Full Name	Email	Assignment	Date of Submission
Deepthi Angara	deepthi.poduri@gmail.com	01	Aug 08th , 2020

Summary Report

Machine learning is diving deep into the world. Many organisations are trying to create systems that can learn, adopt, think and take actions on their own. In machine learning we come across the word "**predict**" more and we build models which learn from its experiences. For Ex:

1. Based on the customers buying options **predict** weekend offers.
2. Based on a scheduled event **predict** the traffic and give alternative paths.

The key components in building an ML machine are – Data, Models, Cost Function and Algorithms.

Data - For building a model we need lots and lots of training data. Having right data is important then having more data, which gives proper results. Data given to model during training phase is called training data and we come across training errors during execution similarly, data given to model during testing phase is called testing data and we come across testing errors during execution.

Model - Consists of set of protocols and transformation of data happens here to get our predicted output.

Cost Function/ Loss Function – Loss function is used to evaluating how well the algorithm models the given data. It depends upon parameters and dataset. Minimised loss function implies best values of models.

Algorithms – Once we have data and a model, we design an algorithm which searches for the best possible parameters for minimizing the loss function.

Few kinds of machine learnings -
Supervised Learning – This kind of learning can be understood as teacher supervising the learning process. Here a model is designed keeping targets/labels (y) in mind. Inputs are called features or covariates (x). Each (x,y) pair is called as an example or an instance. Regression, Classification, Tagging, Search and Ranking, Recommender System, Sequence Learning are types of supervised learning.

Unsupervised Learning – Unlike supervised learning there are no target values and there is no teacher. Generative Adversarial Network(GANs) is a recent machine learning framework developed in unsupervised learning which handles data such as images and audio files.

Reinforcement Learning – In this kind of learning the software interacts with the environment and takes actions accordingly. Neural Network problems comes into pictures here.

Exercise 1.8

1. Which parts of code that you are currently writing could be "learned", i.e., improved by learning and automatically determining design choices that are made in your code? Does your code include heuristic design choices?

    a. Though I am not writing any code, If I have to take a problem and discuss the learning strategy in it, maybe I shall take a web scraping problem which I'm currently working for a client and develop a model which recognizes the pattern in which the data is pulled out and learns the pattern. Model should be able to identify the addresses that are written in different ways and still be able to scrap them and fill the document in appropriate fields. A heuristic design choice can be applied for further scraping.

2. Which problems that you encounter have many examples for how to solve them, yet no specific way to automate them? These may be prime candidates for using deep learning.

    a. I feel health care industry can be considered as a best example here. A patient diagnosed with diabetes will eventually get into a mess of Diabetic Retinopathy, Cardiovascular disease, Neuropathy, Nephropathy, etc. Once the problem has occurred doctors have many ways for solving them based on existing examples and previous case studies as references. If we can automate this process of predicting which organ is going to be effected in near future based on vital readings or blood samples or some diagnosis then it would be helpful to many diabetic patients all over world or may be if model can predict well before a patient falls under a pre diabetic condition then the world will be on the other side.

3. Viewing the development of artificial intelligence as a new industrial revolution, what is the relationship between algorithms and data? Is it similar to steam engines and coal (what is the fundamental difference)?

    a. Undoubtedly, the relation between algorithm and data is similar to as steam engines and coal. So as to reach a destination we need correct amount of quality coal. Similarly, so as to get our predicted outputs we need correct amount of quality data as input to an model. The data can be a well collected inputs or dynamically generated values during execution of algorithm.

4. Where else can you apply the end-to-end training approach? Physics? Engineering? Econometrics?

    a. If a self-driving car can be taken as an example for end-to-end training approach then this process involves lot of Math, Physics and Engineering. May be end-to-end approach is applied here.