Mussa Ibn Ashraf Al Arafi

Student Id: 209139176

Course: Machine Learning

York St. John University(London)

Q1 - Machine Learning and In-Depth Learning

a.

Definition. In machine-readable learning, the algorithm is trained under the supervision of input data. Supervised learning algorithms are learned from training data and find relationships between input and output features.

Unattended reading is a symbol of a hidden structure or distribution within data. It has no difference of output against training data

Goal. The purpose of supervised learning is to determine the function of the relationship between input elements versus output components. This function is not a direct relationship between input and output factors but a limited function of estimating the target value.

The goal of the unregulated learning model is to detect hidden patterns within the input data.

Output. The supervised learning model always yields results.

Unattended learning model detects hidden distribution of data.

Supervised learning phases.

- Separation
- Descending
- Unsupervised learning stages.
- Consolidation
- Organizations
- Accuracy.

The results of the unregulated learning model may not be as accurate as the moderated study models.

Monitored ML

- Separation
- Decrease
- Linear decline
- Support vector machine

Unattended ML

- Cluster algorithms
- K-methods
- Category collection

b.

- KNN is a moderated machine learning model and K-mean is an unregulated machine learning algorithm
- KNN is a learning algorithm for regression and separation machine while K-means is an algorithm for learning machine integration.
- KNN is a slow (lazy) student while K-Means is a fast learner.
- KNN performance is best if all data is the same size but this is not true for K-means.
- In KNN the number ‘K’ represents the number of nearby neighbors, while K-mean ‘K’ indicates the number of classes.
- At KNN we can find predictive errors, but there is no concept of predictive error in K-mean

c.

K-Nearest Neighbor or (KNN) is a comprehensive system for identifying patterns between data values and their distribution. The KNN cable can be used in voting systems to reach conclusions about basic data patterns. In the voting system KNN detects patterns from unknown patterns in a given data and divides them into similar elements. If most of the data belongs to one category it tries to paste anonymous data into more data categories. We can define a value in k to specify neighbors to be divided as the same category.

d.

Loss of Machine Learning:

During the preparation of our machine learning model, we split the data into two parts for train and test data. Training is used to train the machine learning model. After training our machine learning model, we need to test it to see if it works properly or not.

To validate our model, we feed test data into our model. Our model produces results called speculation against experimental data. The difference between the predicted value and the actual value is called **LOSS**.

In the reverse type of machine learning model, MSE (Mean Squared Error).

Loss = Y - Y '

Where “Y” is the real value and “Y’ ”is the approximate value.

MSE Census:

MSE = 1 / n * Σ (actual value - predicted value) ^ 2

Here:

n = number of items,

Σ = abridged comment,

Actual value = recorded value of the outgoing element (Y)

Prediction Price = Estimated or predicted value by regressor (Y ').

The process

- First, find the regression line.
- Enter your X values in the lineback count to get the new Y’s (Y’s) values.
- Subtracting the predicted value Y’s and the actual Y value will give us the MSE
- Add all the errors you have calculated against the specific values checked.
- Get an explanation for the errors.

Example problem:

Let's find the MSE for the following values: (44,42), (45,46), (46,48),(47,44).

Step 1: Find the regression line. The bottom line is rated when y = 9.2 + 0.8x.

Step 2: Find new Y values:

9.2 + 0.8 (44) = 44.4

9.2 + 0.8(45) = 45.2

9.2 + 0.8 (46) = 46

9.2 + 0.8(47) = 46.8

Step 3: Y-Y ':

42 – 44.4 = -2.4

46 – 45.2 = 0.8

48 - 46 = 2

44 – 46.8 = -2.8

Step 4: Square value:

-2.4 = 5.76

0.8 = 0.64

2 = 4

-2.8 = 7.84

Step 5: Add all square error values: 5.76 + 0.64 + 4 + 7.84 = 18.24.

Step 6: Find the error that means square:

18.24 / 4 = 4.56.

This line is the worst rate for these pairs.

e.

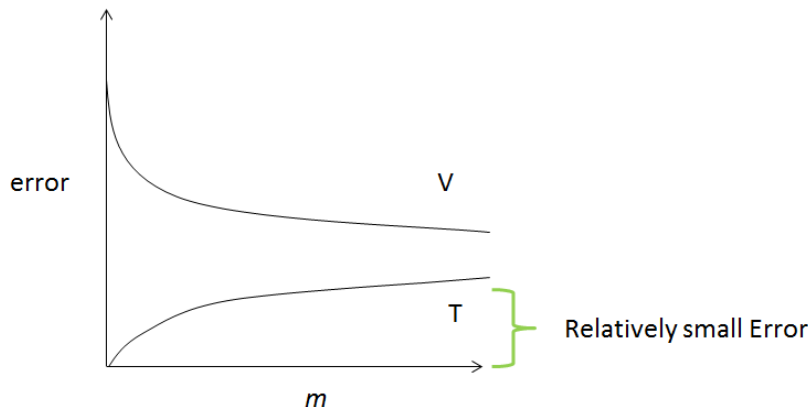
Excessive input into machine learning

Over-entry in Machine Learning An over-filling event occurs in machine learning when the machine learning model is highly aligned with the data trained in it, so it does very poorly in data testing. We may find that our ML model is overloaded or by checking it out. If our ML model performs very well or the training data and does not provide significant values without training data, it means that your model is overcrowded and will predict erroneous and problematic results soon. In Overfitting the model learns about instability within the training data until it acts as a malfunction in the new data. This means that random variables within the training data are taken and read as concepts by the model. The problem is that this concept does not apply to test data and reduces the accuracy of the ML model in new data. In particular we see Overfitting problems with non-parametric and indirect models with more flexibility when studying targeted work.

Many nonparametric machine learning algorithms also include parameters or techniques that we can use to limit the ML model in terms of what concepts to learn from training data. For example, decision trees are a non-metallic machine learning algorithm which is the most prominent opportunity to complete training data. This problem is usually solved by pruning the tree once it has been read to eliminate many details that have been picked up.

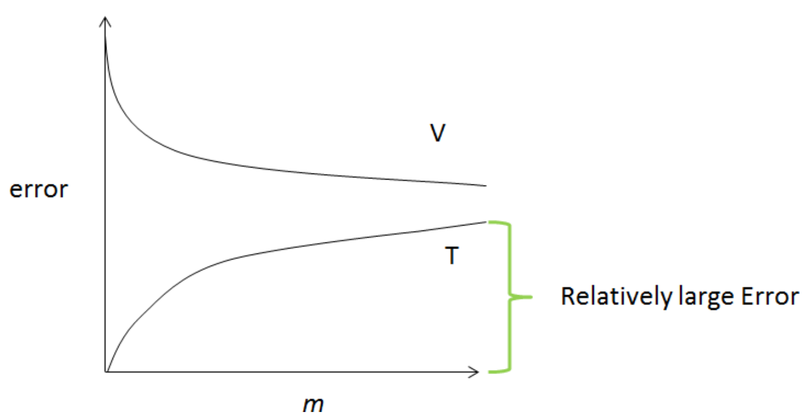
f.

1)



Here the we can see that the validation error is reducing by increasing the number of training pairs. So, by keeping the model to learn for more pairs of training we can decrease validation error and our model will perform when on the testing data and will also perform well in real world.

2)



Here by the passage of time and training pairs the its error is increasing and leading the model towards underfitting. By using early stopping we can save the modes for being underfit.

---

g

Early Stopping is a parameter regularizing technique for DL networks that stops training when parameter updates give no improvement. When the parameters are already regularized then training the model for more iterations is useless. It just increases the complexity and training time of model.

h.

1)

Number of Weights with one Hidden Layer

NW = Number_of_inputs * size_of_hidden_layer + size_of_hidden_layer * number_of_outputs

NW = 10 x 20 + 20 x 1

NW = 220

2)

Number of Weights with Two Hidden Layers

NW = Number_of_inputs * size_of_hidden_layer_1 + size_of_hidden_layer_1*size_of_hidden_layer_2 + size_of_hidden_layer_2 * number_of_outputs

NW = 256 x 512 + 512*512 + 512 x 10

NW = 131072 + 262144 + 5120

NW = 398336
