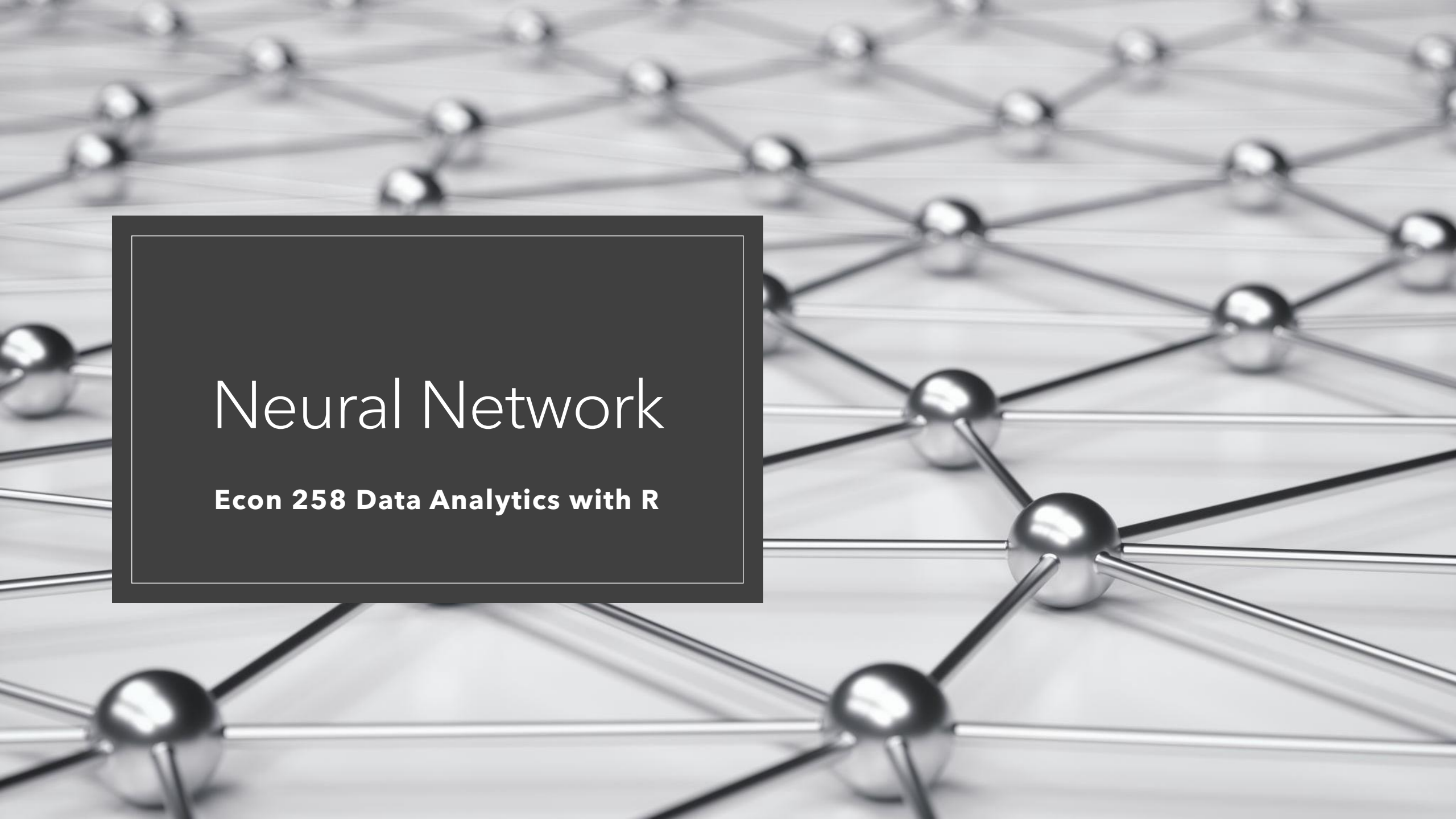


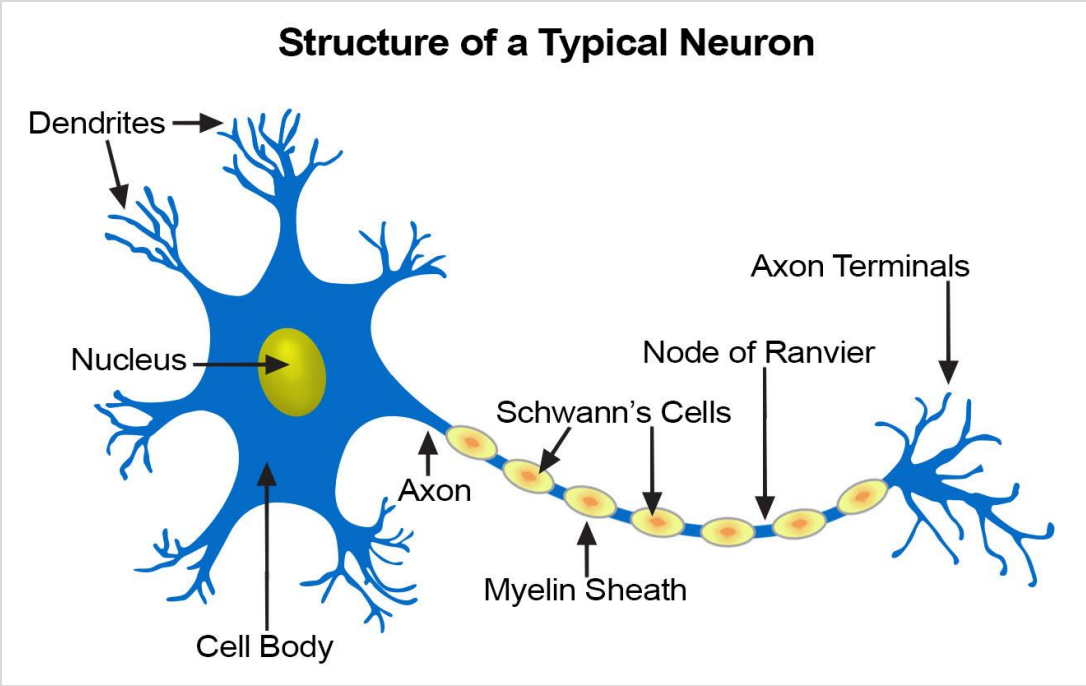
Neural Network

Econ 258 Data Analytics with R

A SUMMARY SO FAR

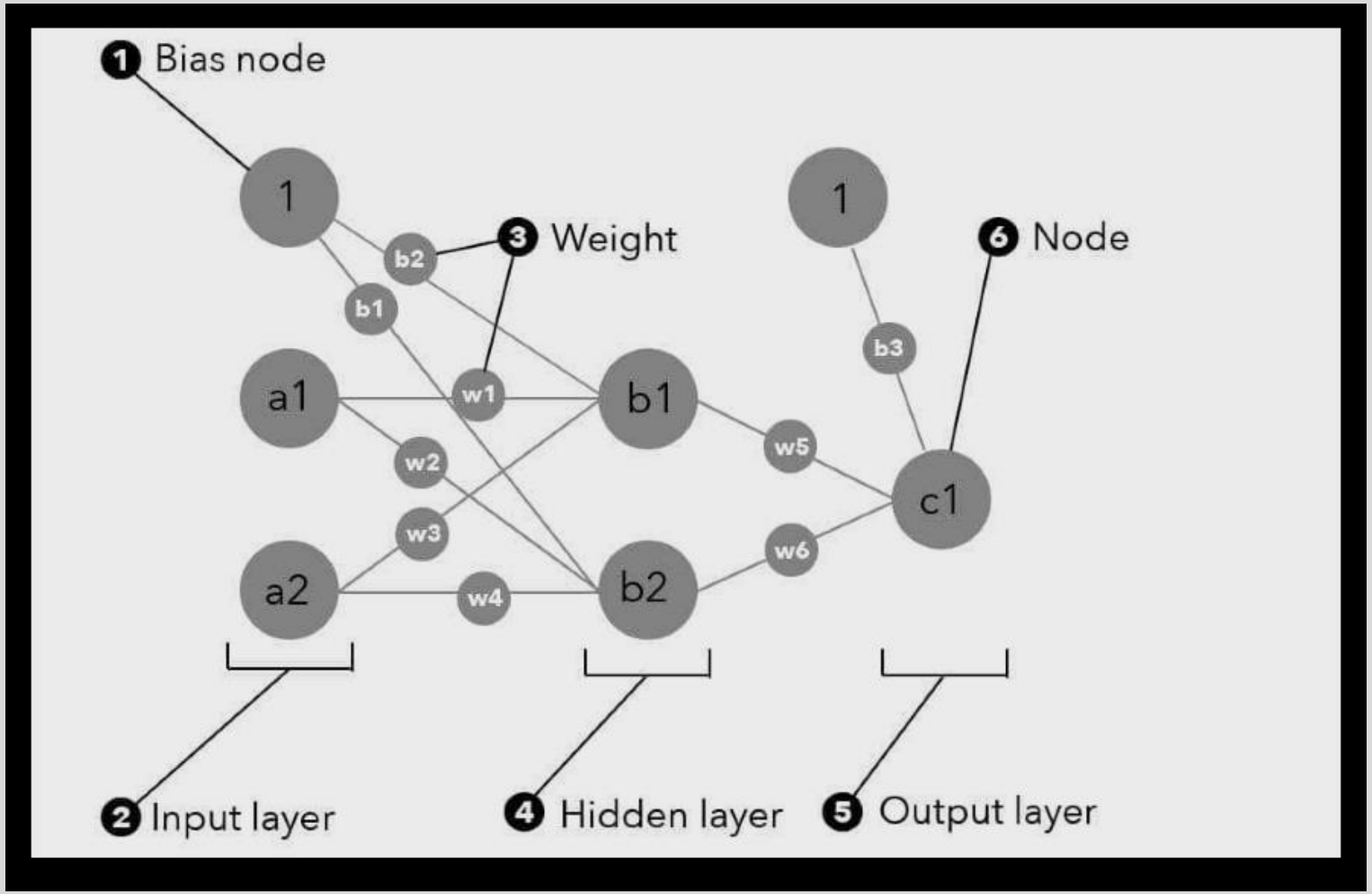
TYPE	NAME	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Linear	Linear regression	**The "best fit" line** through all data points. Predictions are numerical.	Easy to understand — you clearly see what the biggest drivers of the model are.	✗ Sometimes too simple to capture complex relationships between variables. ✗ Tendency for the model to "overfit".
	Logistic regression	The adaptation of **linear regression** to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	✗ Sometimes too simple to capture complex relationships between variables. ✗ Tendency for the model to "overfit".
Tree-based	Decision tree	A graph that uses a **branching method** to match all possible outcomes of a decision.	Easy to understand and implement.	✗ Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.
	Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but **by combining them we get better overall performance.**	A sort of "wisdom of the crowd". Tends to result in very high quality models. Fast to train.	✗ Can be slow to output predictions relative to other algorithms. ✗ Not easy to understand predictions.
	Gradient Boosting	Uses even weaker decision trees, that are increasingly **focused on "hard" examples.**	High-performing.	✗ A small change in the feature set or training set can create radical changes in the model. ✗ Not easy to understand predictions.
Neural networks	Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several **layers of neural networks put one after the other.**	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	✗ Very, very slow to train, because they have so many layers. Require a lot of power. ✗ Almost impossible to understand predictions.

The Way Our Brain Learns



(1) The brain learns by processing all inputs
(2) Neurons connect with each other to process information
(3) Get an output/outcome after processing information
(4) Evaluate whether output contains much error
(5) Repeat until get it right.

Terminologies and Intuition



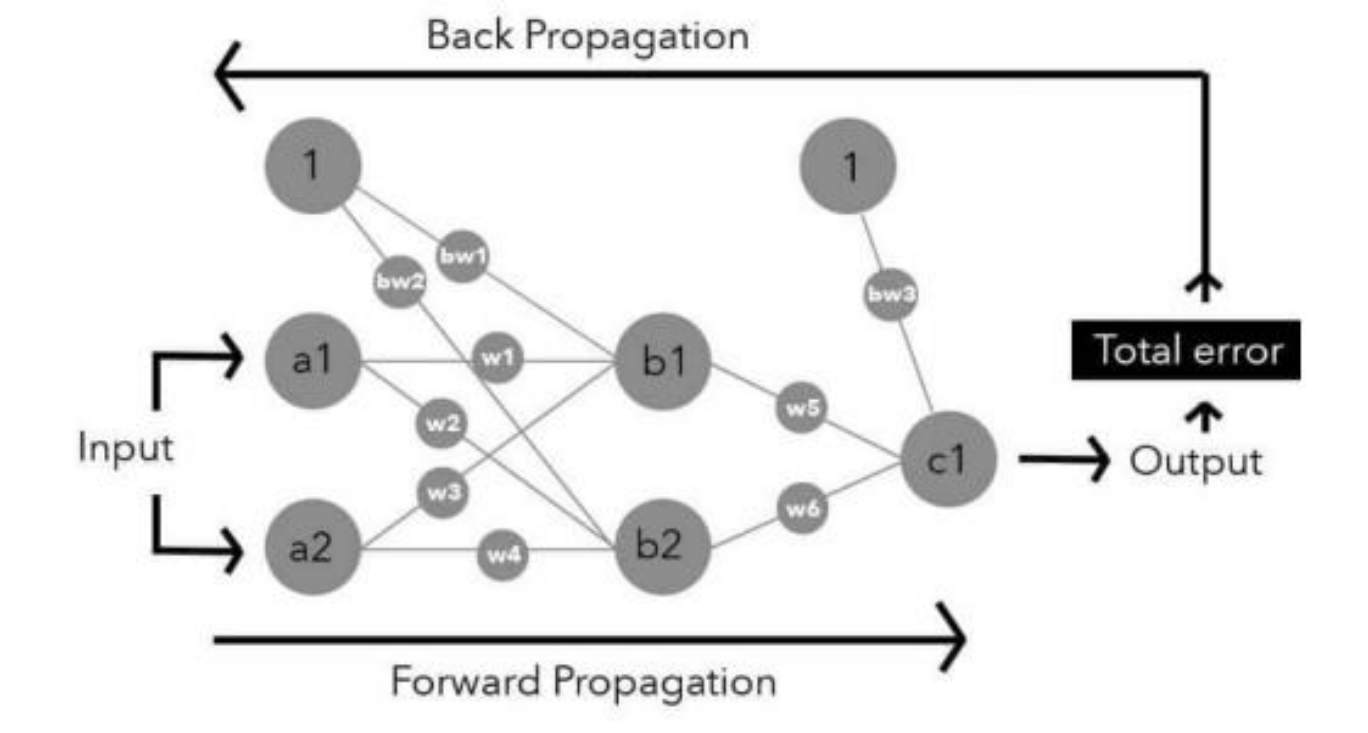
Very simple intuition using the toddler toy example:

- A block has inputs: color, number of pointed edges, thickness, number of round edges.

- Toddler has some sort of bias and weights of how important each of the inputs are.

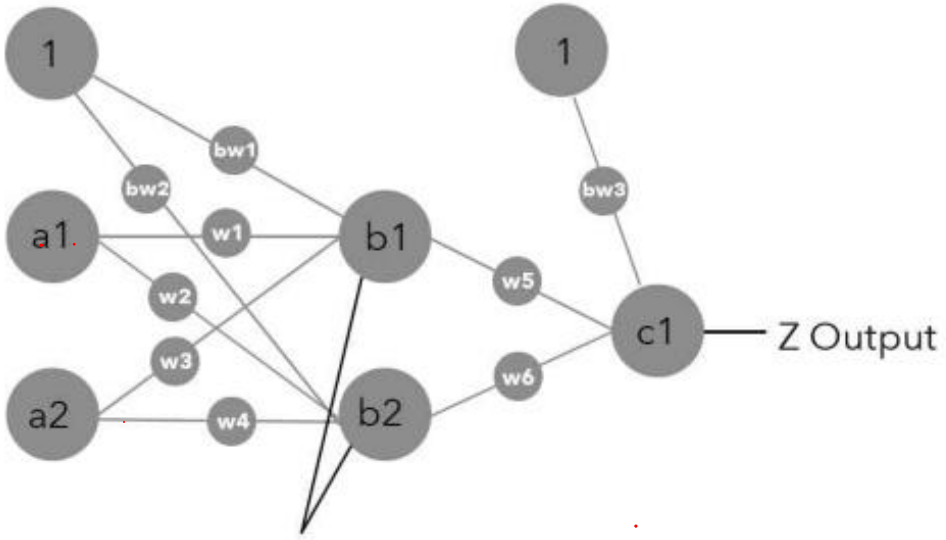
- In the hidden layer, information on inputs are weighted and combined, and moved through the network, called Forward Propagation.

Learning Through Trial and Error



- After going through all hidden layers, information goes through an **Activation Function** that determines output.
- Toddler calculates how much error was made, this block did not fit here but there, another one fits there.
- Based on that error (MSE or RSS or other), go back to the beginning (back propagation) and change weights.

Nodes Accept a Single Number



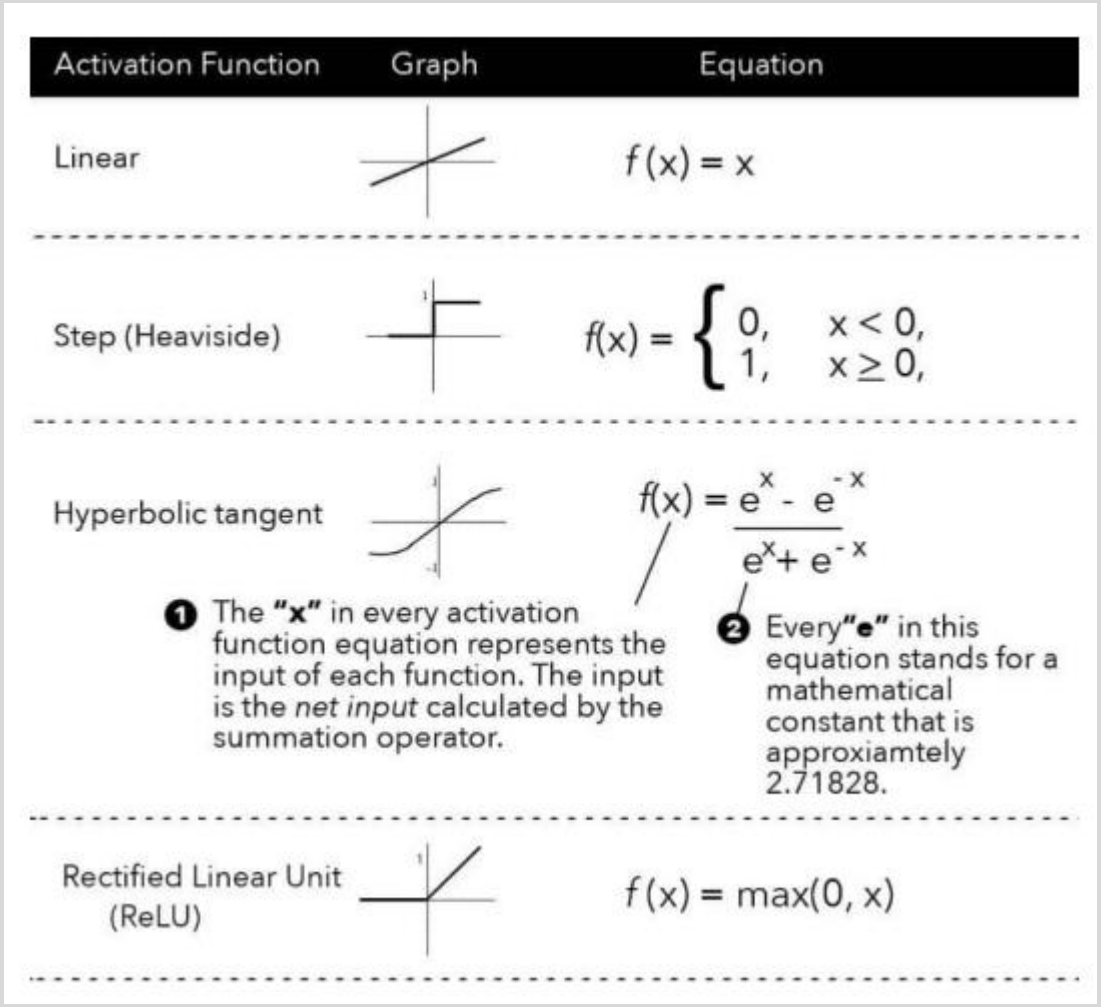
❶ To calculate the net input of **b1** and **b2**, we need to multiply **a1** and **a2** by their respective weights and then sum the answers into **b1** and **b2**, respectively. The bias also needs to be added.

❶ The net input to node b2 (**b2net**), is calculated by multiplying the circled elements in the vector and matrix. The bias is also added.

$$x_{in} = \begin{bmatrix} w1 & w3 \\ w2 & w4 \end{bmatrix} \cdot \begin{bmatrix} a1 \\ a2 \end{bmatrix} + B = \begin{bmatrix} b1net \\ b2net \end{bmatrix}$$

Activation Function

- The activation function receives the final information processed in the hidden layer and assign it to outputs.
- Can choose from a number of activation functions.

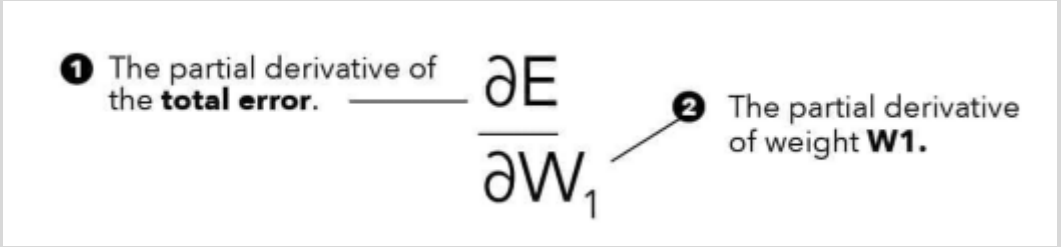


Tweaking the Weights

How are weights tweaked?

- The starting weights are random.
- Weights are re-chosen based on partial derivatives. Remember calculus!
- All this reweighting makes neural network takes a lot of time when you have a lot of information.

For example, just for W1.



The derivative can be more complicated in more forward nodes.

Pros and Cons

- Pro: allows for a lot of complexity
- Cons: Very hard to interpret what is going on

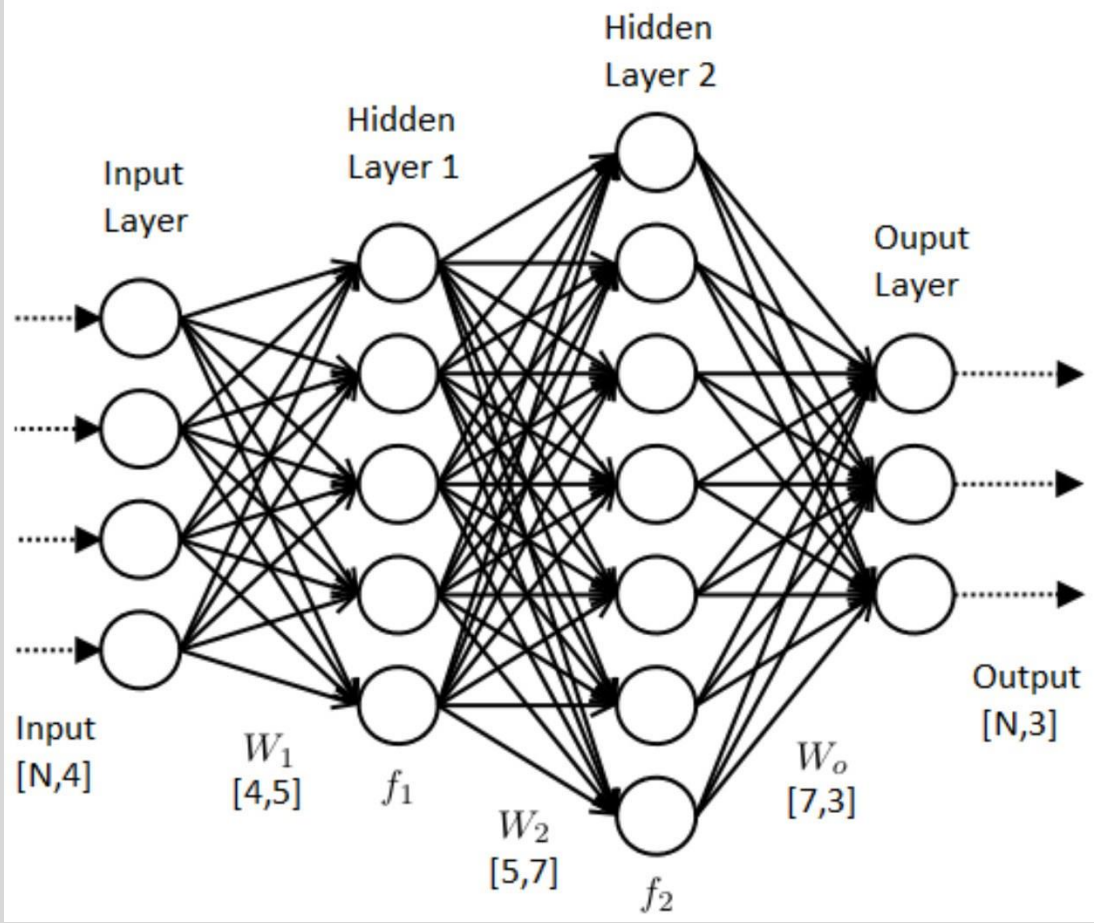
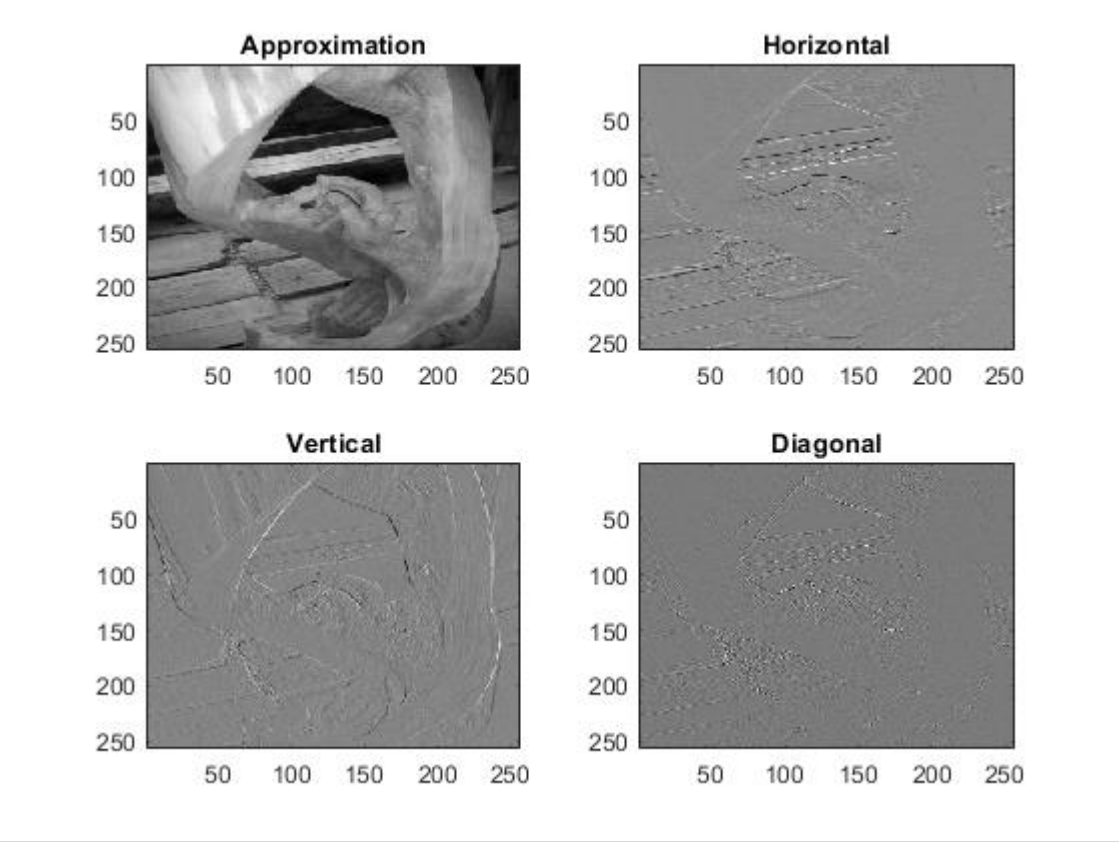


Image Recognition with Data

- How does an algorithm recognize image patterns?
  - Need to turn image into data points.
  - One way: if you have an image with 20 x 20 pixels, each pixel (400 of them total) can be recognized by a color scheme ranging from 0 to 1 for example.
  - Another way, wavelet image transformation.

Will practice an exercise an algorithm to recognize counterfeit money.



Coding Exercise

- Example 1 Predict different types of Irises

- Example 2 (Exercise, Somewhat Related to Econ): Predict counterfeit money through images

Reference

- A very good intuitive guide for neural networks for beginners:

"Machine Learning with Neural Networks: An In-depth Visual Introduction with Python" 2017 by Michael Taylor.

- Projects taken from:

"R Projects for Dummies" (2018) by Joseph Schmuller.