CS60012 : Computing lab II
Spring 2021
Assignment - 10
Cloth Classification using RNN and CNN

Ravi pratap singh (20CS60R60)
ravips@kgpian.iitkgp.ac.in

{code for RNN and CNN also present in Google Colaboratory :
https://colab.research.google.com/drive/1LKQET7Ql_T0Pfukz8j1aVWFkOOCfodMA}

RNN

Model :

Hidden state size	: 100
Timesteps for unfolding the RNN	: 28
Non-linear activation	: tanh
Loss function	: categorical softmax cross entropy with logits
Optimizer	: Adam
Dropout with Retention probability	: 0.8

Parameters:

Number of BasicRNNCell layers	: 3
Loss scope	: 3 dense layers with 64, 32 and 10 outputs
Learning Rate	: 0.0005
Total No. of epochs	: 400
Batch size on training	: 200
Patience	: 30
Adam optimizer	: beta1=0.9, beta2=0.999, epsilon=1e-08

Results of training : (epochs on step of 5)

```
Epoch 0         Training accuracy: 77.500%      Validation accuracy: 75.467%    Loss: 0.645
Epoch 5         Training accuracy: 84.500%      Validation accuracy: 84.150%    Loss: 0.407
Epoch 10        Training accuracy: 86.000%      Validation accuracy: 84.383%    Loss: 0.420
Epoch 15        Training accuracy: 89.000%      Validation accuracy: 85.417%    Loss: 0.338
Epoch 20        Training accuracy: 87.500%      Validation accuracy: 87.150%    Loss: 0.316
Epoch 25        Training accuracy: 83.500%      Validation accuracy: 86.683%    Loss: 0.465
Epoch 30        Training accuracy: 89.500%      Validation accuracy: 88.217%    Loss: 0.341
Epoch 35        Training accuracy: 86.000%      Validation accuracy: 86.633%    Loss: 0.383
Epoch 40        Training accuracy: 91.500%      Validation accuracy: 88.483%    Loss: 0.271
Epoch 45        Training accuracy: 87.000%      Validation accuracy: 88.350%    Loss: 0.337
Epoch 50        Training accuracy: 90.500%      Validation accuracy: 88.300%    Loss: 0.246
Epoch 55        Training accuracy: 91.000%      Validation accuracy: 87.750%    Loss: 0.245
Epoch 60        Training accuracy: 87.000%      Validation accuracy: 88.300%    Loss: 0.321
Epoch 65        Training accuracy: 90.000%      Validation accuracy: 88.717%    Loss: 0.289
Epoch 70        Training accuracy: 89.000%      Validation accuracy: 88.733%    Loss: 0.291
Epoch 75        Training accuracy: 88.000%      Validation accuracy: 88.083%    Loss: 0.352
Epoch 80        Training accuracy: 85.500%      Validation accuracy: 89.100%    Loss: 0.303
Epoch 85        Training accuracy: 93.000%      Validation accuracy: 88.667%    Loss: 0.214
Epoch 90        Training accuracy: 91.500%      Validation accuracy: 88.767%    Loss: 0.284
Epoch 95        Training accuracy: 82.000%      Validation accuracy: 89.683%    Loss: 0.481
Epoch 100       Training accuracy: 88.000%      Validation accuracy: 89.100%    Loss: 0.327
Epoch 105       Training accuracy: 90.000%      Validation accuracy: 89.417%    Loss: 0.329
Epoch 110       Training accuracy: 88.000%      Validation accuracy: 88.800%    Loss: 0.310
Epoch 115       Training accuracy: 91.000%      Validation accuracy: 89.100%    Loss: 0.237
Early Stopping
```

Result on testing :

```
Test Loss: 0.306          Test Accuracy: 89.120
```

Inferences :

1. Out of learning rates [0.05,0.005,0.001,0.0005,0.0001] , choosed 0.0005 according to highest accuracy .
2. Out of batch sizes [100,150,200,250,300] , choosed 200 .
3. With 3 BasicRNNCell layers , the training time increased but test accuracy increased .That indicates that deeper the network, better learning is done. At deeper layers, more detailed features are learned.
4. Softmax prevents the overflow of the loss values .In the Loss scope used three dense layers with 64, 32 and 10 outputs, particularly, then the sparse softmax cross entropy with logits which is the default choice for multi-output classification problem.
5. **Dropout** of 0.8 had helped reduce overfitting and improve model performance.
6. Randomly Splitting training set into 90:10 for training and validation with seed of 101.
7. Early stopping is done when :$epochs\_without\_progress > patience$, where epochs without progress is incremented when $loss\_of\_current\_batch > best\_loss$. and $patience$ is set to 30 to avoid overfitting.

CNN

Model:

1st convolutional layer : 32 filters of dimension 5x5 followed by batch normalization, relu activation and max pooling (with 2x2 subsampling).
2nd convolutional layer : 64 filters of dimension 5x5 followed by batch normalization, relu activation and max pooling (with 2x2 subsampling).
Densely connected layer : 1024 hidden units
Output dense layer : 10 logits followed by batch normalization and softmax activation.
Loss function : Categorical softmax cross entropy with logits
Dropout with Retention probability : 0.8
Optimizer : Adam

Parameters :

Learning rate : 0.001
Number of epochs : 16
Batch size : 128
Patience : 5

Logits Batch normalization momentum : 0.9

Adam optimizer : beta1=0.9, beta2=0.999, epsilon=1e-08

Random seed : 101

Results of training :

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Epoch 0         Train accuracy: 92.188%         Validation accuracy: 87.158%    Loss: 0.223
Epoch 1         Train accuracy: 89.062%         Validation accuracy: 89.800%    Loss: 0.296
Epoch 2         Train accuracy: 92.969%         Validation accuracy: 90.825%    Loss: 0.196
Epoch 3         Train accuracy: 91.406%         Validation accuracy: 90.733%    Loss: 0.268
Epoch 4         Train accuracy: 96.094%         Validation accuracy: 91.700%    Loss: 0.155
Epoch 5         Train accuracy: 93.750%         Validation accuracy: 91.842%    Loss: 0.179
Epoch 6         Train accuracy: 93.750%         Validation accuracy: 92.125%    Loss: 0.160
Epoch 7         Train accuracy: 97.656%         Validation accuracy: 92.475%    Loss: 0.110
Epoch 8         Train accuracy: 100.000%        Validation accuracy: 92.142%    Loss: 0.027
Epoch 9         Train accuracy: 98.438%         Validation accuracy: 91.625%    Loss: 0.043
Epoch 10        Train accuracy: 99.219%         Validation accuracy: 92.600%    Loss: 0.032
Early Stopping
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Result on testing :

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO:tensorflow:Restoring parameters from ./log_cnn/log_cnn
Test Loss: 0.267         Test Accuracy: 91.740
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Inferences :

1. Out of learning rates[0.01,0.001,0.005,0.0001,0.0005] , choosed 0.0005 according to highest accuracy .
2. Out of batch sizes [32,64,128,256,512] , choosed 128 .
3. With 16 epochs the performance was almost the same when increasing the count .
4. Batch normalization method helped to regularize a convolutional network.It gave convolutional network a resistance to vanishing gradient during training and decreased training time and resulted in better performance.
5. **Dropout** of 0.6 has NOT helped to improve model performance and values close to 1.0, such as 0.8will be good for retaining inputs from the visible layer..
6. Randomly Splitting training set into 80:20 for training and validation with seed of 101. It's essential to have a separate dataset which doesn't take part in the training and is used to make sure that what we've learned can actually be generalised .
7. Early stopping is done when :$epochs\_without\_progress > patience$, where epochs without progress is incremented when $loss\_of\_current\_batch > best\_loss$. and $patience$ is set to 5.
8. Some initial epoch's iterations were overfitted(val acc>tran acc), but in consecutive iterations it was not seen.
9. Patience of 5 was good for the model as there was not much validation loss.