https://ttpoll.com/p/965065

COSC 325: Introduction to Machine Learning

Dr. Hector Santos-Villalobos

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Lecture 21: Artificial Neural Networks and Deep Learning

THE UNIVERSITY OF TENNESSEE
KNOXVILLE

Class Announcements

Homework

Almost done with all the homework!!!!!!!

Course Project:

- Midterm report grades will be available 11/17
- Course Project Presentation Poster Logistics

Quizzes:

Weekly quiz as usual.

Exams:

Exam #2 is **next Thursday**, 11/21—same format.

Lectures:

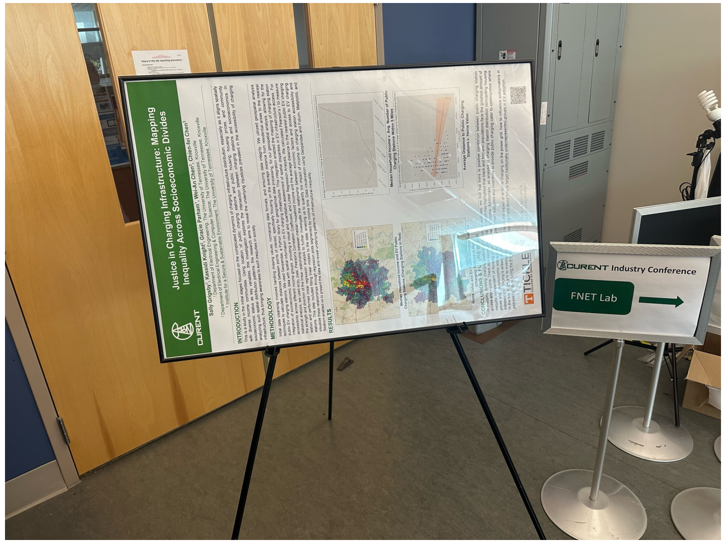
- **Last 15 Minutes:** Tenure Teaching Evaluation
- **Panel on Ethical AI 11/26**. You will get attendance points by posting a question in the Discord ***#panel-on-ethical-ai*** channel (https://discord.com/channels/1263144544082596050/1306342338926346260)

https://ttpoll.com/p/965065

Poster Session Logistics

- 75 minutes lecture
  - Session 1 (16 teams)
    - 10 mins setup
    - 25 mins poster session
  - Session 2 (15 teams)
    - 10 mins setup
    - 25 mins poster session
  - Clean up
    - Last 5 minutes

- Peer Reviews
  - Students presenting in Session 1 will review projects in Session 2 and vice versa.
  - You will be assigned three projects to review
    - Spend 5-7 mins per project
  - Check Canvas Quiz **CP Presentation Scoring Sheet (DRAFT)** https://utk.instructure.com/courses/206990/quizzes/439418

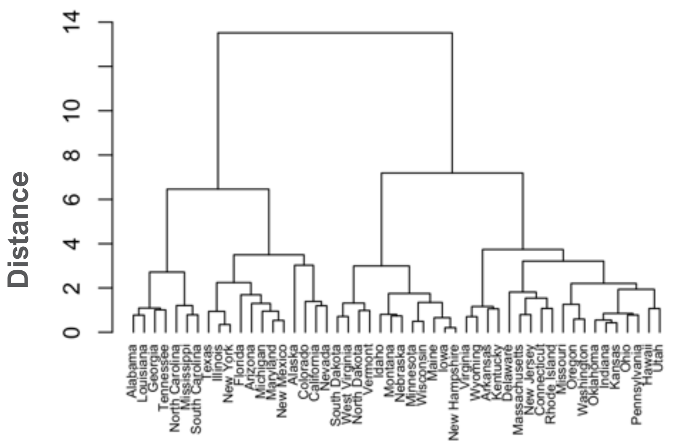


THE UNIVERSITY OF TENNESSEE KNOXVILLE

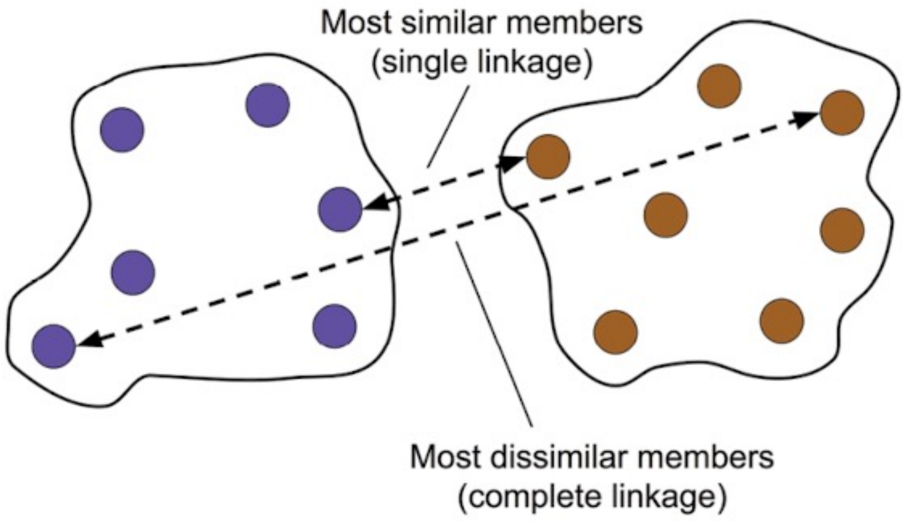
Review

- Hierarchical Clustering
  - Bottom-up approach: Agglomerative Complete Linkage
  - Dendrograms

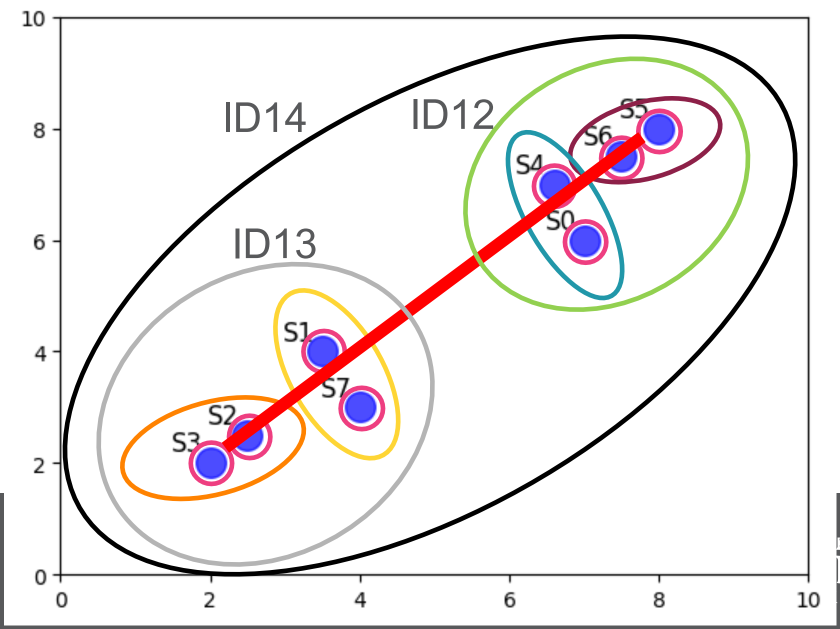
USA Arrests Cluster Dendrogram



https://www.kaggle.com/datasets/halimedogan/usarrests

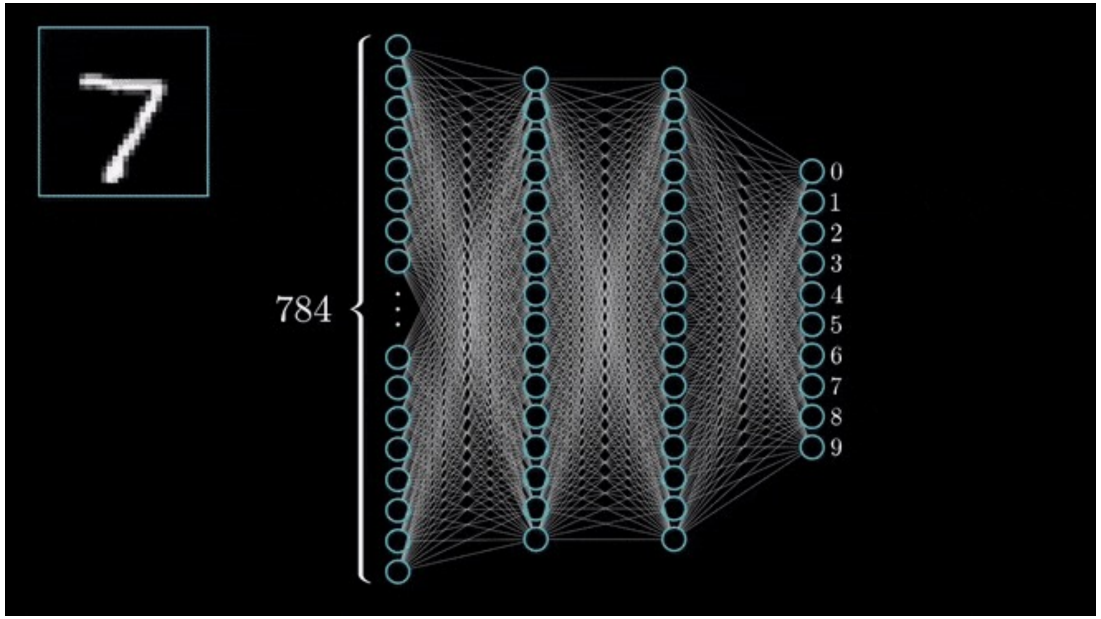


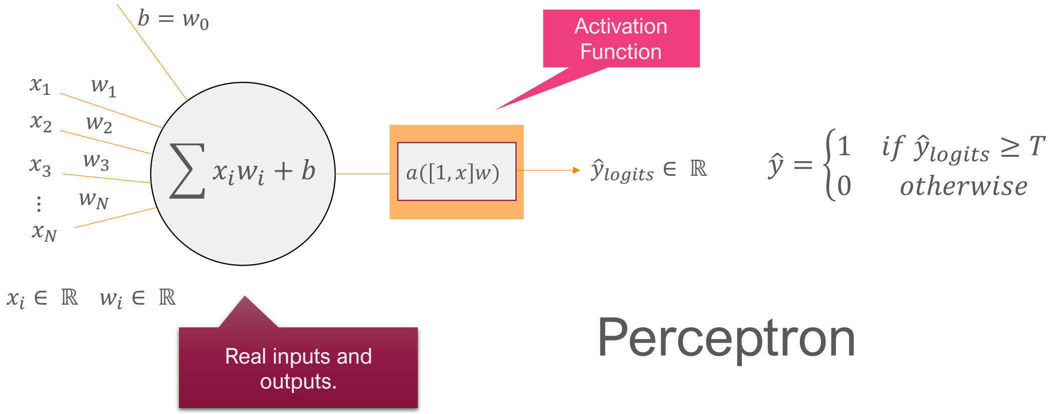
Raschka, et. al., “Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python”



Review

- Neural Networks
  - Universal function approximator
  - Learning with backpropagation
- The perceptron (artificial neuron)
- By the end of the lecture, we started discussing ***Connectionism*** and ***Connectionist Machines***.



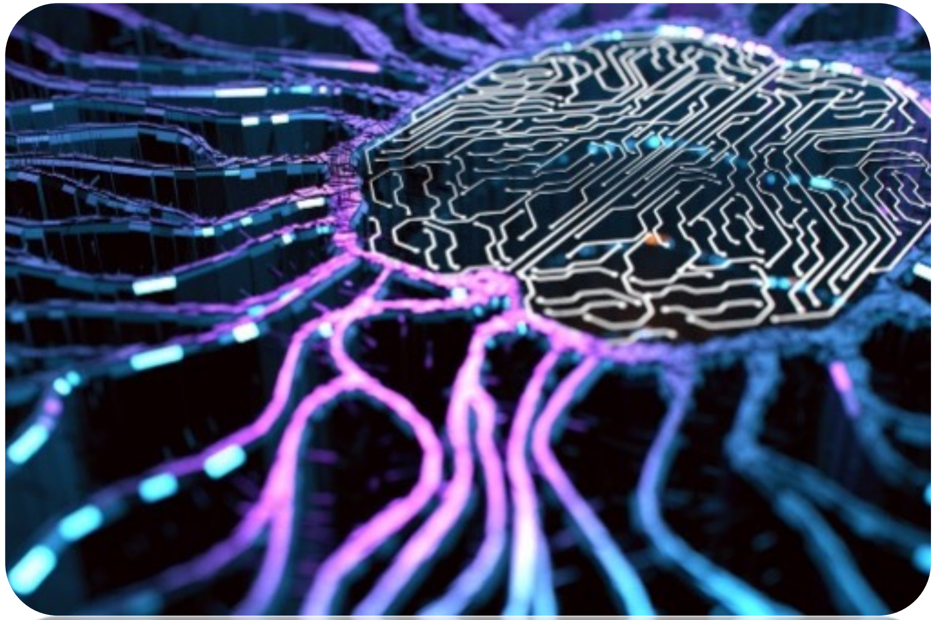


Perceptron

Today’s Topics

Artificial Neural Networks

*Deep Learning**



Touching a flame.

Response at scale.

Mid 1800s: The brain is comprised of interconnected neurons.
~100 Trillion Connections

Emotions
Movement

Connectionism (1873)

- Alexander Bain: philosopher, psychologist, mathematician, logician, linguist, professor
- Main ideas in the book “Mind and Body”
  - Neural groupings
    - Neurons excite and stimulate each other
    - Different input combinations can result in different outputs
    - Activation intensity influences the activation of connected neurons
  - Making memories
    - Neurons connections strengthen with repetitive inputs (Before Hebb’s Law 1949)

Hebb's Law: Model for Neural Plasticity

- Novelist, schoolteacher, psychologist
- Main idea in book "The Organization of Behavior" (1949):
  - If neuron A repeatedly triggers neuron B, the synapses connecting these neurons get larger.
  - Hebb's Law: "Neurons that fire together wire together."

Response of a neuron x_j

$$x_j = f\left(\sum_{i=1}^{N} w_i x_i\right)$$

Weights can be different now.

Evolution of synaptic weight

$$\Delta w_{ij} = \eta x_i x_j$$

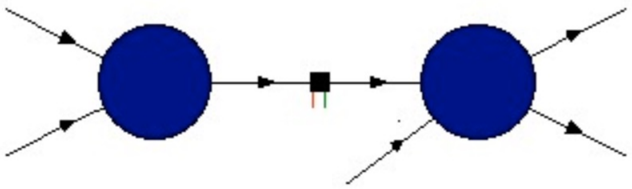
Weight Update

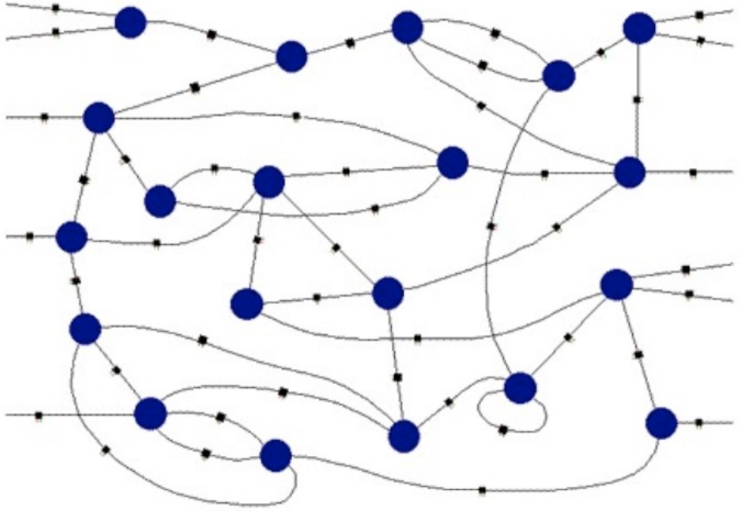
Learning Rate

Interaction between neurons i and j

Connectionist Machines

- Multiple connectionist paradigms proposed
  - Alan Turing’s Connectionist model (1948):
  - Parallel Distributed Processing (1986)
    - Rumelhart, Hinton, McClelland
  - Requirements of a connectionist system
    - Bechtel and Abrahamson (1991)
- Main properties
  - Network of processing elements
  - All world knowledge is stored in the connections between the elements

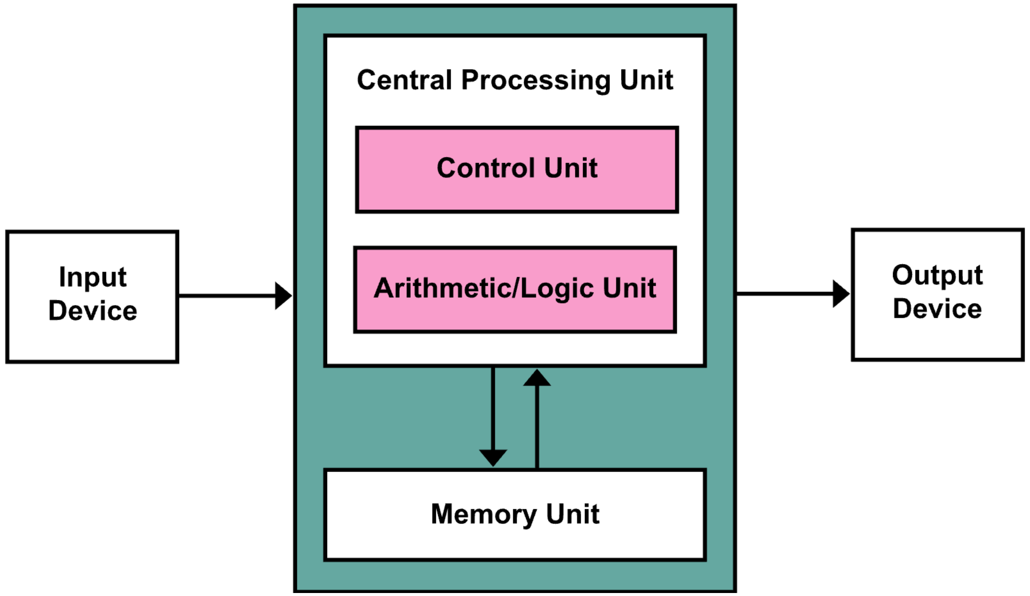




Source: Carnegie Mellon Univeristy, Bhiksha Raj and Rita Singh, 11-785 Introduction to Deep Learning

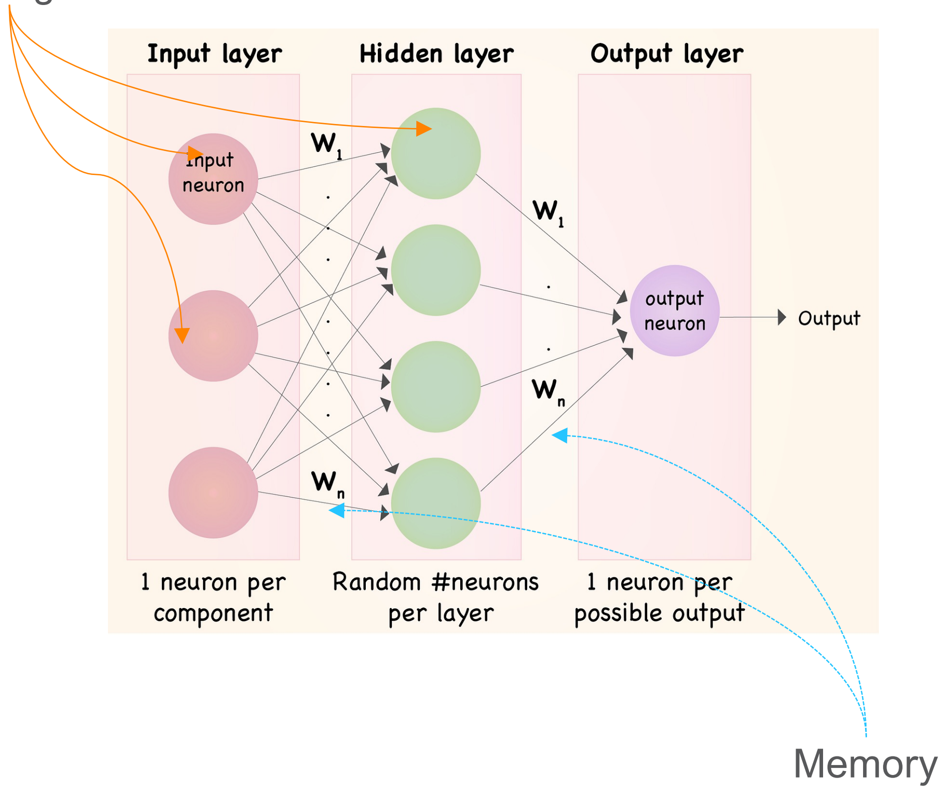
Von Neumann vs Connectionist Machines

Von Neumann Machines



The typical modern computer!

Connectionist Machines



Wikipedia is the source of the Von Neumann Machine diagram.

Pop Quiz

A neural network is a **Von Neumann Machine** because it is a network of processing elements, and all world knowledge is stored in the connections between the elements.

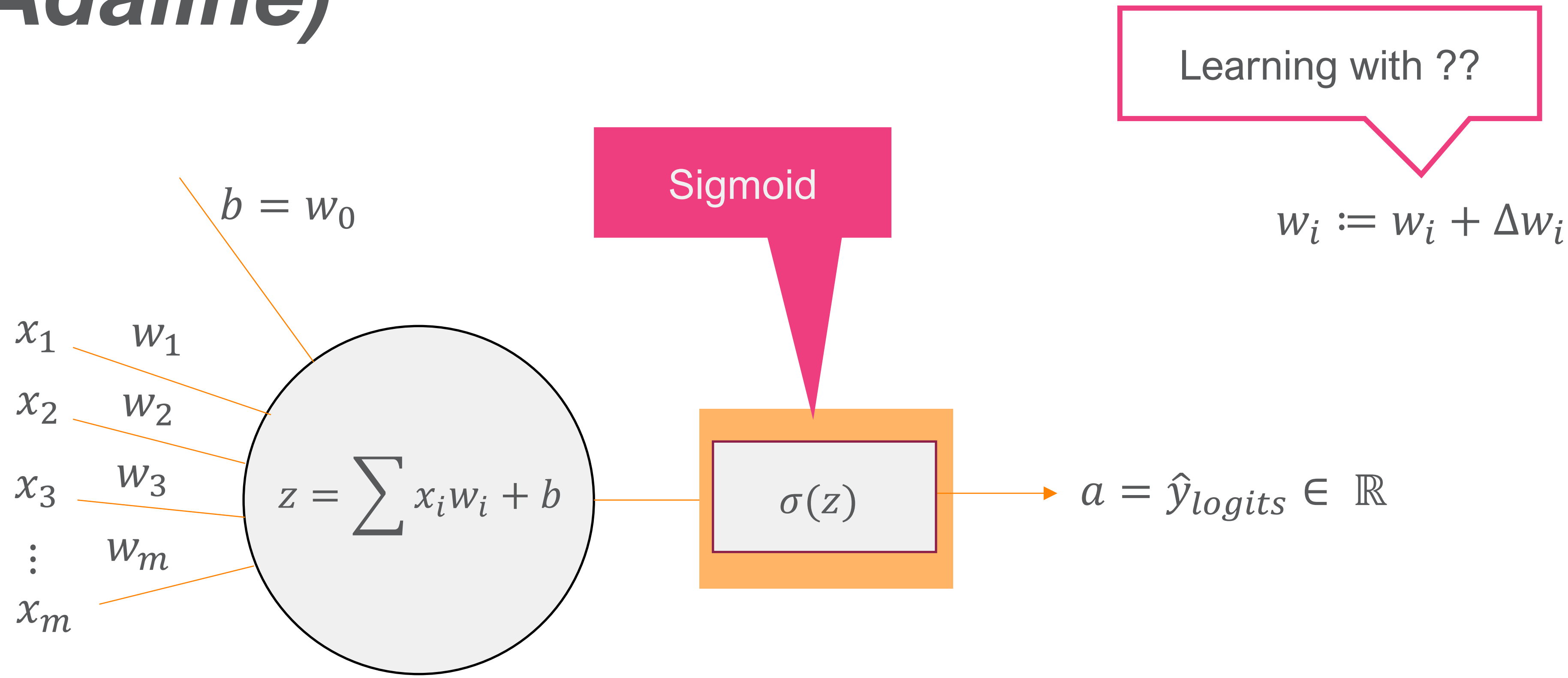
A. True

B. False

Math of a Neural Network

THE UNIVERSITY OF TENNESSEE KNOXVILLE

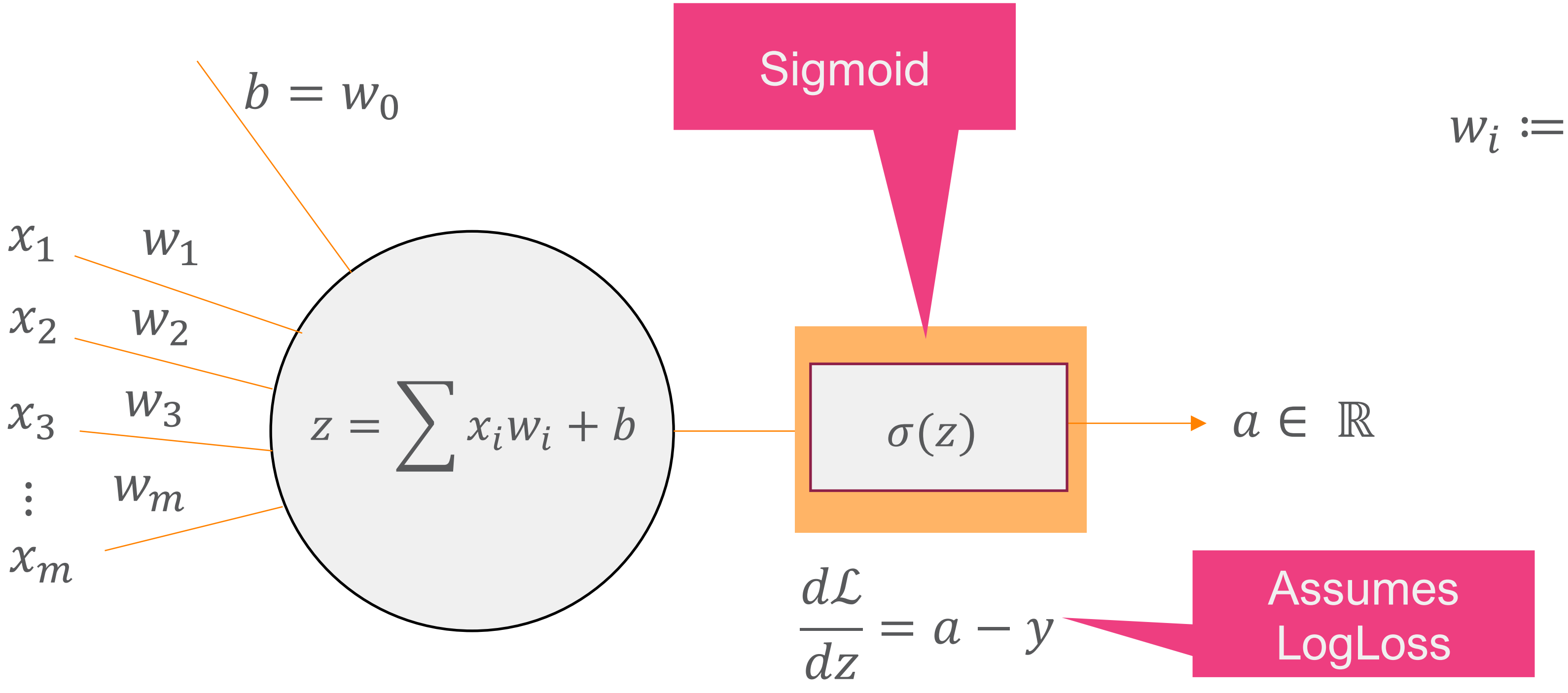
Modern Perceptron: *Adaptive Linear Neuron (Adaline)*



Modern Perceptron: *Adaptive Linear Neuron (Adaline)*

Learning with Gradient Descent

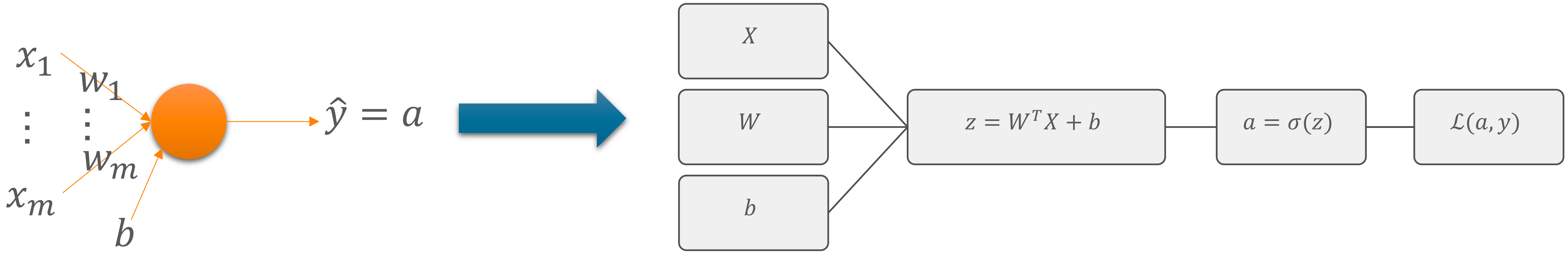
$$w_i := w_i + \Delta w_i \,, \quad \Delta w_i = -\lambda \frac{\partial L}{\partial w_i}$$

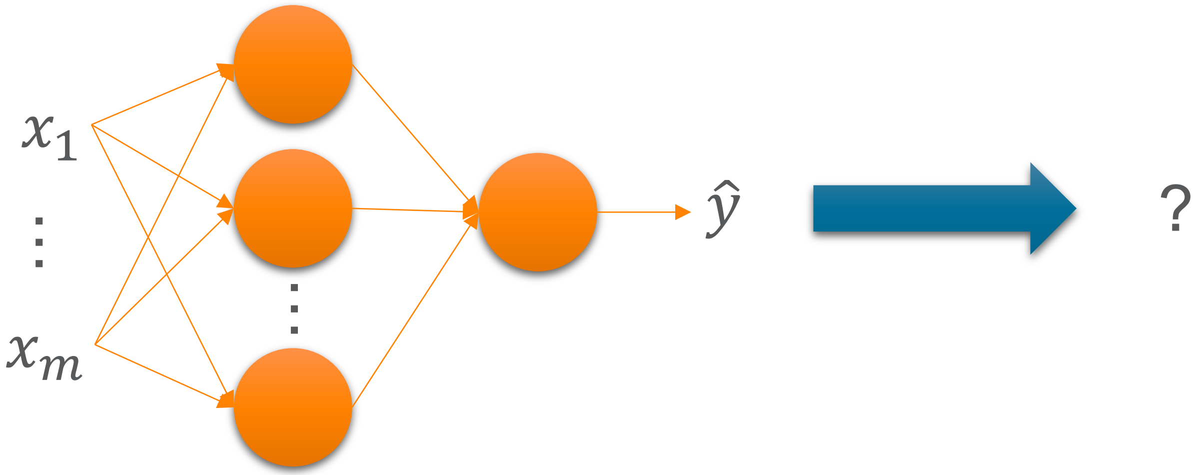


$$\frac{d\mathcal{L}}{dw_1} = (a-y)x_1 \quad \frac{d\mathcal{L}}{dw_2} = (a-y)x_2 \quad \frac{d\mathcal{L}}{db} = (a-y)$$

Check Lecture 8

Neural Network

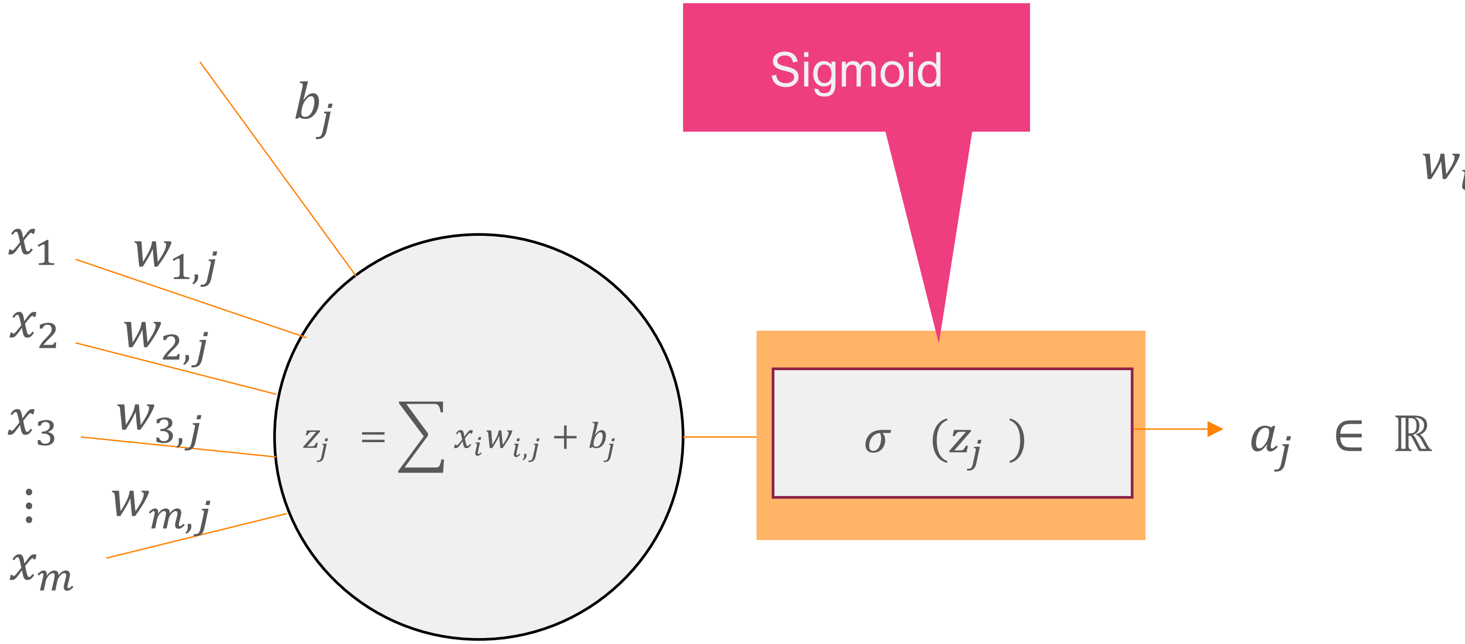




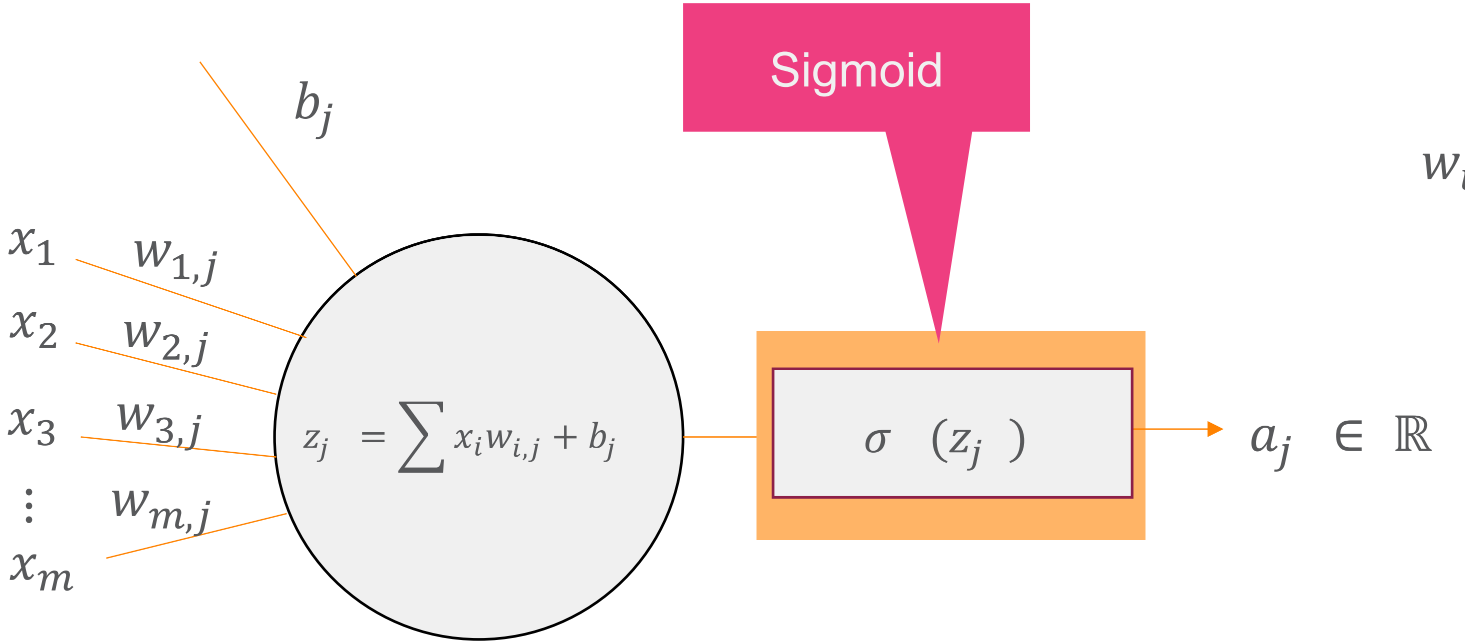
Perceptron Layer

Learning with Gradient Descent

$$w_{i,j} := w_{i,j} + \Delta w_{i,j}, \qquad \Delta w_{i,j} = -\lambda \frac{\partial L}{\partial w_{i,j}}$$



Perceptron Layer (2)

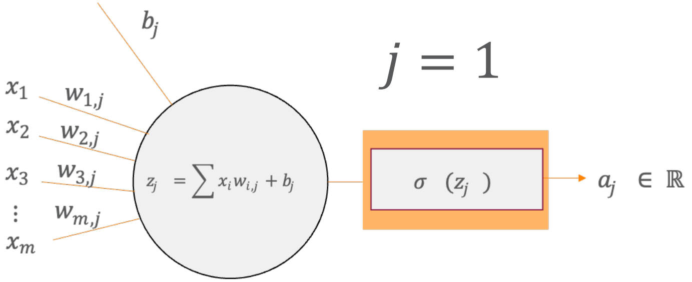


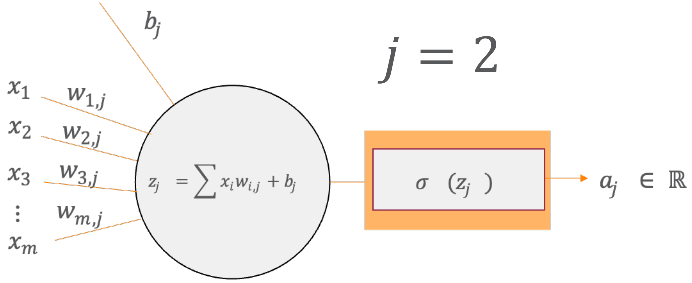
Learning with Gradient Descent

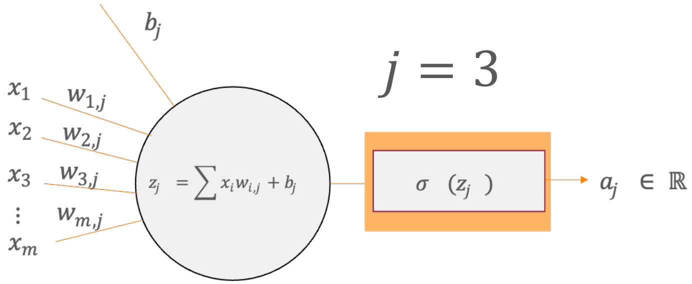
$$w_{i,j} := w_{i,j} + \Delta w_{i,j}, \qquad \Delta w_{i,j} = -\lambda \frac{\partial L}{\partial w_{i,j}}$$

We can define: $W_j = \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{m,j} \end{bmatrix} \Rightarrow z_j = XW_j + b_j$

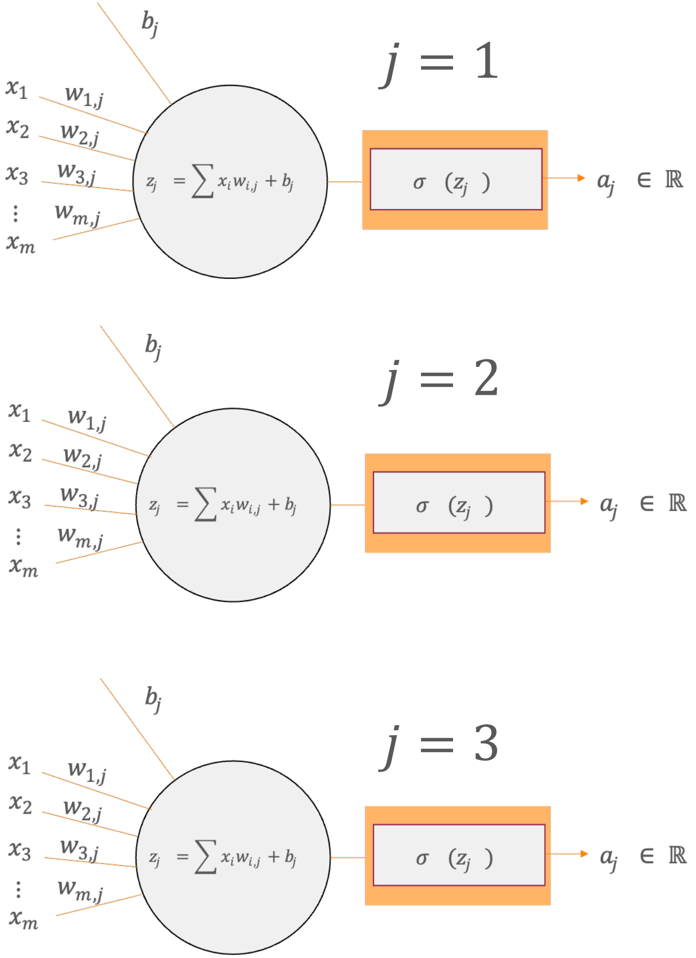
Perceptron Layer (3)







Perceptron Layer (4)



X size (n, m)

W_1 size $(m, 1)$

$$a_1 = \sigma(XW_1 + b_1)$$

a_1 Size $(n, 1)$

Scalar (Addition by Broadcasting)

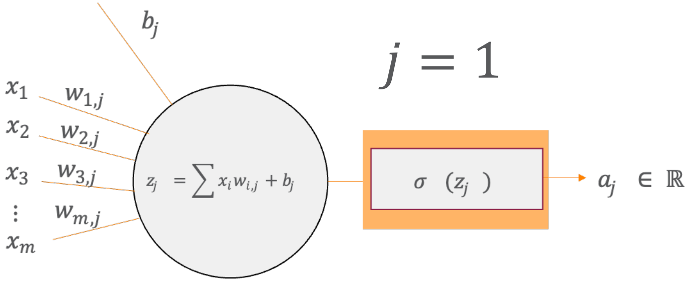
$$a_2 = \sigma(XW_2 + b_2)$$

a_2 Size $(n, 1)$

$$a_3 = \sigma(XW_3 + b_3)$$

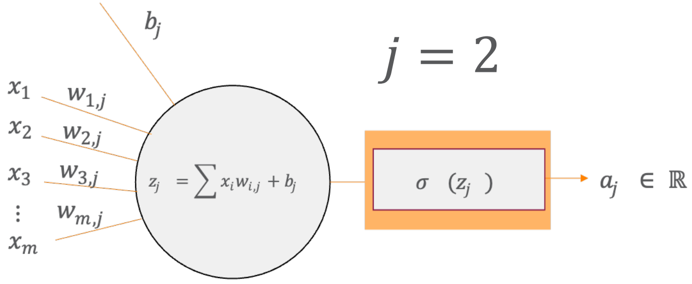
a_3 Size $(n, 1)$

Perceptron Layer (5)



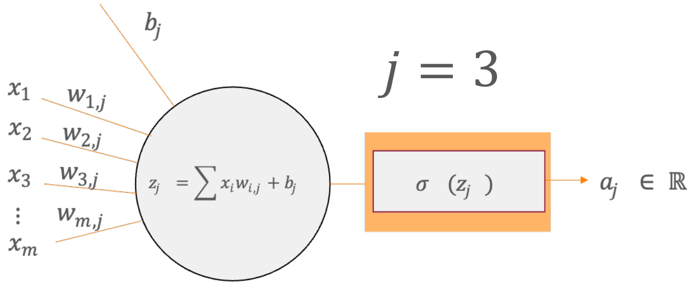
$$a_1 = \sigma(XW_1 + b_1)$$

a_1
Size $(n, 1)$



$$a_2 = \sigma(XW_2 + b_2)$$

a_2
Size $(n, 1)$



$$a_3 = \sigma(XW_3 + b_3)$$

a_3
Size $(n, 1)$

$[a_1, a_2, a_3]$
Size $(n, 3)$

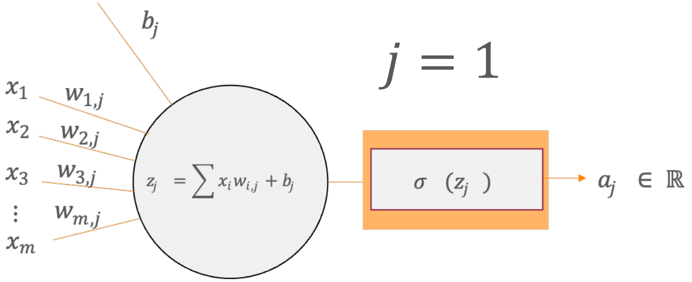
Layer output is a matrix of size (n, d), where d is the number of neurons in the layer.

$$[a_1 \quad a_2 \quad a_3] =$$
$$= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)]$$
$$= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3])$$

Activation function is an element-wise operation.

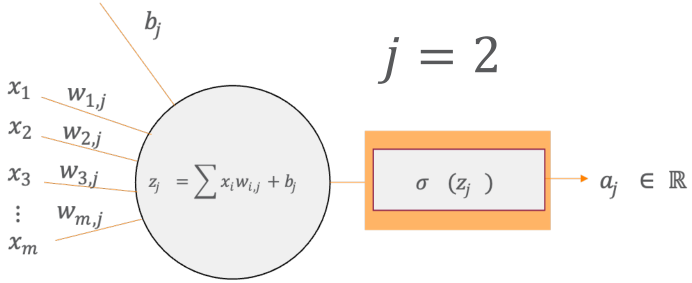
Perceptron Layer (5)

$j = 1$



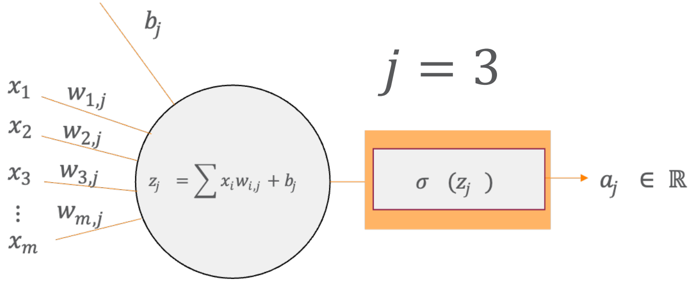
$$a_1 = \sigma(XW_1 + b_1)$$

$j = 2$



$$a_2 = \sigma(XW_2 + b_2)$$

$j = 3$



$$a_3 = \sigma(XW_3 + b_3)$$

$[a_1, a_2, a_3]$
Size $(n, 3)$

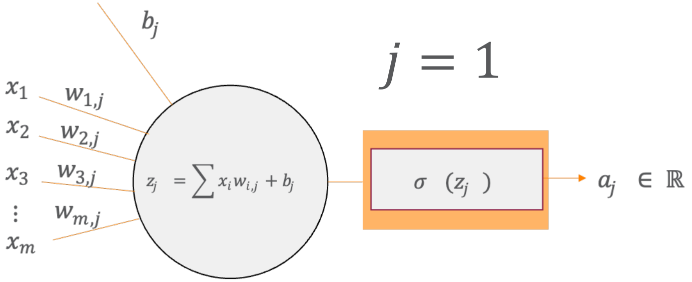
$$\begin{aligned}[a_1 \quad a_2 \quad a_3] &= \\ &= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)] \\ &= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3]) \\ &= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])\end{aligned}$$

X size
(n, m)

W_1, W_2, and W_3
Size $(m, 1)$

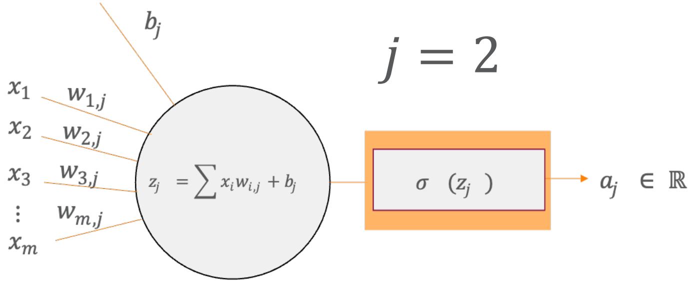
Perceptron Layer (5)

$j = 1$



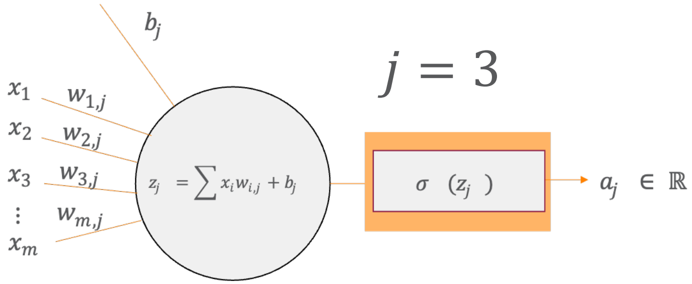
$$a_1 = \sigma(XW_1 + b_1)$$

$j = 2$



$$a_2 = \sigma(XW_2 + b_2)$$

$j = 3$



$$a_3 = \sigma(XW_3 + b_3)$$

$[a_1, a_2, a_3]$
Size $(n, 3)$

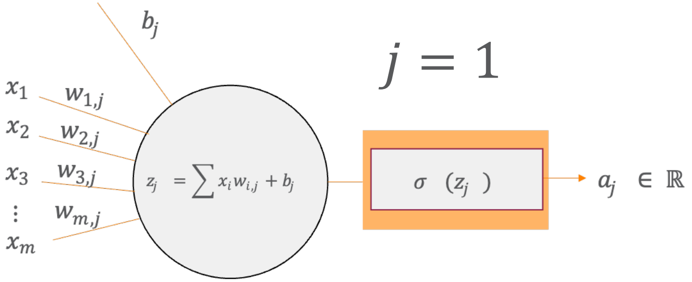
$$[a_1 \quad a_2 \quad a_3] =$$
$$= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)]$$
$$= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3])$$
$$= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])$$

X size
(n, m)

$[W_1, W_2, W_3]$
Size $(m, 3)$

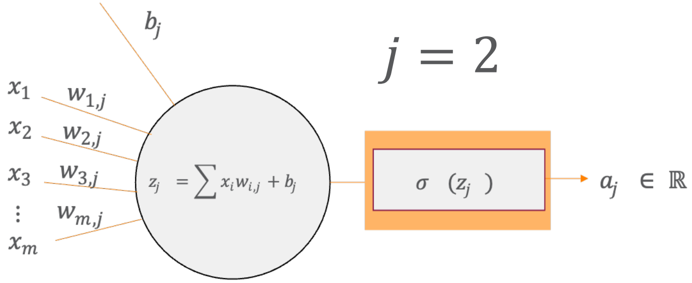
Perceptron Layer (5)

$j = 1$



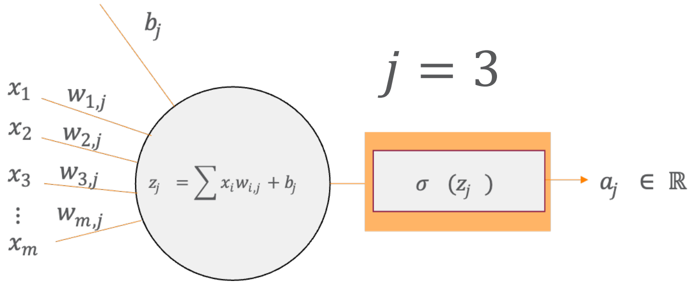
$$a_1 = \sigma(XW_1 + b_1)$$

$j = 2$



$$a_2 = \sigma(XW_2 + b_2)$$

$j = 3$



$$a_3 = \sigma(XW_3 + b_3)$$

$[a_1, a_2, a_3]$
Size $(n, 3)$

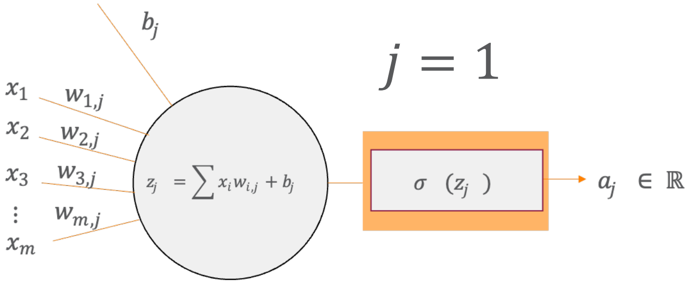
$$\begin{aligned}
[a_1 \quad a_2 \quad a_3] &= \\
&= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)] \\
&= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3]) \\
&= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])
\end{aligned}$$

X size
(n, m)

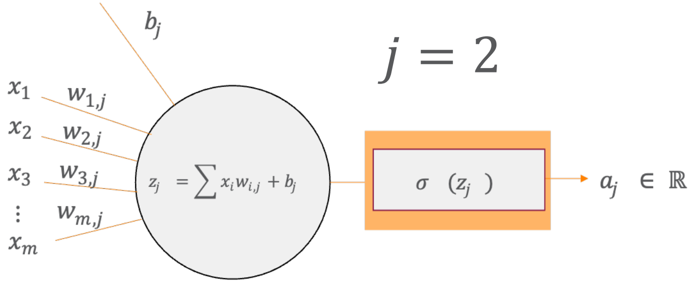
$[W_1, W_2, W_3]$
Size $(m, 3)$

Dot product $\boldsymbol{X[W_1, W_2, W_3]}$
size
$(n, 3)$

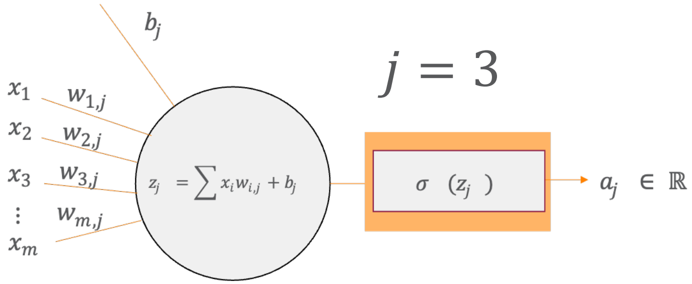
Multi-Layer Perceptron



$$a_1 = \sigma(XW_1 + b_1)$$



$$a_2 = \sigma(XW_2 + b_2)$$



$$a_3 = \sigma(XW_3 + b_3)$$

$$\begin{aligned}[a_1 \quad a_2 \quad a_3] &= \\ &= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)] \\ &= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3]) \\ &= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])\end{aligned}$$

$$a^{[1]} = \left[a_1^{[1]}, a_2^{[1]}, a_3^{[1]}\right] = \sigma\left(XW^{[1]} + b^{[1]}\right)$$

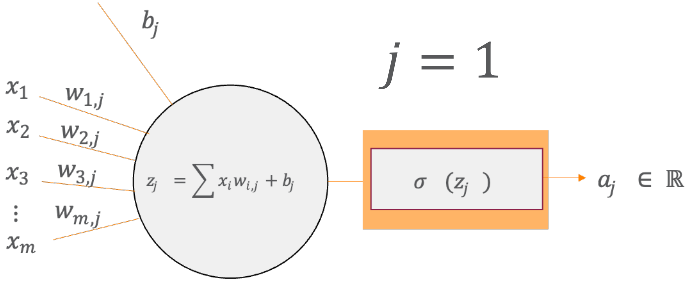
$$\left[W_1^{[1]}, W_2^{[1]}, W_3^{[1]}\right] \quad (n, 3)$$

$$\left[b_1^{[1]}, b_2^{[1]}, b_3^{[1]}\right] \quad (1,3)$$

Numpy adds these with broadcasting (i.e., $ones(n, 1) \cdot b^{[l]}$)

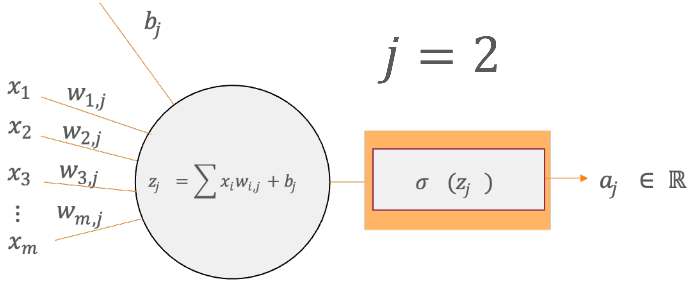
Multi-Layer Perceptron

$j = 1$



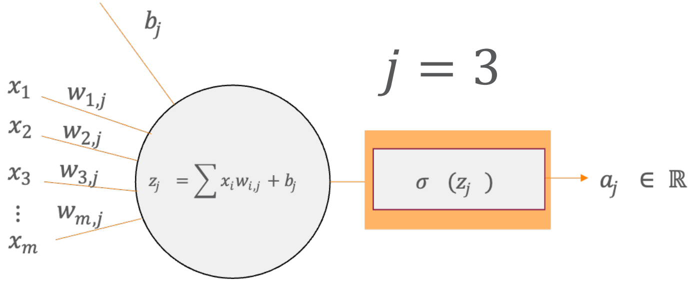
$$a_1 = \sigma(XW_1 + b_1)$$

$j = 2$



$$a_2 = \sigma(XW_2 + b_2)$$

$j = 3$



$$a_3 = \sigma(XW_3 + b_3)$$

$$[a_1 \quad a_2 \quad a_3] =$$

$$= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)]$$

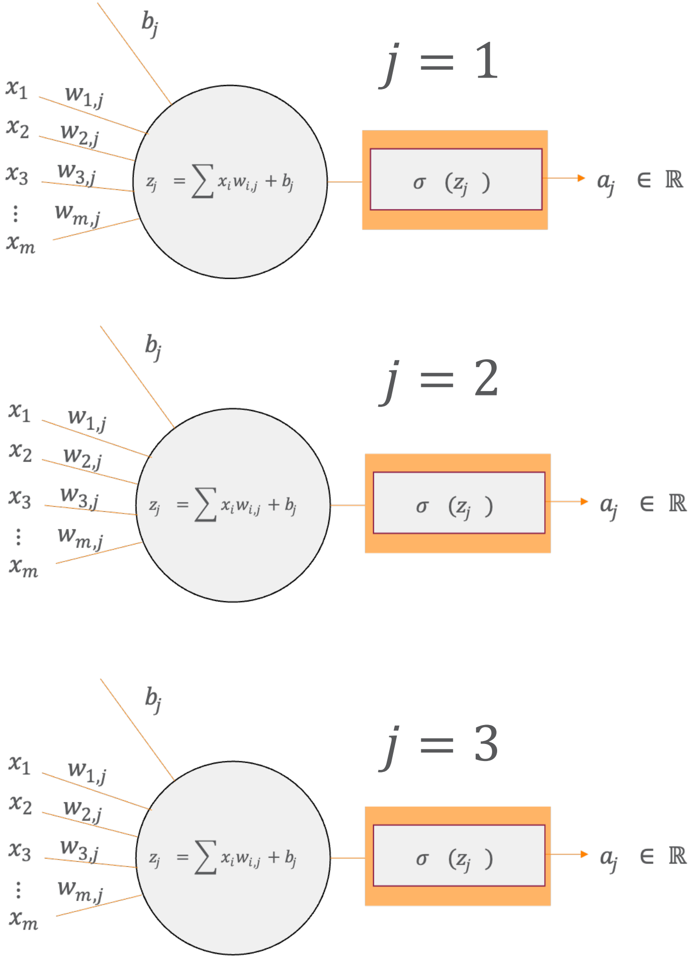
$$= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3])$$

$$= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])$$

$$a^{[1]} = \left[a_1^{[1]}, a_2^{[1]}, a_3^{[1]}\right] = \sigma\left(XW^{[1]} + b^{[1]}\right)$$

$$a^{[0]} = X \Rightarrow \; a^{[1]} = \sigma\left(a^{[0]}W^{[1]} + b^{[1]}\right)$$

Multi-Layer Perceptron



$$a_1 = \sigma(XW_1 + b_1)$$

$$a_2 = \sigma(XW_2 + b_2)$$

$$a_3 = \sigma(XW_3 + b_3)$$

$$[a_1 \quad a_2 \quad a_3] =$$
$$= [\sigma(XW_1 + b_1), \sigma(XW_2 + b_2), \sigma(XW_3 + b_3)]$$
$$= \sigma([XW_1 + b_1, XW_2 + b_2, XW_3 + b_3])$$
$$= \sigma(X[W_1, W_2, W_3] + [b_1, b_2, b_3])$$

$$a^{[1]} = \left[a_1^{[1]}, a_2^{[1]}, a_3^{[1]}\right] = \sigma\left(XW^{[1]} + b^{[1]}\right)$$

$$a^{[0]} = X \Rightarrow \quad a^{[1]} = \sigma\left(a^{[0]}W^{[1]} + b^{[1]}\right)$$

Output of layer l: $\boldsymbol{a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)}$

Example

$$a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$$

$$W^{[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

With this information, answer:

- How many layers are in this network?
- How many neurons are in layer 1?
- How many features are in the data?
- How many neurons are in the output layer?
- How many parameters are in the network?

Example

$$a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$$

$$W^{[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

With this information, answer:

- **How many layers are in this network?**
- How many neurons are in layer 1?
- How many features are in the data?
- How many neurons are in the output layer?
- How many parameters are in the network?

Recall input layer does not count as a layer.
We have two weight matrices for layers 1 and 2.
Therefore, we have a 2-Layer network.

Example

$$a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$$

$$W^{[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

With this information, answer:

- How many layers are in this network?
- **How many neurons are in layer 1?**
- **How many features are in the data?**
- How many neurons are in the output layer?
- How many parameters are in the network?

Layer weight matrix dimensions $\left(m^{[l-1]}, m^{[l]}\right)$, where $m^{[l]}$ is the number of neurons in layer l and $m^{[l-1]}$ is the number of neurons in the previous layer $l-1$.

- Layer 1 $W^{[1]} \rightarrow \left(m^{[0]}, m^{[1]}\right) = (3,2)$
- There are 2 neurons in Layer 1
- There are 3 neurons in input layer $\Rightarrow$ There are three features in our dataset.

Example

$$a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$$

$$W^{[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

With this information, answer:

- How many layers are in this network?
- How many neurons are in layer 1?
- How many features are in the data?
- **How many neurons are in the output layer?**
- How many parameters are in the network?

Layer weight matrix dimensions $\left(m^{[l-1]}, m^{[l]}\right)$, where $m^{[l]}$ is the number of neurons in layer l and $m^{[l-1]}$ is the number of neurons in the previous layer $l - 1$.

- Layer 2: $W^{[2]} \to \left(m^{[1]}, m^{[2]}\right) = (2,1)$
- There is 1 neuron in Layer 2 (Output Layer)
- We also confirm there are 2 neurons in Layer 1

Example

$$a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$$

$$W^{[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

With this information, answer:

- How many layers are in this network?
- How many neurons are in layer 1?
- How many features are in the data?
- How many neurons are in the output layer?
- **How many parameters are in the network?**

The number of parameters in a layer is the number of weights and biases or $m^{[l-1]} \times m^{[l]} + m^{[l]}$.

- Layer 1 $W^{[1]} \rightarrow (3,2) \Rightarrow params = 3\times 2 + 2 = 8$
- Layer 2 $W^{[2]} \rightarrow (2,1) \Rightarrow params = 2\times 1 + 1 = 3$
- Total number of parameters = 11

Pop Quiz

If the second and third layers of a neural network has 5 and 4 neurons, respectively. How many parameters are in the third layer?

Recall $a^{[l]} = \sigma\left(a^{[l-1]}W^{[l]} + b^{[l]}\right)$

A. 4

B. 20

C. 24

D. 5

Review

- ANN
  - Connectionism machines
    - Network of processing units
    - Memory is in the connections
  - Math
    - Matrix multiplication

Next Lecture

- Deep Neural Networks
- Convolutional Neural Networks
- Applications

Helper Slides