Anmol More – 11915043 **Report**

1. *Create a simple softmax classifier with 3 neurons in output layer and no hidden layer using TensorFlow Core APIs.*
2. Read train and test set data and one hot encode the 3 classes of iris dataset
3. Declare weight matrix of size 4x3 and 3-dimensional bias of random normal shape with standard deviation of 0.5 and mean 0.
4. Next, we define the input x and y for our predictors
5. Softmax classifier is used - 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝑤𝑥+𝑏) along with simple gradient descent optimizer for cross entropy loss
6. Learn rate is chosen as 0.01 and each neural network is run over 1000 epochs
7. Since we have to report mean accuracy and standard deviation of 10 runs, step i) to v) and result for each run looks like –

A screenshot of a cell phone

Description automatically generated

1. is run 10 times and final mean accuracy and standard deviation is reported as –

**Mean Accuracy across all 10 runs : 0.923**

**Mean Standard Deviation across all 10 runs : 0.262**

1. *Create an MLP classifier with 3 hidden layers of sizes 5, 10, 5 using tensorflow Core APIs*
2. Read train and test set data and one hot encode the 3 classes of iris dataset
3. Next, we create a model with three hidden layers of size 5,10 and 5, each of them having relu as activation function and output layer with sigmoid function
4. In each of hidden layers, declare weight matrix of size 4x3 and 3 dimensional bias of random normal shape with standard deviation of 0.5 and mean 0. Here the value 0.5 for std is chosen based on 1/np.sqrt(4) since we have 4 features
5. Softmax cross entropy with logits is used as loss function, and adam optimizer is used for neural network learning
6. Learn rate is chosen as 0.01 and each neural network is run over 1000 iterations with above setting.
7. Since we have to report mean accuracy and standard deviation of 10 runs, step i) to v) is run 10 times

A screenshot of a cell phone

Description automatically generated

1. Final mean accuracy and standard deviation is reported as –
   1. **Mean Accuracy across all 10 runs : 0.72**
   2. **Mean Standard Deviation across all 10 runs : 0.399**

**Reason for reduced accuracy compared to simple neural network –**

1. We can see that just adding more and more hidden layers for simple dataset like iris doesn’t improves accuracy further. Reason for this is out of 10 runs, only 2 times we got an accuracy of .99 and other times accuracy reduced to 65-70% due to random initialization of weight and bias matrix

*5. Create an MLP classifier with 3 hidden layers of sizes 5, 10, 5 using Keras.*

1. Read train and test set data and one hot encode the 3 classes of iris dataset
2. Next, we create a model with three hidden layers and output layer with sigmoid function
3. Keras makes things very simple as each of out hidden layers of size 5, 10 and 5 is using relu for activation and output layer is of size 3 (we have 3 classes to predict) with sigmoid as activation function.
4. Since we have 3 classes to predict, we use categorical\_crossentropy for our loss function, but there are other possible loss functions which could have been used.
5. Keras model summary -

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

Hidden layer 1 (Dense) (None, 5) 25

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hidden layer 2 (Dense) (None, 10) 60

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hidden layer 3 (Dense) (None, 5) 55

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Output layer (Dense) (None, 3) 18

=================================================================

Total params: 158

1. Since we have to report mean accuracy and standard deviation of 10 runs, step i) to v) is run 10 times, and result for each run looks like –

A screenshot of a cell phone

Description automatically generated

1. final mean accuracy and standard deviation is reported as –
   1. **Mean Accuracy across all 10 runs : 0.956**
   2. **Mean Standard Deviation across all 10 runs : 0.016**