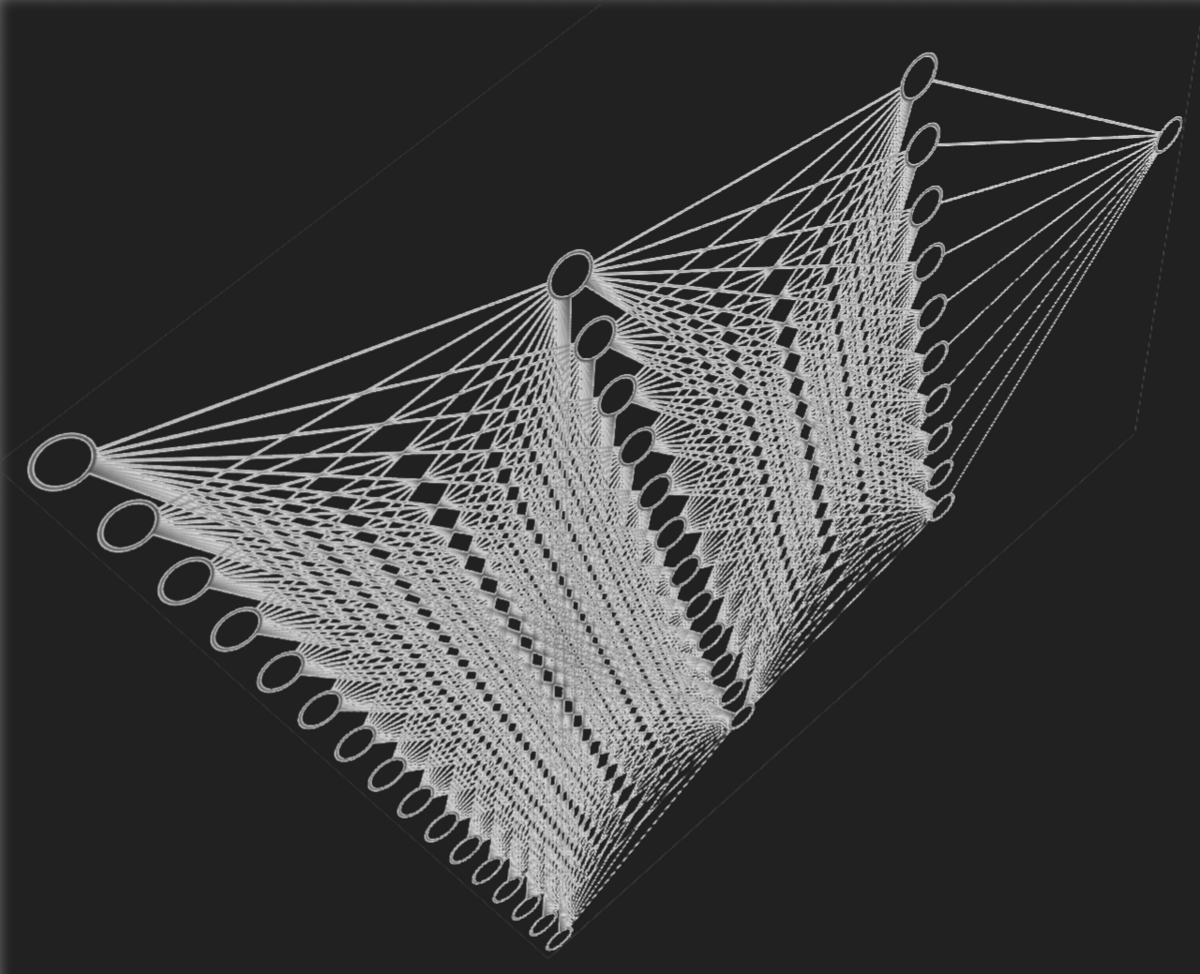
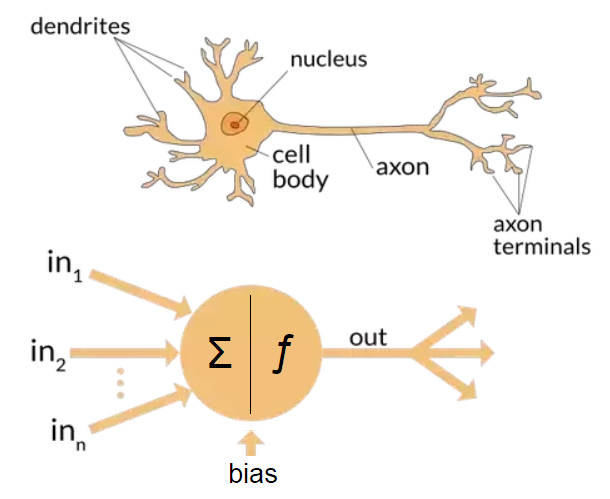
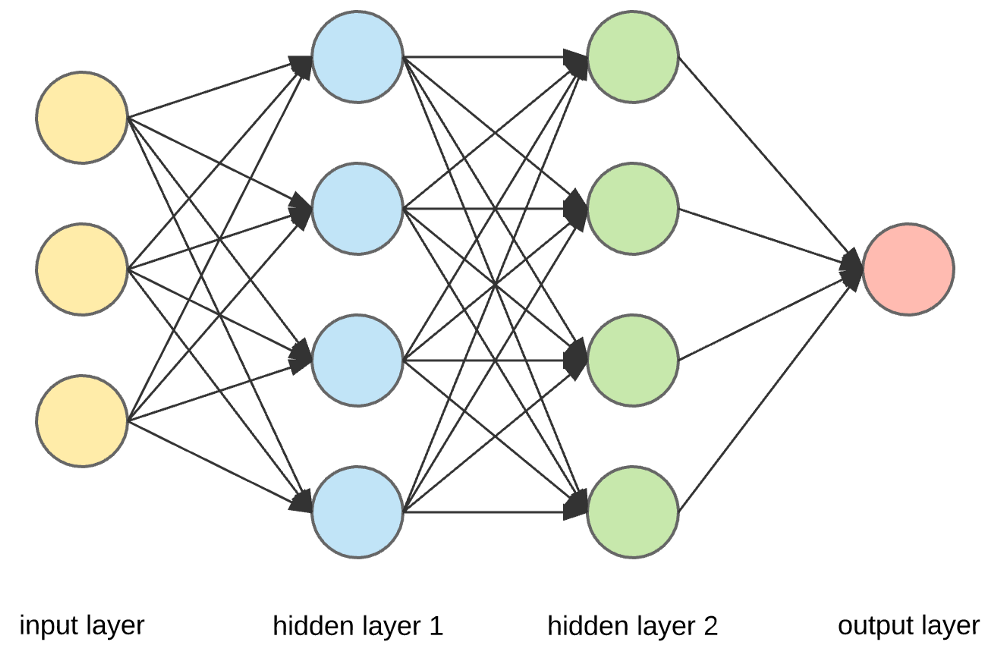
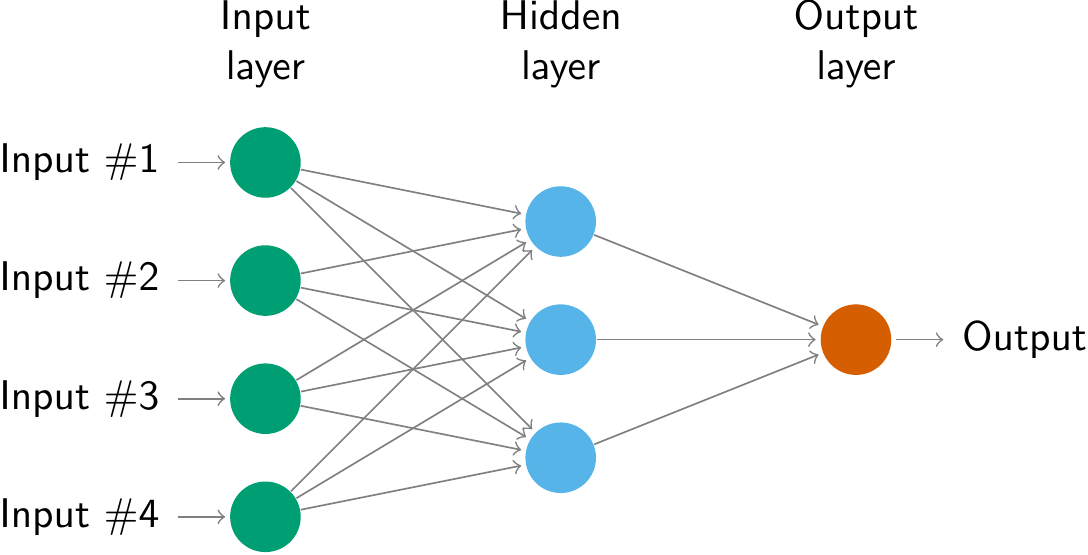
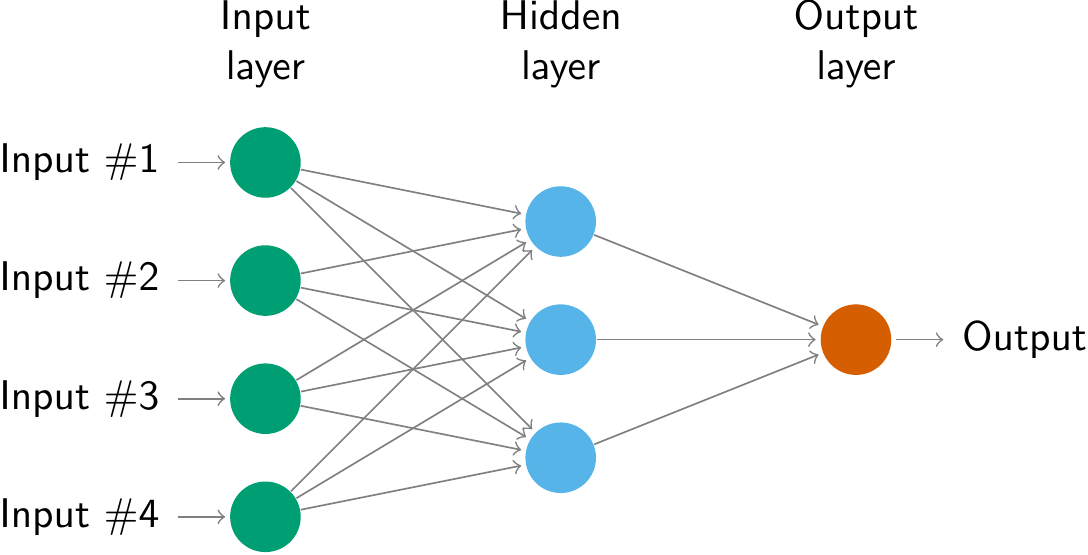
NEURAL NETWORK NOTES

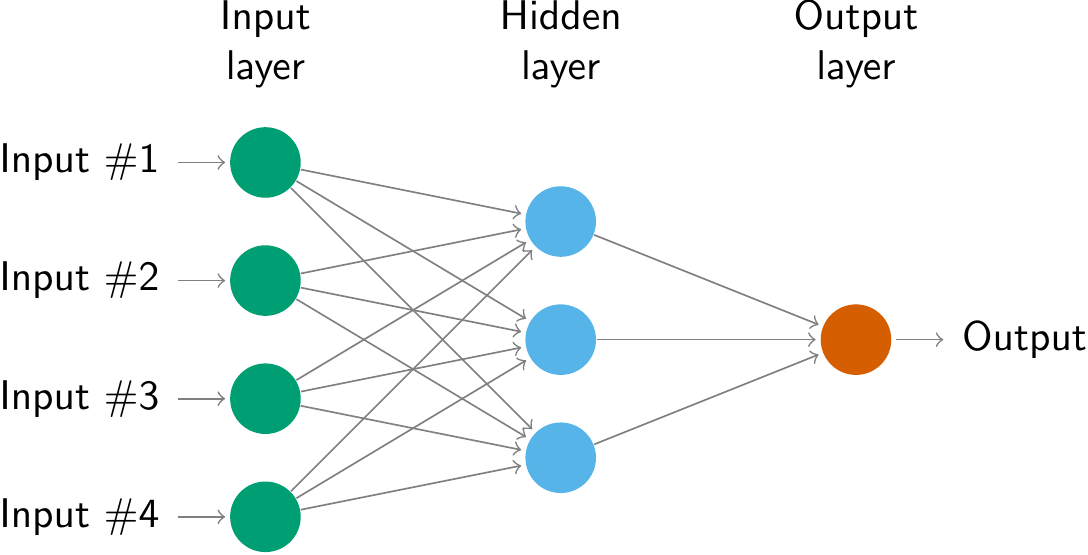


NEURAL NETWORKS DEFINITION

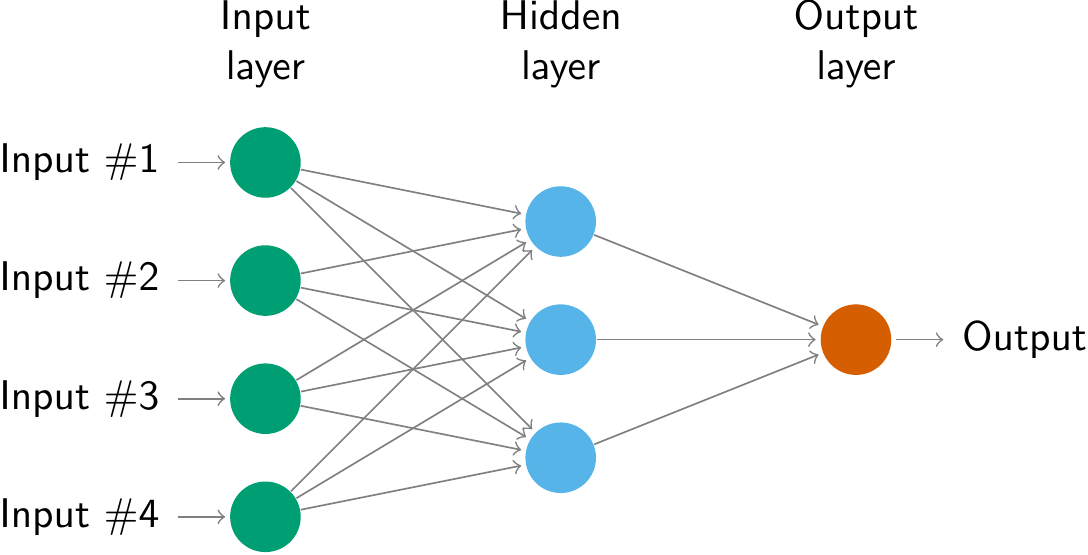
* + Definition of Neural Networks
    - A neural network is a computational model inspired by the way biological neural networks in the human brain process information. It consists of interconnected nodes, called neurons, organized into layers. These networks are designed to recognize patterns and make predictions based on input data.
    - Used Libraries for implementation
      * TensorFlow
      * PyTorch
    - Can Compute the NN Values with two types of Solutions
      * Forward Propagation
      * Backward Propagation
    - Intelligence Neural Network VS. biological neural networks
      * Process Steps
        + Receive Inputs
        + Process the data
        + Export the data ‘outputs’
      * 
  + Neural Networks Components
    - 
    - Inputs Layer
      * Inputs accessed for the model to predict the outputs
    - Output Layer
      * Output/s predicted from the model
      * Usually, it is giving the probability of prediction
      * Each node has its activation function
    - Hidden Layers
      * Collection of Nodes
      * Have m layers, each layer has n nodes
    - Nodes ‘Neurons’
      * Basic Component of the neural network
      * Make process at the accessed data to move the next layer till be output
      * Each node has its own activation function
      * Each Node Connect with all next layer’s nodes

NEURAL NETWORKS STRUCTURE

* + Let’s Go deep in neural networks structure
    - Using Activation Function Sigmoid is
    - Have 3 Layers
      * Layer 0 ‘first layer’, input layer, 4 neurons
      * Layer 1, hidden layer, 3 neurons, uses sigmoid activation function
      * Layer 2 ‘last layer’, 1 neuron, uses sigmoid activation function
      * 
    - Input Layer Structure
    - Hidden Layer Structure



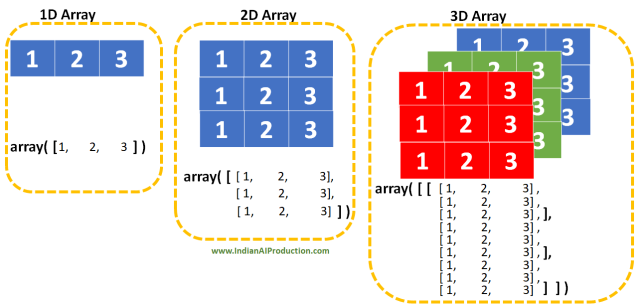
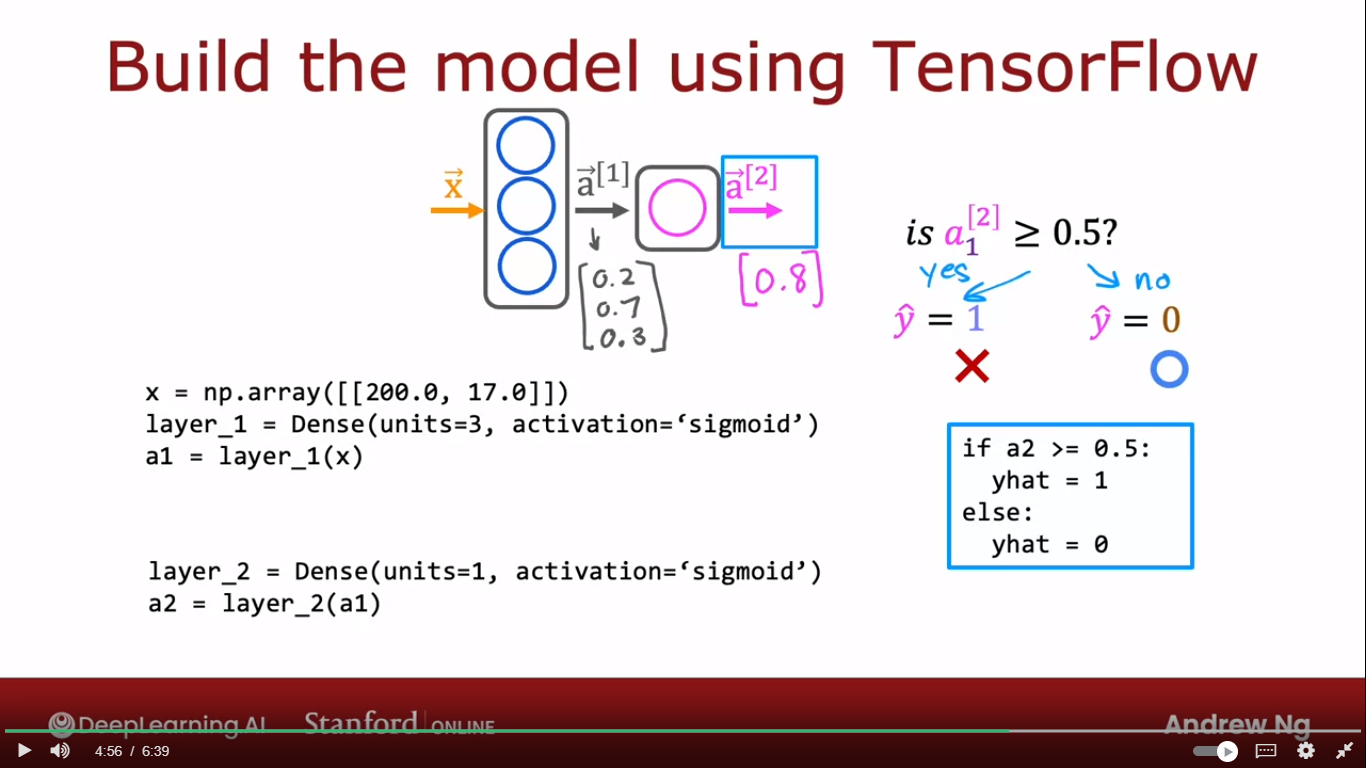
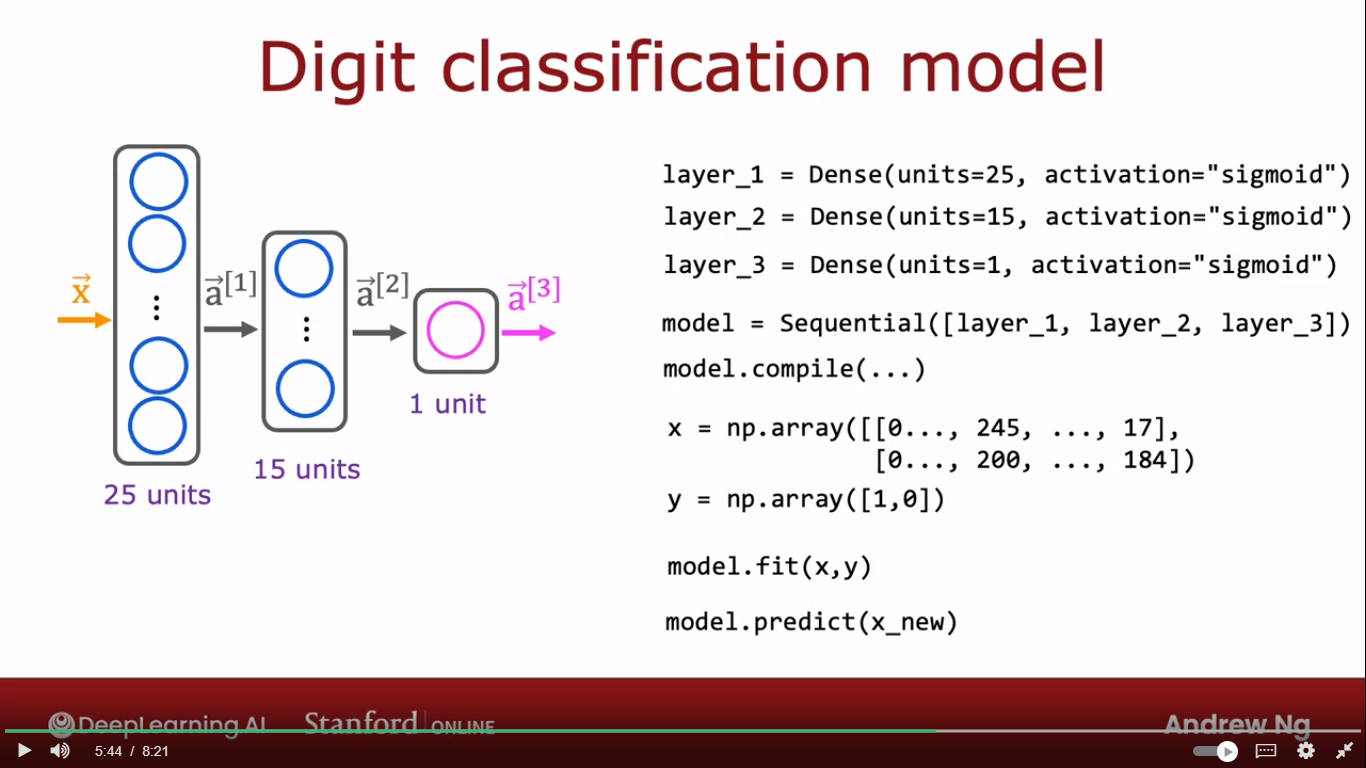
* + - Output Layer Structure



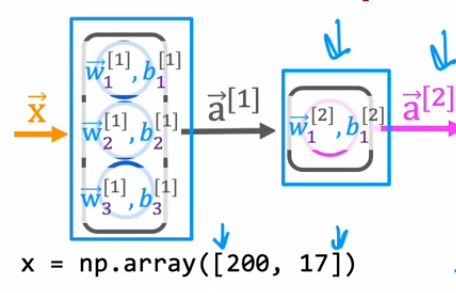
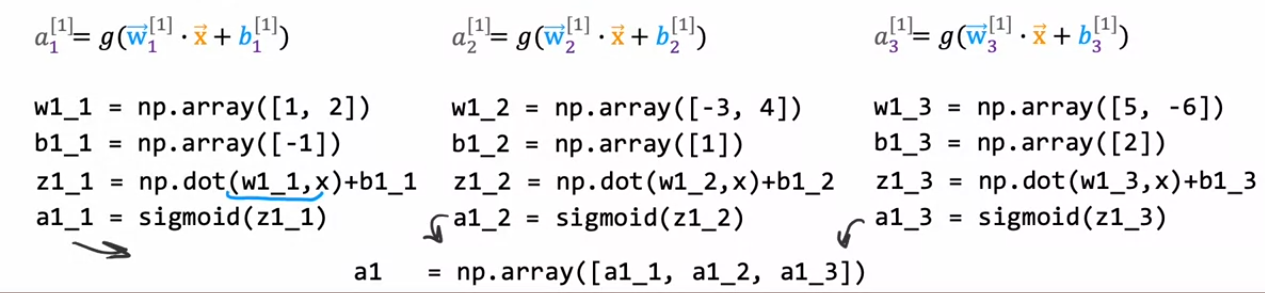
* + - The Public Formula of Activation function is

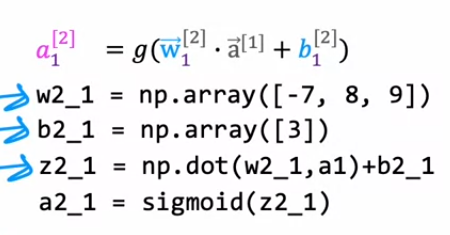
      * L, is the number of layers
      * J, is the Number of neurons
    - Finally,

TENSORFLOW IMPLEMENTAION

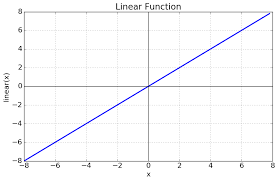
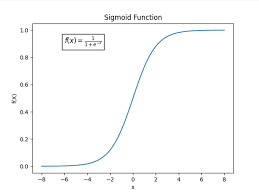
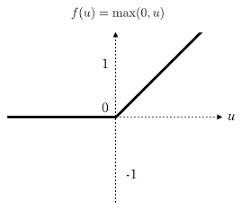
* + TensorFlow it a Python Library
  + Using for deep learning models building
  + Components Implantation
    - Create new array of features
      * Using NumPy library
      * 
    - Create new layer, Use it
      * Units, is number of neurons in this layer
      * Activation, is the activation functions used in this neuron
      * 
      * **a1 = layer\_1(x),** is using of the created layer
    - Using Sequential Function
      * Use for connect between layers
      * 

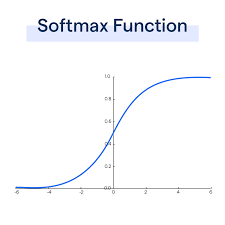
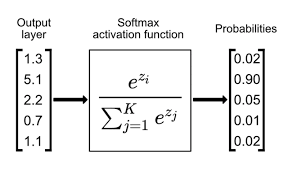
FORWARD PROPAGATION IMPLEMENTAION IN PY & NUMPY

* + Forward Propagation, it is shape of solutions using for solve the neural network model
  + Will Show steps to solve using forward prop using python and NumPy
    - This NN Which Showen in Image have 3 Layers
    - 
      * To Solve Using this NN
      * Will Divide to 3 layers
        + Input Layer ‘X’
        + Hidden Layer
  + 
    - * + Output Layer

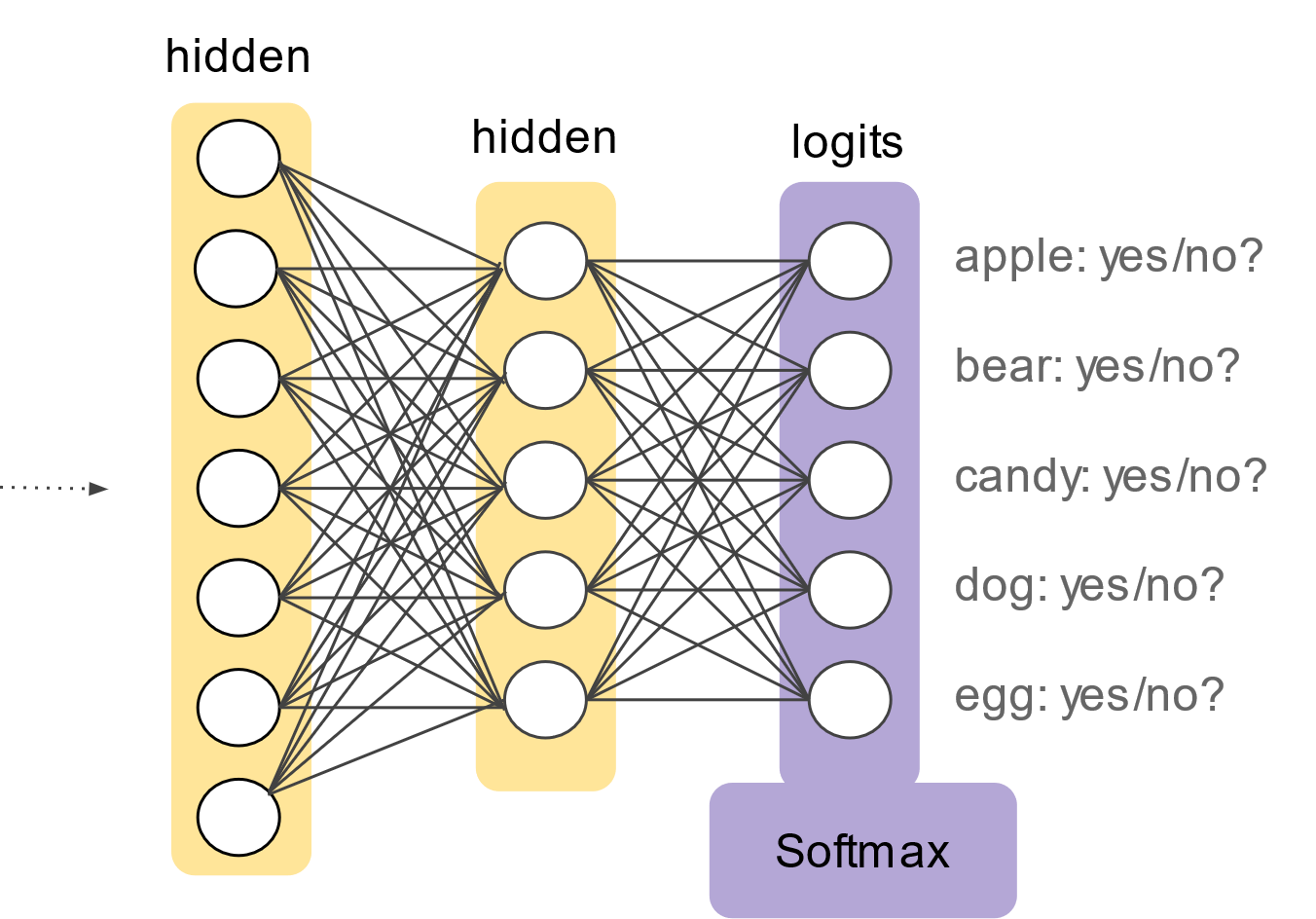
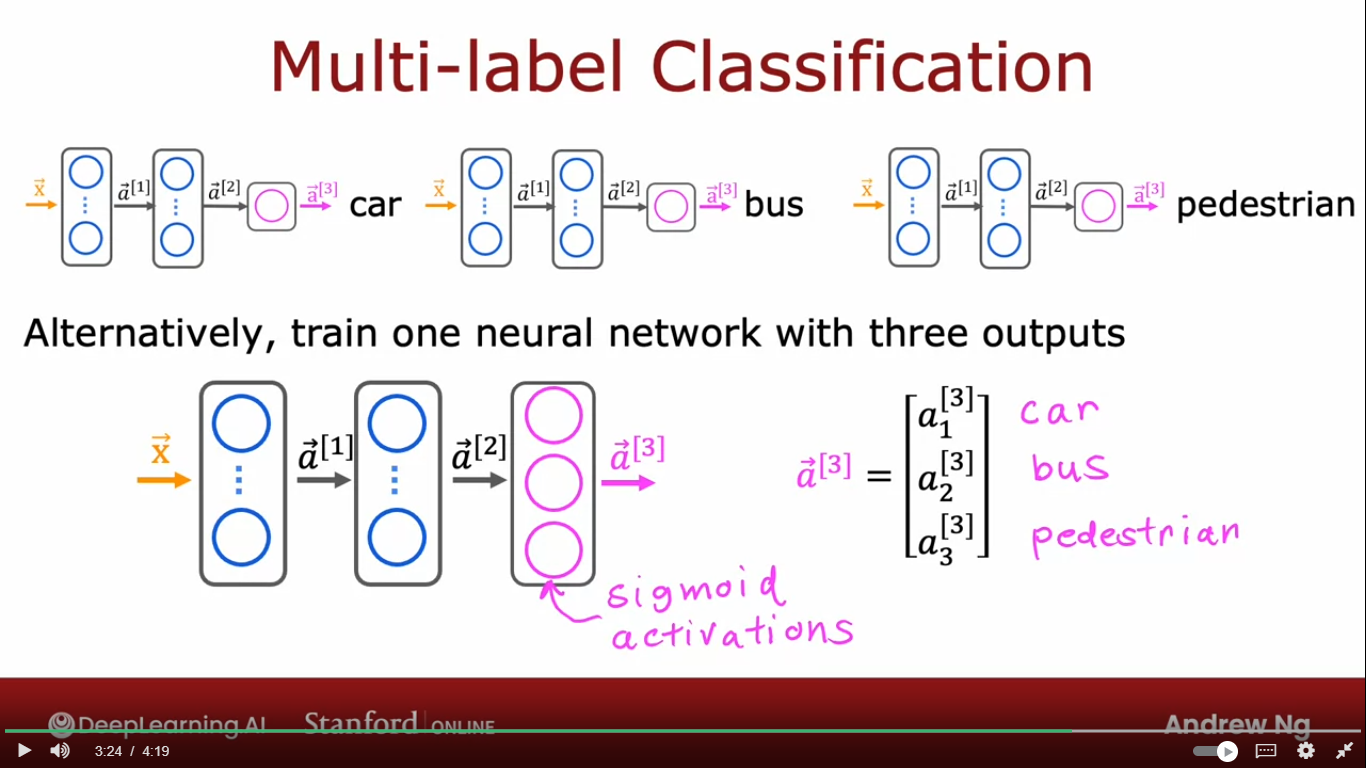


ACTIVETION FUNCTIONS

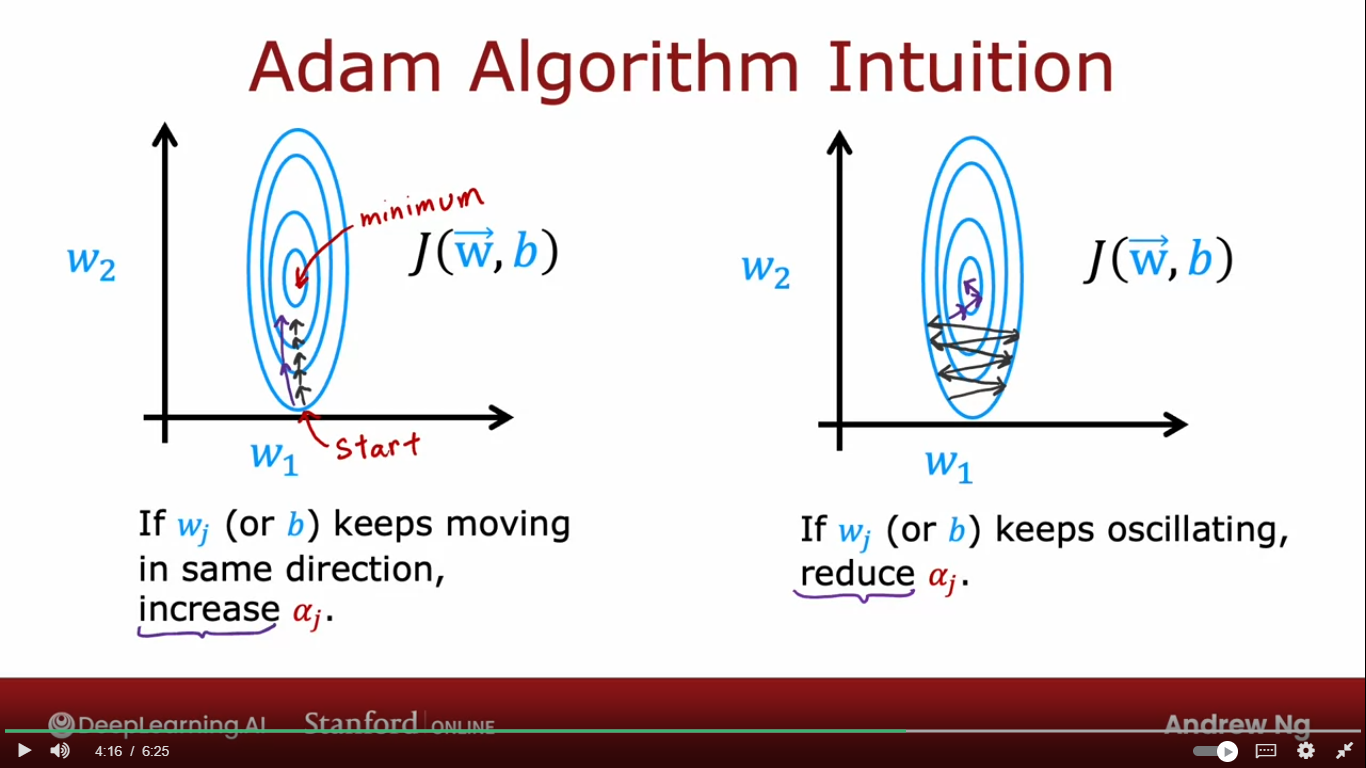
* + Activation Function, it is main Component of Neuron
  + Types of Activation Function
    - Linear Activation Function
      * It is Usually Uses for regression with negative and positive Values as
        + Predicting Stock market Changes
      * 
    - Sigmoid Activation Function
      * It is Usually Uses for Classify Binary Values 0/1 or T/F as
        + Using for Classify if have disease or not
      * 
    - ReLU Activation Function
      * It Is Usually uses for predict positive numbers as
        + Predicting House Price
      * 
    - SoftMax Activation Function

      * It is Usually Using for Predict Multiclass as
        + Predict Handwritten Numbers 0 – 9
      * 
      * Loss Function for SoftMax
      * Cost Function for SoftMax Activation Function
      * 

MULTICLASS VS MULTILABEL

* + Multiclass
    - Output Layer have N Neurons Should Choose One of Them
    - 
  + Multilabel
    - Output Layer have N Neurons Should Detect all of it Values
    - 

ADAM ALGORITHM

* + ADAM Algorithm, it is algorithm using for optimization using with gradient descent to achieve lowest cost fumction
  + 
  + Using for Minimize the Cost Function
  + ADAM Structure
    - If Gradient Decent moving in same direction
      * Will Increase learning rate ‘alpha’
    - If Gradient Decent moving in different, changed direction
      * Will reduce learning rate ‘alpha’

MARTIC MULTIPLICATION

BACK PROPAGATION