CSCI 5922 -Neural Networks and Deep Learning
Problem Set - 1 Solutions

Gowri Shankar Raju Kurapati
Student ID 110568555

August 29, 2022

1 Artifical Intelligence vs Machine Learning vs Deep Learning

- **Artificial Intelligence** - AI is the ability to incorporate human intelligence into machines through a set of rules (algorithm). AI is composed of two words: ”artificial” in the sense of something created by humans, and ”intelligence” in the sense of the ability to understand or think appropriately about a situation or problem and find a solution. AI can be thought of as the study of training computers to mimic human intelligence and thinking abilities.
- **Machine Learning** - ML is a subfield of AI that uses statistical learning algorithms to develop intelligent systems. ML systems can automatically learn and improve without being explicitly programmed.
- **Deep Learning** - DL is a subfield of machine learning that deals with algorithms inspired by the structure and function of the brain, called artificial neural networks.

In a nutshell, AI is seen as a broad spectrum of everything that we do to mimic intelligence in machines/computers. Out of those, Machine Learning is a subfield that helps us achieve that intelligence with data, algorithms, and statistical methods & inferences. DL is the subset of ML and a very specific genre of ML that uses artificial neural networks as of neurons in the human brain.

$$DL \subset ML \subset AI \tag{1}$$

2 Supervised Learning Generalization

a
- Since the goal of the supervised learning algorithm is to classify data/ predict outcomes by looking at past trends or patterns, we will need the models to learn from the existing data and predict unseen data.
- For the mathematical model to predict has to be constructed, the model needs existing data and we call it ”Training Dataset” since it is used to train the algorithms to construct the desired model for the task.

  But when the question of how well a constructed model is performing on a task is asked, we will need to use the model on the unseen data (we made the model for this very purpose), only then will we get the gist of how well it can predict. In other words, we want to know well in advance, how well the algorithm is trained to predict the future outcome. But since we want to answer this question even before we use it for actual purposes, what we do is split the data we have into training and test, so that when the model is constructed on the training data, we will have a beginner dataset to test on, which is the ”test data” so as to understand how well the model is performing on unseen data. So for the model, the test data is the unseen data or data for future prediction and model analysis.

b
- As stated above in (a), the very purpose to having test data is to understand how well the model is performing on UNSEEN data(for the model, it is the future data that it has to predict).
- So it makes NO logical sense to have an overlap for training and test data sets.
- Also, if that overlap exists, the model would have already seen the data that it has to predict, and there are very high chances for the model to perform so much better than it is inherently on the unseen data.
- Therefore, there MUST NOT be any overlap between training and test data.

3 Artifical Neurons

a **Model Training**

Given $\eta = 0.1$, $w_0 = 0, w_1 = 0, w_2 = 0, w_3 = 0$ and training data points as

$$X_1 = [1\ 0\ 1]$$
$$X_2 = [1\ 1\ 0]$$
$$X_3 = [1\ 0\ 0]$$

- First Epoch
  - at X_1
    - Calculating Output

$$W_T[1\, X_1] = (0)(1) + (0)(1) + (0)(0) + (0)(1) = 0$$
$$\implies Output^{(1)} = +1$$

    - $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_1 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_2 = (0.1) * ((1) - (1)) * (0) = 0$$
$$\Delta w_3 = (0.1) * ((1) - (1)) * (1) = 0$$

    - $W = W + \Delta w$ = [0 0 0 0] + [0 0 0 0]= [0 0 0 0]
  - at X_2
    - Calculating Output

$$W_T[1\, X_2] = (0)(1) + (0)(1) + (0)(1) + (0)(0) = 0$$
$$\implies Output^{(2)} = +1$$

    - $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_1 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_2 = (0.1) * ((1) - (1)) * (1) = 0$$

$$\Delta w_3 = (0.1) * ((1) - (1)) * (0) = 0$$

    * $W = W + \Delta w$ = [0 0 0 0] + [0 0 0 0]= [0 0 0 0]

  - at X_3
    * Calculating Output

$$W_T[1\,X_3] = (0)(1) + (0)(1) + (0)(0) + (0)(0) = 0$$
$$\implies Output^{(3)} = +1$$

    * $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((-1) - (1)) * (1) = -0.2$$
$$\Delta w_1 = (0.1) * ((-1) - (1)) * (1) = -0.2$$
$$\Delta w_2 = (0.1) * ((-1) - (1)) * (0) = 0$$
$$\Delta w_3 = (0.1) * ((-1) - (1)) * (0) = 0$$

    * $W = W + \Delta w$ = [0 0 0 0] + [-0.2 -0.2 0 0]= [-0.2 -0.2 0 0]

- Second Epoch
  - at X_1
    * Calculating Output

$$W_T[1\,X_1] = (-0.2)(1) + (-0.2)(1) + (0)(0) + (0)(1) = -0.4$$
$$\implies Output^{(1)} = \text{-1}$$

    * $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((1) - (-1)) * (1) = 0.2$$
$$\Delta w_1 = (0.1) * ((1) - (-1)) * (1) = 0.2$$
$$\Delta w_2 = (0.1) * ((1) - (-1)) * (0) = 0$$
$$\Delta w_3 = (0.1) * ((1) - (-1)) * (1) = 0.2$$

    * $W = W + \Delta w$ = [-0.2 -0.2 0 0] + [0.2 0.2 0 0.2]= [0 0 0 0.2]

  - at X_2
    * Calculating Output

$$W_T[1\,X_2] = (0)(1) + (0)(1) + (0)(1) + (0.2)(0) = 0$$
$$\implies Output^{(2)} = +1$$

    * $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_1 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_2 = (0.1) * ((1) - (1)) * (1) = 0$$
$$\Delta w_3 = (0.1) * ((1) - (1)) * (0) = 0$$

    * $W = W + \Delta w$ = [0 0 0 0.2] + [0 0 0 0]= [0 0 0 0.2]

  - at X_3
    * Calculating Output

$$W_T[1\,X_3] = (0)(1) + (0)(1) + (0)(0) + (0.2)(0) = 0$$
$$\implies Output^{(3)} = +1$$

    * $\Delta w_j = \eta(target^i - output^i)x_j$

$$\Delta w_0 = (0.1) * ((-1) - (1)) * (1) = -0.2$$
$$\Delta w_1 = (0.1) * ((-1) - (1)) * (1) = -0.2$$
$$\Delta w_2 = (0.1) * ((-1) - (1)) * (0) = 0$$
$$\Delta w_3 = (0.1) * ((-1) - (1)) * (0) = 0$$

* $W = W + \Delta w =$ [0 0 0 0.2] + [-0.2 -0.2 0 0]= [-0.2 -0.2 0 0.2]

The final weight vector after 2 epochs is **[-0.2 -0.2 0 0.2]**

b **Model Testing**

x_0	x_1	x_2	x_3	$W^T[1X_i]$	$Output_i$	$Target_i$ (Y_i)
1	1	1	0	(-0.2)(1) + (-0.2)(1) + (0)(1) + (0.2)(0) = -0.4	-1	-1
1	1	0	1	(-0.2)(1) + (-0.2)(1) + (0)(0) + (0.2)(1) = -0.2	-1	-1
1	1	1	1	(-0.2)(1) + (-0.2)(1) + (0)(1) + (0.2)(1) = -0.2	-1	1
1	0	0	0	(-0.2)(1) + (-0.2)(0) + (0)(0) + (0.2)(0) = -0.2	-1	1

c **Model Evaluation** Confusion Matrix

n = 4	Actual +1	Actual -1
Predicted +1	0	0
Predicted -1	2	2

d **Model Evaluation**

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{2}{4} = 0.5 = 50\%$
- Precision = $\frac{TP}{TP+FP}$
  Since TP + FP = 0 ,there are no positive cases in the test data, so any analysis of this case has no information, and so no conclusion about how positive cases are handled. The precision is undefined.
- Recall = $\frac{TP}{TP+FN} = \frac{0}{2} = 0$