JAYASANKAR J N
S5 CSE
ROLL N0 39
TVE19CS039

HONOURS ASSIGNMENT 1

1. Design a simple Neural Network on MNIST Handwritten Digit Dataset. You can decide the number of layers and number of neurons in your network. Use cross entropy loss function with adam optimizer for your model. The performance measurement can be done in terms of accuracy. Compare the performances of at least two different models created by you by varying the number of layers and number of neurons in hidden layers. Also write an analysis of your two models in terms of number of parameters and epochs.

Github Link:

https://github.com/Jayasankar-JN/Honours-Assignment1

I have used 2 models. They are:

Model 1:

- Input layer having 784 units
- A hidden layer having 25 units and ReLU activation function
- A output layer having 10 units for 10 classes and softmax activation function
- Epoch = 5

Model 2:

- Input layer having 784 units
- A hidden layer having 64 units and ReLU activation function
- Another hidden layer having 32 units and ReLU activation function
- A output layer having 10 units and softmax activation function
- Epoch = 20

Here, model 2 is much more complex than model 1 as it has 2 hidden layers compared to 1 hidden layer of model 1. Also the number of neurons of hidden

layers for model 2 (96) is much higher than the number of neurons of hidden layers for model 1 (25)
Here I got a training accuracy of 95.75 % and test accuracy of 95.17 %for model 1. Since there isn't a realistic gap between training and test accuracy, the model is not an overfitting model. Since the model has high training accuracy, the model is also not an underfitting model. So we can say that model 1 is a good neural network model.

Whereas for Model 2, I got a training accuracy of 99.27 % and test accuracy of 96.07 %. Since there is a slight gap between training and test accuracy, we can say that the model is slightly overfitting. Since the model has high training accuracy, the model is not an underfitting model.

The reason for slight overfitting of model 2 may be due to the high complexity as there are a high number of parameters and more hidden layers. We can solve this by introducing a dropout layer or reducing the number of neurons in hidden layers.

Code for Model1:

```
model1 = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(25, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])

loss_fn = tf.keras.losses.SparseCategoricalCrossentropy()

model1.compile(optimizer='adam', loss=loss_fn, metrics=['accuracy'])
model1.summary()

print("Training...")
model1.fit(x_train, y_train, epochs=5)

print("Testing...")
model1.evaluate(x_test, y_test)
```

Code for Model2:

```
model2 = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(32, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])

loss_fn = tf.keras.losses.SparseCategoricalCrossentropy()

model2.compile(optimizer='adam', loss=loss_fn, metrics=['accuracy'])
model2.summary()

print("Training...")
model1.fit(x_train, y_train, epochs=20)

print("Testing...")
model1.evaluate(x_test, y_test)
```

Analysis:

For model 1, the number of parameters is 19885 which is much less thanthe training size of 60000. In fact the training size is more than 3 times the number of parameters. So we are reducing the chances of overfitting.Also the number of epochs is 5. Even with 5 epochs we are getting a training accuracy of 95.75% which is quite large.

For model 2, the number of parameters is 52650 which is really close to the training size of 60000. The training size is not 2 or 3 times the number of parameters. So there is a chance that the model can overfit. Also the number of epochs is 20 with which we get a training accuracy of 99.27 % !. But we can note that the training accuracy had already crossed 99 % in the 7td epoch. There isn't much change in accuracy in the later epochs. So 20 epochs is inefficient and not required and can be done in around 10 epochs.