EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México
Instituto Tecnológico de Tijuana

ACADEMIC SUBDIRECTION
Systems and Computing Department

SEMESTER
August - December 2021

CAREER
Information and Communication Technologies Engineer

SUBJECT AND KEY:
Big Data BDD-1704TI9A

STUDENT'S NAME AND REGISTRATION:

Castillo Ramirez Guadalupe 17213043
Velázquez Farrera César Alejandro   17212937

NAME OF THE JOB:
Practice # 5 - Multilayer Perceptron classifier

UNIT TO BE EVALUATED
Unit II

TEACHER'S NAME:
M.C. Jose Christian Romero Hernandez

Introduction

Se desarrollará la práctica de acuerdo al tema Multilayer Perceptron classifier, la cual será desarrollado en scala, la cual fue tomado de ejemplo del documento de apache spark.

Multilayer Perceptron classifier

Multilayer perceptron classifier (MLPC) is a classifier based on the feedforward artificial neural network. MLPC consists of multiple layers of nodes. Each layer is fully connected to the next layer in the network. Nodes in the input layer represent the input data. All other nodes map inputs to outputs by a linear combination of the inputs with the node's weights w and bias b and applying an activation function. This can be written in matrix form for MLPC with K+1 layers as follows:

$$\mathrm{y}(\mathbf{x}) = \mathrm{f_K}(\ldots \mathrm{f_2}(\mathbf{w}_2^T \mathrm{f_1}(\mathbf{w}_1^T \mathbf{x} + b_1) + b_2) \ldots + b_K)$$

Nodes in intermediate layers use sigmoid (logistic) function:

$$\mathrm{f}(z_i) = \frac{1}{1 + e^{-z_i}}$$

Nodes in the output layer use softmax function:

$$\mathrm{f}(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}}$$

import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator

1.- Load the data stored in LIBSVM format as a DataFrame

```
val data = spark.read.format("libsvm")
  .load("data/mllib/sample_multiclass_classification_data.txt")
```

2.- Split the data into train and test

```
val splits = data.randomSplit(Array(0.6, 0.4), seed = 1234L)
val train = splits(0)
val test = splits(1)
```

3.- specify layers for the neural network:
-input layer of size 4 (features), two intermediate of size 5 and 4
-and output of size 3 (classes)

```
val layers = Array[Int](4, 5, 4, 3)
```

4.- create the trainer and set its parameters

```
val trainer = new MultilayerPerceptronClassifier()
  .setLayers(layers)
  .setBlockSize(128)
  .setSeed(1234L)
  .setMaxIter(100)
```

5.- train the model

```
val model = trainer.fit(train)
```

6.- compute accuracy on the test set

```
val result = model.transform(test)
val predictionAndLabels = result.select("prediction", "label")
val evaluator = new MulticlassClassificationEvaluator()
  .setMetricName("accuracy")

println(s"Test set accuracy = ${evaluator.evaluate(predictionAndLabels)}")
```