Intro to Machine Learning!

(Starting 5:10)

Agenda

- Speaker Intros
- How does a machine learn?
- Buzzword clarifications
- Supervised vs Unsupervised vs Reinforcement Learning
- ML Models
- Neural Networks (Structure, Activation and Cost Functions, Gradient Descent, Backpropagation)
- Code Exercise Setup
- Code Exercise Walkthrough
- Q&A

Speaker Intros

Let’s Start with a Question

What does it mean for a Machine to LEARN?

How does a Computer LEARN?

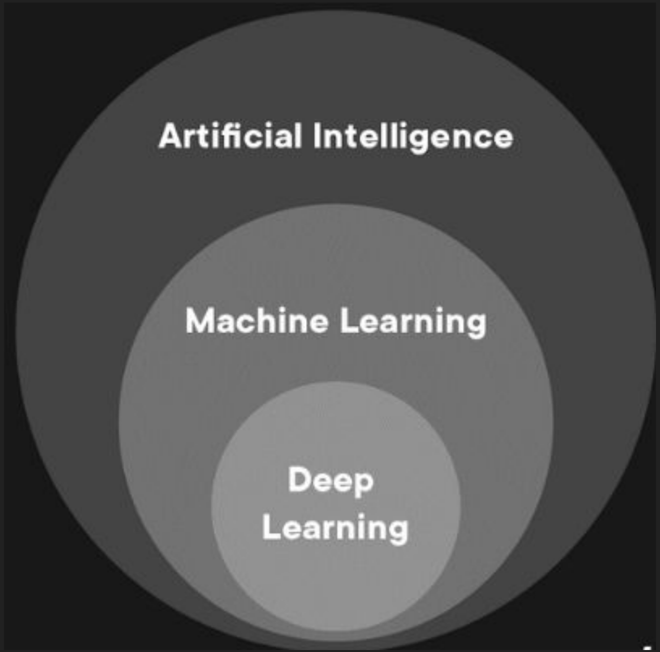
- First we have to gather relevant data
- Then feed the data into our model, sample by sample
- We then compare our models output to the output we want and we adjust the many parameters (weights, biases) within the model so that it can output a more accurate answer on future iterations
- After repeated iterations, the model’s answers with each inputted sample will get better and better
- This is called Training the model
- This is how a computer learns (with supervised learning)

Buzzwords

Can you think of any you hear often?

AI vs ML vs Deep Learning vs Data Science

- AI
  - Scientific field aimed at making machines act like humans
- ML
  - Subset of AI focused on using computational models to allow computers to perform intelligent tasks without being explicitly programed to do so
- Deep Learning
  - Subfield of ML dedicated to the engineering, research, and use of neural networks
- Data Science
  - Field of statistics aimed at trying to find useful patterns and structures within data

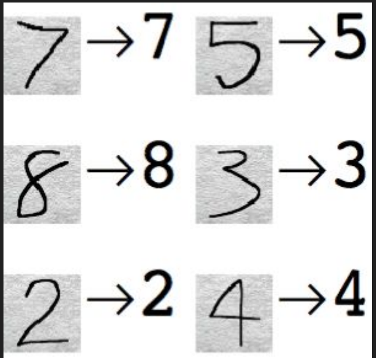


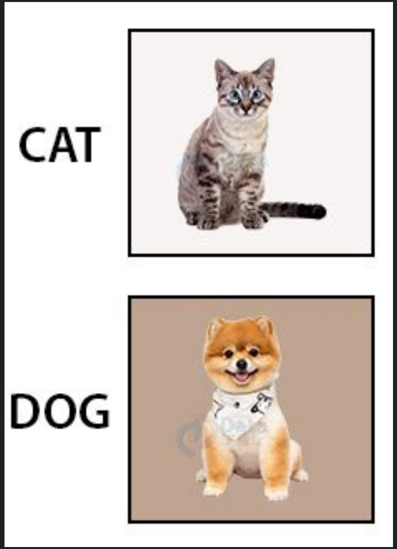
Supervised Learning

- Uses LABELED training data
- Each input in the data set has a corresponding output allowing us to measure how well our model is performing
- Usually used to solve two main types of problems:
    - Classification problems
    - Regression problems

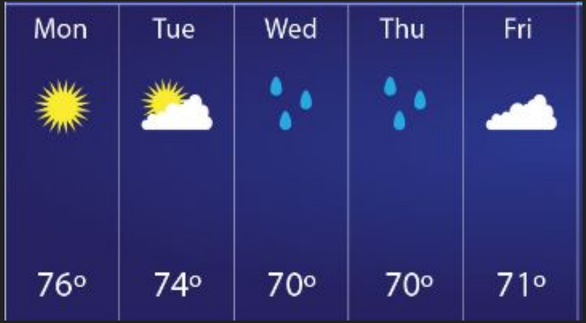
Supervised Learning Cont.

Classification:





Regression:



Unsupervised Learning

- Uses non-labeled data
- There are no output labels to allow us to check how our model is doing
- Usually used to find patterns, structures, relationships within your data
- Used to solve problems like:
  - Clustering
  - Anomaly Detection
  - Association Rule Mining
  - etc.

Reinforcement Learning

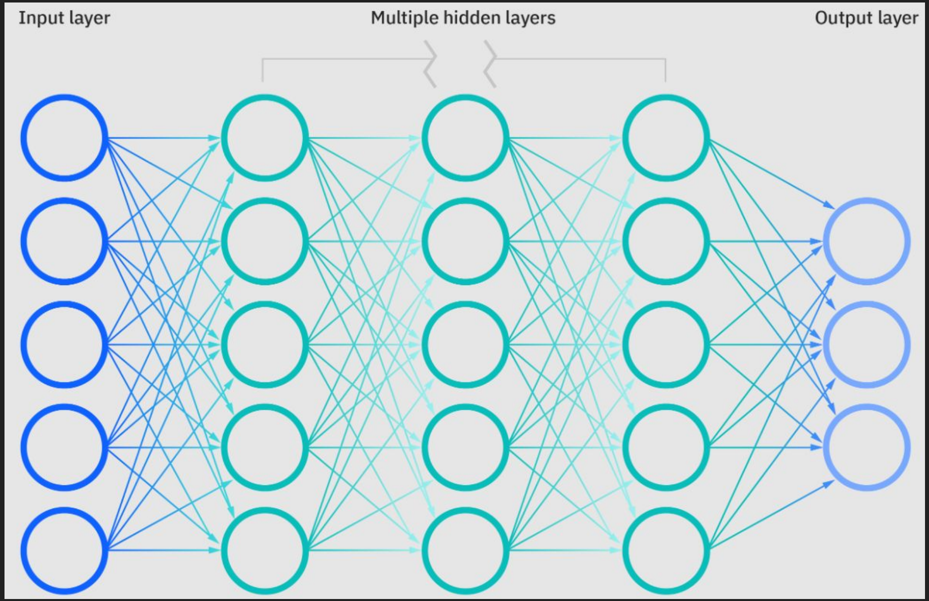
- Involves an agent that can interact with the world in ways that can give a reward
- The agent is tasked with exploring its environment to find ways to maximize the reward
- Does not require a data set
- Similar to how we would train a dog
- Can be used to for:
  - Robotics
  - Game playing (Chess, Go, Dota 2, etc)
  - Self driving

ML Models

- A machine learning model is an algorithm (or set of algorithms) and parameters that the computer uses to learn
- The model takes in input data, gives an output, and adjusts its parameters to make itself more accurate
- The models we use are problem specific i.e. some models are good at certain problems while others are not
- Model examples:
  - Linear Regression: Fitting a linear equation to data, used in predicting stock prices, home prices, etc
  - K-Nearest Neighbors (KNN): Clusters data points into groups based on similar features, used for classification and regression
  - Decision Trees: Tree-like structure, used for both classification and regression
  - **Neural Networks**: Used in deep learning models for image recognition, NLP, etc.

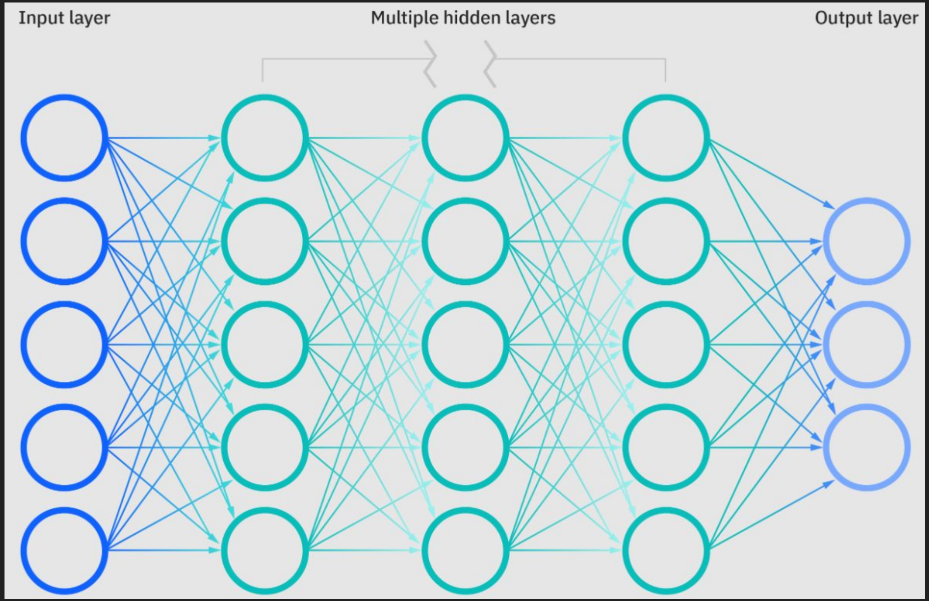
Neural Networks: Structure

- Structure inspired by the human brain hence the term ‘neural’
- Each circle is called a ‘neuron’
- Neurons are arranged in columns called layers
- Each neuron in one layer is connected to every neuron in the following layer
- Usually always 1 input and 1 output layer
- We can have many hidden layers (depth)
- There’s typically many more neurons in in the layers than shown in the diagram



Neural Networks: Structure Cont.

- Neurons can 'fire' with values from 0-1
- Specific groups of neurons firing at the same time cause neurons in the next layer to fire
- Each connection between neurons has a 'weight' describing how much a neuron activation affects the activation of the adjacent neuron
- Each neuron also has a value to act as a 'bias' against activation so a neuron only fires after a certain threshold



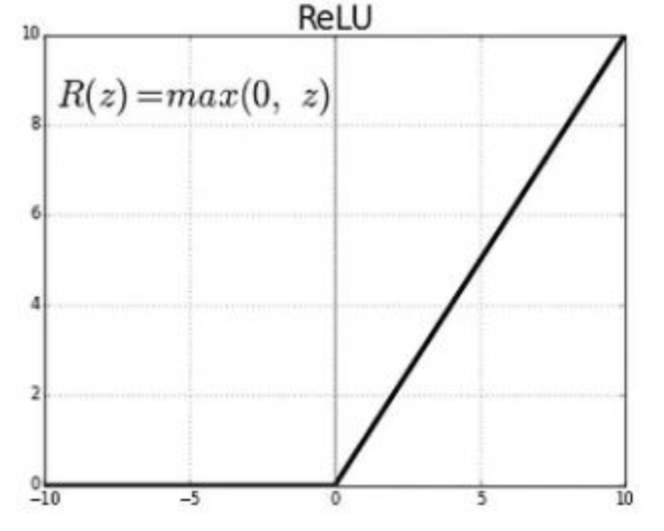
Neural Networks: Activation Functions

- The activation function is a function that gives the activation for a single neuron
- It takes in as input, the sum of the weighted activations of all the neurons in the previous layer + bias
- If Z is the input to the activation function, then we get:

  Z = w1x1 + w2x2 + w3x3 + ... + b

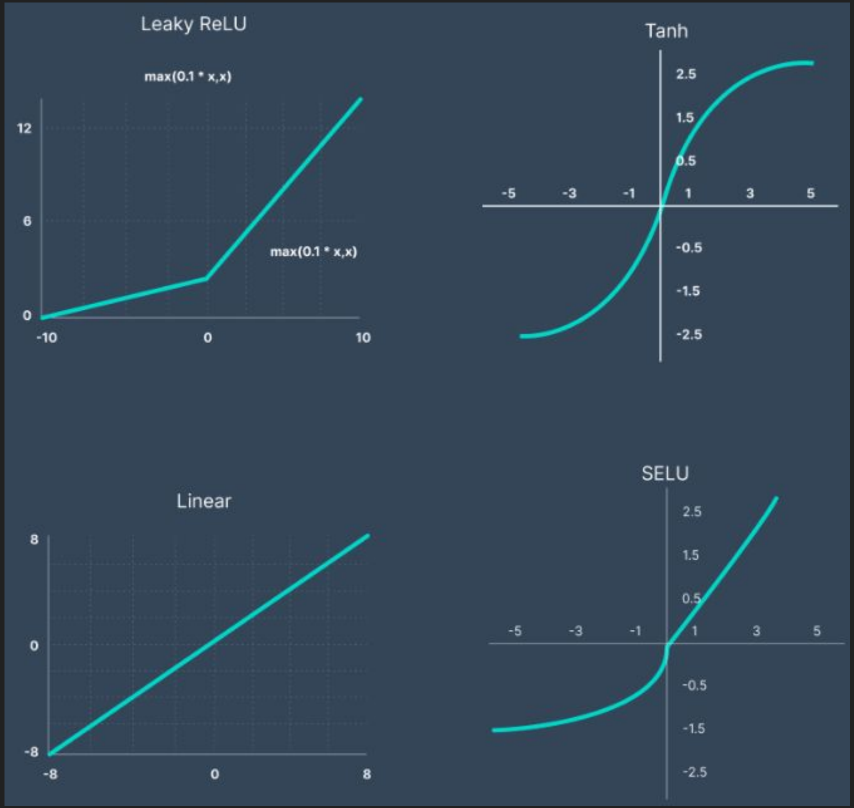
  Where wn and xn is the weight and activation for input neuron n respectively, and b is the bias for the current neuron

Example: Rectified Linear Unit activation function (ReLU)



Neural Networks: Activation Functions Cont.

- Each Neuron can be thought of as a function
- Each neuron takes in the weights and activations of the neurons in the previous layer then adds in its bias and outputs its own activation value using the activation function
- Multiple different activation functions can be used for different problems



Neural Networks: Cost Functions

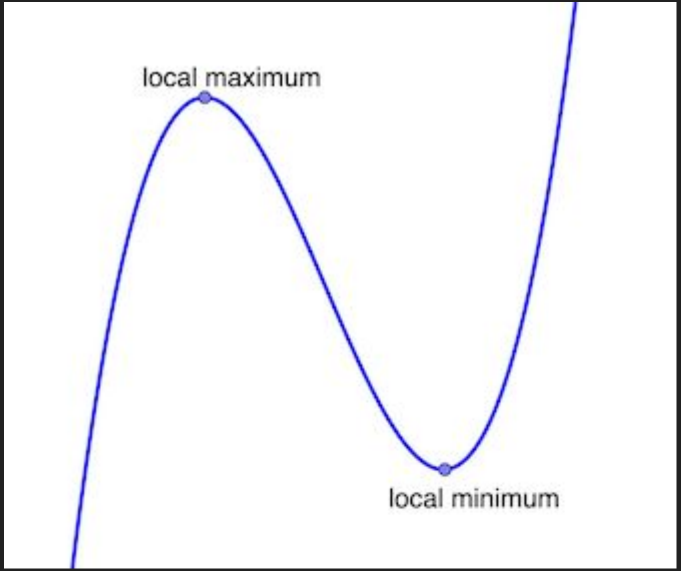
- The ‘Cost function’ is used to quantify how well our models outputs match the correct output values
- Can also be referred to as ‘Loss Functions’
- We use the Cost function to see how well our model is doing and use that metric to guide the optimization of the parameters (weights, biases) of our model
- Examples of Cost Functions:
    - Mean Squared Error (MSE)
    - Binary Cross-Entropy Loss (Log Loss)
    - Negative Log-Likelihood (NLL)

Neural Networks: Cost Functions Cont.

- Cost functions take as input ALL the weights and biases of the neural network
- Then run all the training data on the model using the given weights/biases
- Then output a single value called the ‘Cost’ which represents how good/bad our model is at outputting the correct answers
- A higher cost value implies a larger difference between the models outputs and the correct outputs
- A lower cost value implies a smaller difference between the models outputs and the correct outputs
- The goal is to minimize the Cost function by adjusting the weights and biases of our model
- Minimizing Cost function == LEARNING

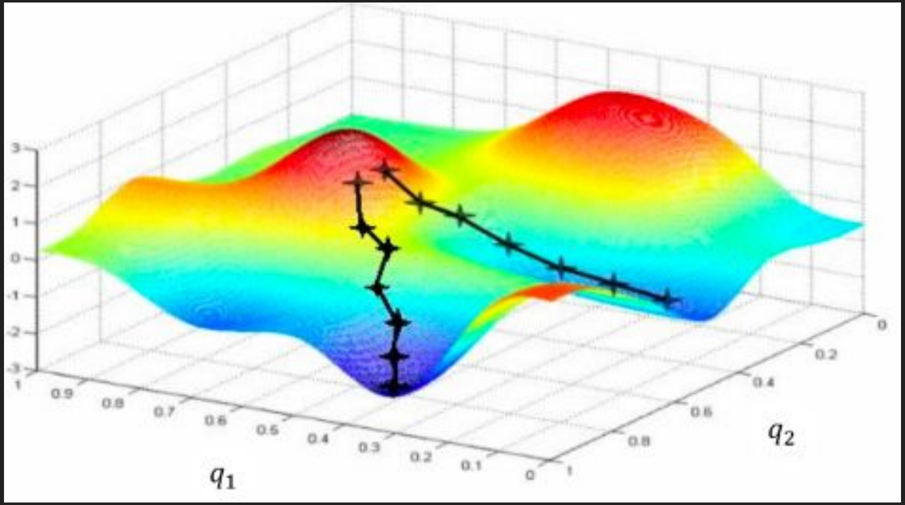
Neural Networks: Minimizing Cost

- If we want our model to perform better, we need to minimize our Cost function
- Similar to high school Calculus ‘finding minimum of the function’ type of questions
- Unfortunately when we’re dealing with a large number of inputs to our function, it’s much harder to find local minima
- We require another approach



Neural Networks: Minimizing Cost Cont.

- One common approach is called ‘Gradient Descent’
- Gradient Descent involves finding a direction that we can move to that lowers cost
- Once we find the Gradient, we use it to go backwards in our neural network to update the weights and biases
- Going backwards in our network to update the weights and biases is called ‘Backpropagation’
- We repeat this until we get to the local minimum



PyTorch

- What is PyTorch?
  - An open-source machine learning framework.
  - Developed by Facebook's AI Research lab (FAIR).
- Key Features:
  - Dynamic computational graph.
  - Pythonic and flexible.
  - Excellent support for deep learning.
- Popularity:
  - Widely adopted in research and industry.
  - Preferred by researchers and practitioners for its flexibility.
- Use Cases:
  - Deep learning research.
  - Computer vision, natural language processing, reinforcement learning, and more.

Handwritten Digit Recognition

- We will be using the MNIST data set for testing and training
  - The MNIST data set contains lots of handwriting samples
  - Each image is 28x28 in dimension *

- This is supervised learning for a classification task.

- There are six tasks for you to complete, 4 of them in main.py and 2 in extract.py

- Please work with others but do not give the answers to everyone!!
- Feel free to ask for help.

Code Exercise Setup

1. Open Powershell (Windows) or Terminal (Mac) AS ADMIN if possible
2. Enter ‘python --version’ and make sure version >=3.9
3. If you have an older version, download and install latest python version
4. In your Powershell/Terminal, Enter ‘pip install poetry’ and make sure it downloads successfully
5. Go to the link https://t.ly/PoBkE -> Click ‘Code’ -> Click ‘Download ZIP’ and unzip the file
6. Copy the file path that goes to files within the unzipped digit-recognition-main folder

   Windows example file path:
   C:\Users\Name\Downloads\digit-recognition-main\digit-recognition-main\

7. In your Powershell/Terminal, Enter ‘cd FILE_PATH’ where FILE_PATH is the path you copied earlier
8. Enter ‘poetry install’ and make sure it completes successfully
9. Enter ‘poetry shell’
10. Then enter ‘python main.py --help’ to get the full list of commands

Q&A

Thanks for Attending!