COL774 Assignment 4

Dipen Kumar (2018CS50098)
Goalla Varsha (2018CS10334)

January 17, 2021

1. (40 points) Non Competitive Part

1. (10 points) Vanilla Neural Network

Table 1: Learning Rate = 0.01 & Batch Size = 256
[{Accuracy on Train Data}, {Accuracy on Test Data}, {Training Time}]

Epochs	sigmoid	relu	tanh
10	40.04%, 37.37%, 49 sec	56.17%, 39.62%, 49 sec	51.03%, 38.52%, 44 sec
50	61.49%, 38.94%, 74 sec	90.50%, 39.56%, 71 sec	94.29%, 37.14%, 70 sec
100	87.71%, 37.90%, 165 sec	98.82%, 40.04%, 162 sec	99.70%, 37.51%, 159 sec
150	97.57%, 37.11%, 140 sec	99.05%, 39.80%, 166 sec	99.77%, 36.37%, 131 sec
200	99.16%, 37.17%, 210 sec	99.55%, 39.63%, 156 sec	99.85%, 36.03%, 169 sec

We can observe that "relu" best among all the given activation functions. This plain deep neural network having 1 hidden layer with 100 perceptrons clearly doesn't performs well for this problem since accuracy on test data saturates at 40%. This is clear from the above table that no matter how long we train it won't learn the general pattern rather we will eventually end up over-fitting.

Hence we choose activation = "relu" and $\max_e pochs$ = "100"

2. (10 points) Feature Engineering

Yes, some features other than the absolute pixel values be used. Some other high level feature extracted using "gabor" and "hog" filter on images.

"hog" filter takes 513 sec and "gabor" filter takes 687 sec

"hog" has accuracy = 25.11% and "gabor" filter has accuracy = 33.23%

3. (20 points) Convolutional Neural Network

As the model size increases the time to train and test increases. Introducing more layers and increasing the depth of the network increases the performance as seen from accuracy on test data and macro-F1 score. CNN show better accuracy and macro-F1 score with respect to neural network. It is obvious because in plain neural network absolute pixel values are used as features unless we have used any filter for feature extraction. But in convolution neural network we convolve with kernels to better extract high level features like human brain.

metric	plain neural network	convolution neural network
accuracy	98.82%, 40.04%	98.80%, 41.23%
macro-F1 score	0.34804	0.36776
training time	162 sec	1405 sec

Convolution neural network takes less epoch with respect to plain neural network. We kept batch size = "64" because convolution neural network does heavy computation w.r.t plain neural network, learning rate = "0.01", maxepoch = "75" and it takes maximum 25 minutes to run on "cpu" and can run faster on "gpu".

2. (60 points) Competitive Part

Model1:-
First we took the convolution neural network used in 3rd question of Non - Competitive part.
Obtained Train Accuracy = 99.16%
Obtained Test Accuracy = 42.41%
Optimizer used is Stochastic Gradient Descent with learning rate=0.01, momentum = 0.9.

Model2:-
Architecture:
Now 1 extra convolution and 1 extra fc layer are added with some changes in padding and stride.
1. Convolution Layer: Kernel size = (2,2), stride = 2, padding=0, depth = 64. ReLU Activation, Batch Normalisation are applied
2. Convolution Layer: Kernel size = (2,2), stride = 2, padding=0, depth = 128. ReLU Activation, Batch Normalisation are applied
3. Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
4. Convolution Layer: Kernel size = (2,2), stride = 2, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
5. Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
6. Fully Connected Layer with output size 256 followed by ReLU activation. Batch Normalisation
7. Fully Connected Layer with output size 64 followed by ReLU activation. Batch Normalisation
8. Fully Connected Layer with output size 7 followed by Softmax activation

Loss function used is Cross Entropy Loss. SGD with learning rate=0.01, momentum=0.9 is used as optimizer and weight decay = 1e-5 is applied which is equivalent to L2 regularisation,.L2 equivalent is used to avoid overfitting.
Obtained Train Accuracy = 99.8%
Obtained Test Accuracy = 44.77%

Model3:-
Now, a more complex structure is used by adding more convolution and fc layers.
Architecture:
1.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=0, depth = 64. ReLU Activation is applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 64. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
2.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
3.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
4.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch Normalisation are applied

Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
5.
Fully Connected Layer with output size 512 followed by ReLU, batch normalisation and dropout with probability = 0.5
6.
Fully Connected Layer with output size 256 followed by ReLU, batch normalisation and dropout with probability = 0.5
7.
Fully Connected Layer with output size 64 followed by ReLU, batch normalisation and dropout with probability = 0.5
8.
Fully Connected Layer with output size 7 followed by Softmax.
Loss function used is Cross Entropy Loss. SGD with learning rate=0.01, momentum=0.9 is used as optimizer and weight decay = 1e-5 is applied which is equivalent to L2 regularisation, dropout is also applied. L2 equivalent and dropout are used to avoid overfitting.
Epochs = 100
Obtained Train Accuracy = 71.16%
Obtained Test Accuracy = 55.91%

Model4:-
Now more layers are added to model 3.
‘ Architecture:
1.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=0, depth = 64. ReLU Activation is applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 64. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
2.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
3.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
4.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
5.

Fully Connected Layer with output size 512 followed by ReLU, batch normalisation and dropout with probability = 0.5
6.
Fully Connected Layer with output size 256 followed by ReLU, batch normalisation and dropout with probability = 0.5
7.
Fully Connected Layer with output size 64 followed by ReLU, batch normalisation and dropout with probability = 0.5
8.
Fully Connected Layer with output size 7 followed by Softmax.
Loss function used is Cross Entropy Loss. SGD with learning rate=0.01, momentum=0.9 is used as optimizer and weight decay = 1e-5 is applied which is equivalent to L2 regularisation, dropout is also applied. L2 equivalent and dropout are used to avoid overfitting.
Epochs = 100
Obtained Train Accuracy = 92.67 %
Obtained Test Accuracy = 58.28 %

Model5: Improved model4 by adding some layers, removing batch normalisation in fully connected layers and changing padding at some places.
Architecture:
1.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 64. ReLU Activation is applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 64. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 64. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
2.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 128. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
3.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 256. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
4.
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch

Normalisation are applied
Convolution Layer: Kernel size = (3,3), stride = 1, padding=1, depth = 512. ReLU Activation, Batch Normalisation are applied
Max Pooling: Kernel size = (2,2), stride = 2, padding=0.
Dropout with probability = 0.5.
5.
Fully Connected Layer with output size 512 followed by ReLU, dropout with probability = 0.5
6.
Fully Connected Layer with output size 256 followed by ReLU, dropout with probability = 0.5
7.
Fully Connected Layer with output size 128 followed by ReLU, dropout with probability = 0.5
8.
Fully Connected Layer with output size 64 followed by ReLU, dropout with probability = 0.5
9.
Fully Connected Layer with output size 7 followed by Softmax.
Loss function used is Cross Entropy Loss. SGD with learning rate=0.01, momentum=0.9 is used as optimizer and weight decay = 1e-5 is applied which is equivalent to L2 regularisation, dropout is also applied. L2 equivalent and dropout are used to avoid overfitting.
Epochs = 200
Obtained Train Accuracy = 98.26 %
Obtained Test Accuracy = 61.09 %
MacroF1 score = 0.5897
Time Taken = 2-3 hrs.