

Tural Bakhtiyarli

Prof. Lili Mou

Computer Science 466

Apr 25, 2023

## Course project Report

### **Introduction**

For this project, I decided to Iris flower classification. There are multiple classes of Iris flowers, and each have their own unique features. These features include: sepal length, sepal width, petal length, and petal width. I decide to train 3 different machine learning models for classifications to classify three different types of Iris flowers: virginica, versicolor, and setosa. These three machine learning algorithms are: softmax multiclass classification, Neural Network with Adam optimizer and Neural Network with RMSprop optimizer.

### **Problem formulation**

In my machine learning algorithms, the inputs were as following [sepal\_length, sepal width, petal\_length, petal\_width] for each row of input. The output was an integer from [0,1,2]: 0 for setosa, 1 for versicolor, and 2 for virginica. I got the data set from UCI machine learning repository and there were 150 samples: 50 for each flower type. Therefore my input data X had the dimensions of  $150 * 4$  and my output Y had the dimensions of  $150 * 1$ . Then I separated these input and output datasets to training and testing datasets: 120 for training and 30 for testing.

### **Algorithms**

#### **1. *Softmax multiclass classification:***

As this problem is a classification problem, I wanted to use either logistic regression or softmax classification. As logistic regression is used for binary classifications, I decided to use softmax. For tuning the stepsize, I ran the program using different stepsizes from the list [0.1,0.2,...1]. I found out that the

stepsize 0.8 was giving me the best result(best accuracy for test data). Therefore I started to use this value. For batch size as well, I implemented the same logic.

## **2. *Neural Network with Adam optimizer:***

As I have implemented this when doing digit classification for CMPUT 412 course (Experimental Mobile Robotics), I wanted to implement algorithm to this problem as well. Adam optimizer is a modified version of Stochastic Gradient Descent. However, its difference being worse in generalization than SGD. I used the tensorflow module and used keras to implement Adam optimizer. I tuned the number of epochs for this algorithm by testing for epoch values from 1 to 100. After the testing, the value 67 for epoch seemed the best(giving the best test accuracy). As adam is also an adaptive learning rate method, there was no need to tune the learning rate.

## **3. *Neural Network with RMSprop optimizer:***

The benefit of using RMSprop is that it automatically tunes the step size, therefore, there was no need to tune it manually, causing fast convergence. To tune the number of epochs, just like with Adam optimizer, I tested the algorithm for epoch values from 1 to 100. The algorithm was giving accuracy=1 for multiple epoch instances, however the values after 60 epochs were better than before 60 epochs. Therefore, I decide to tune the epoch value to 68, where my algorithm also got accuracy = 1.

## **Evaluation Metric**

In my algorithms, I used crossentropy loss and accuracy = (number of times prediction is correct for test)/number of test samples. As my algorithms deal with classification, it is easy to use the real goal as measure of success: weather the algorithm chose the correct flower or not.

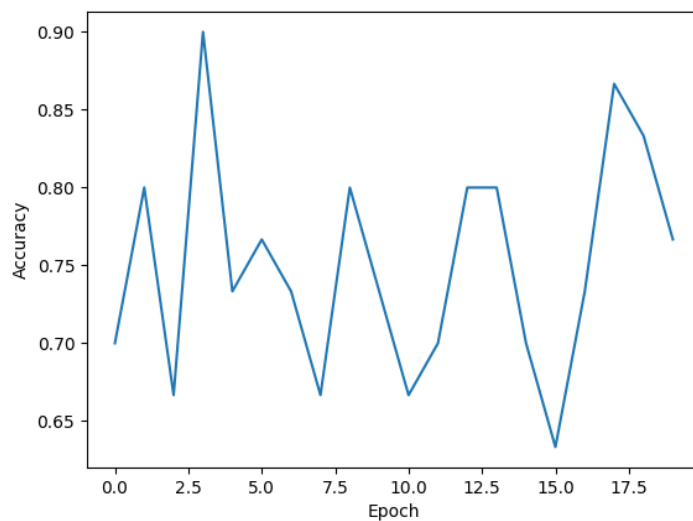
## **Results:**

After doing 20 test runs for each algorithm, and calculating the averages, here are the results:

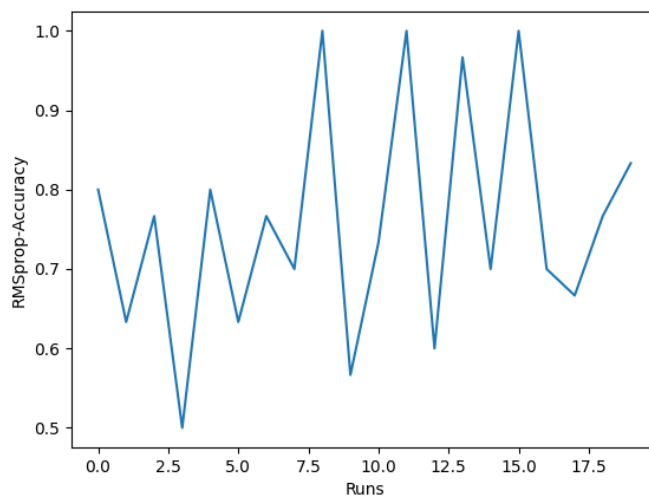
	Softmax	NN with Adam	NN with RMSprop
Test-data Accuracy	0.9617	0.75	0.7567

Here are the figures as well:

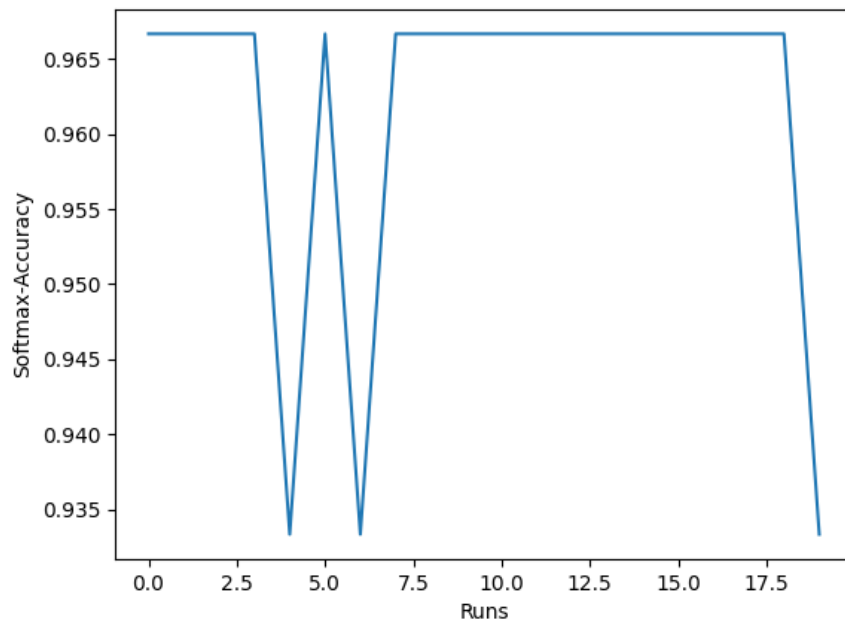
- NN with Adam:



- NN with RMSprop:



- Softmax Multiclass Classification:



It is a known fact that Adam optimizer suffers from generalization, therefore it was giving the highest accuracy when training, but it gave the lowest when testing. As the problem was very simple (only 3 classes and 4 features), Softmax multiclass classification did very well and even better than RMSprop and Adam. It also seems that RMSprop found models for weights where test data accuracy = 1.