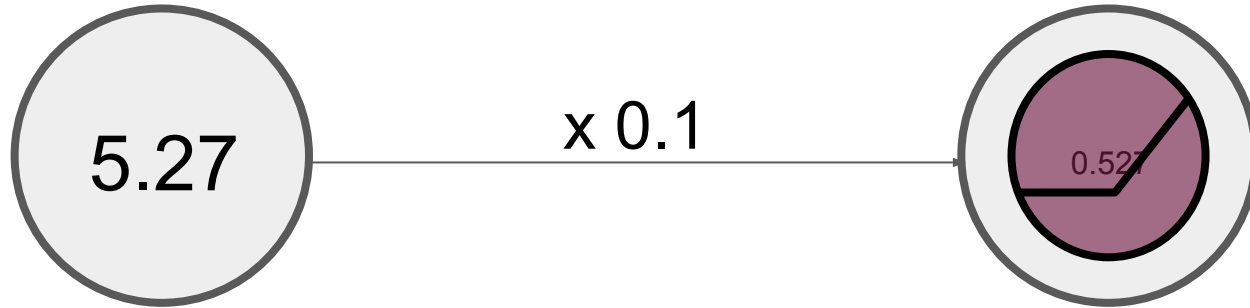
Activation Functions

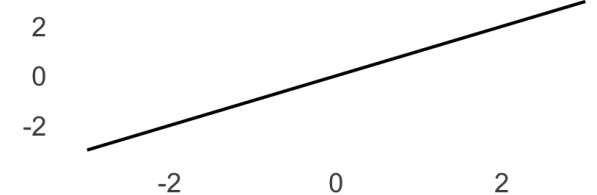
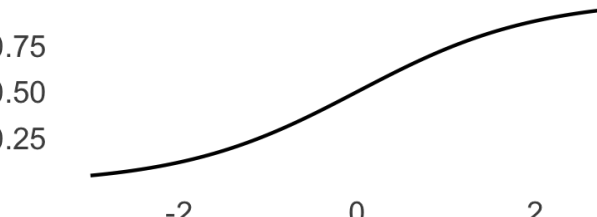
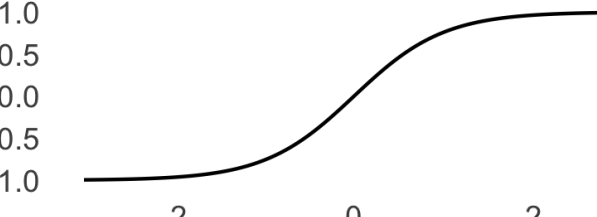
Now with non-linearity!

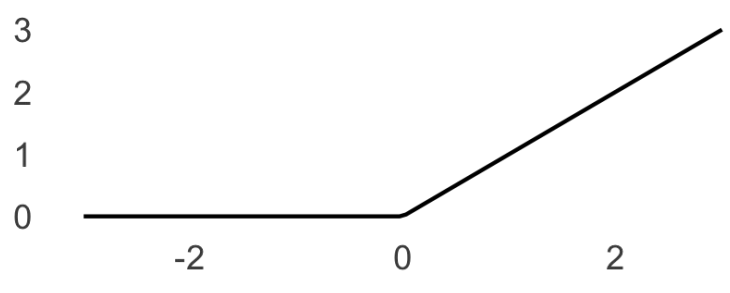
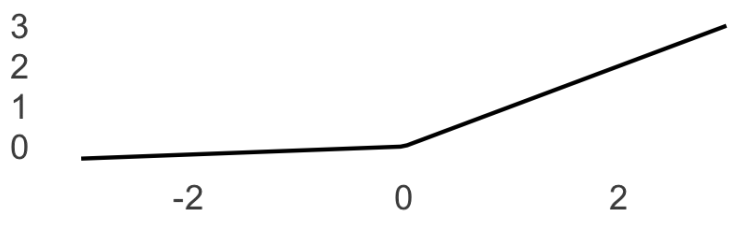


Activation Functions

Now with non-linearity!

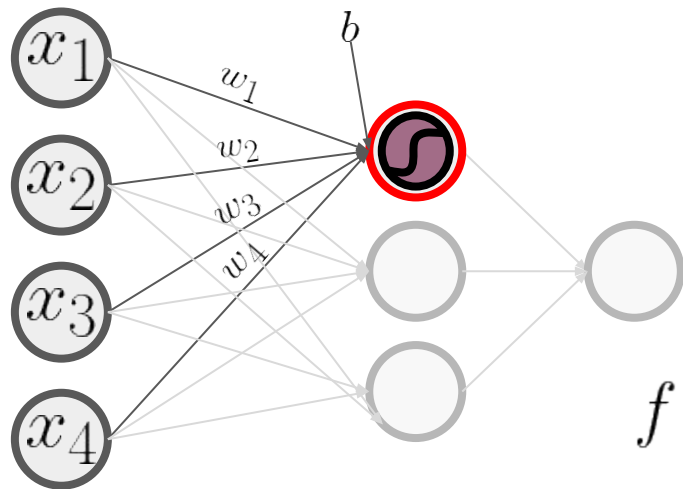


Function	f(x)	Graph
linear	$f(x) = x$	<p>Linear Activation</p>  <p>The graph shows a straight line with a slope of 1, passing through the origin (0,0). The x-axis is labeled with -2, 0, and 2. The y-axis is labeled with -2, 0, and 2.</p>
sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	<p>Sigmoid Activation</p>  <p>The graph shows an S-shaped curve that starts near 0 for negative x and approaches 1 for positive x. The x-axis is labeled with -2, 0, and 2. The y-axis is labeled with 0.25, 0.50, and 0.75.</p>
tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	<p>tanH Activation</p>  <p>The graph shows an S-shaped curve that ranges from -1 to 1, passing through the origin (0,0). The x-axis is labeled with -2, 0, and 2. The y-axis is labeled with -1.0, -0.5, 0.0, 0.5, and 1.0.</p>

Function	$f(x)$	Graph
ReLu	$f(x) = \max(0, x)$	<p>ReLU Activation</p>  <p>The graph shows the ReLU activation function. The x-axis ranges from -3 to 3, with labels at -2, 0, and 2. The y-axis ranges from 0 to 3, with labels at 0, 1, 2, and 3. The function is zero for all negative x values and increases linearly with a slope of 1 for all positive x values.</p>
Leaky ReLu	$f(x) = \max(\alpha * x, x)$	<p>Leaky ReLU Activation with $\alpha = 0.1$</p>  <p>The graph shows the Leaky ReLU activation function with $\alpha = 0.1$. The x-axis ranges from -3 to 3, with labels at -2, 0, and 2. The y-axis ranges from 0 to 3, with labels at 0, 1, 2, and 3. The function is zero for all negative x values and increases linearly with a slope of 1 for all positive x values. For negative x values, the function increases linearly with a slope of 0.1.</p>

Math Notation

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



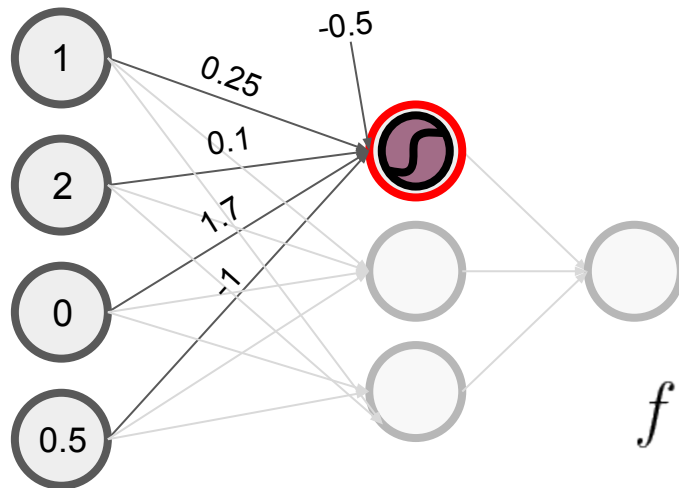
$$f(\mathbf{w}^T \mathbf{x} + b)$$

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

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

Math Notation

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



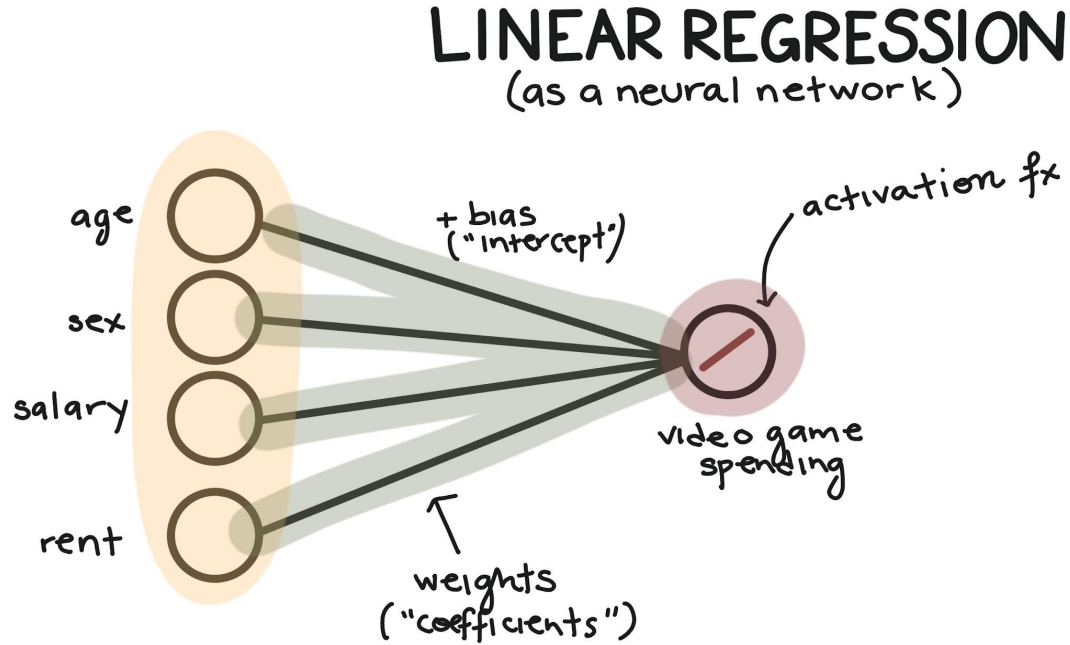
$$f(\mathbf{w}^T \mathbf{x} + b)$$

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

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

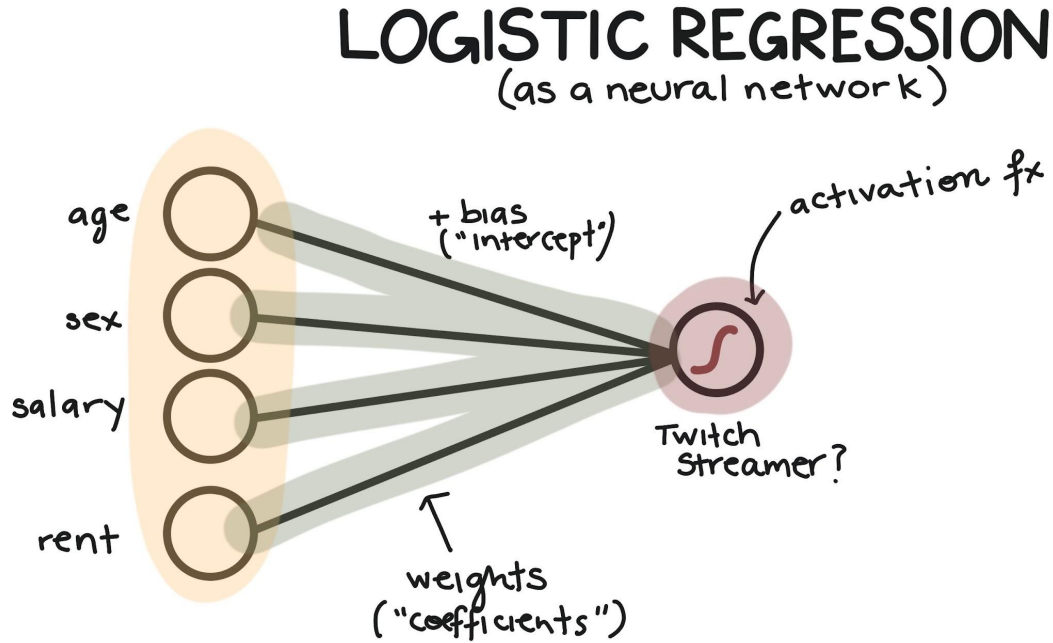
Familiar Models as Neural Networks

Linear Regression as a NN



$$\text{LOSS: } \sum (x_i - \hat{x})^2$$

Logistic Regression as a NN



$$\text{Loss: } \sum -y_i \log(\hat{p}_i) - (1 - y_i) \log(1 - \hat{p}_i)$$

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