Web Information Retrieval and Data Mining - Assignment 3

Student: Yibin, Lei (1422685)

Answer 1.1 (4p)

Loading model from file

I first made a network with 3 hidden layers with 256,128 and 64 neurons but val_accuracy stayed at 0.2.
Main reason is our neurons are too little to fit nonlinear mappings correctly.
So I increased neuron number to 1024,512,256, accuracy increased but still fluctuated at 0.6.
Batchsize too small may lead to local minimum.
Increased batch_size to 128 then max val_accuracy got over 0.7.

Answer 1.2 (2p)

Loading model from file

For each image I convert it to gray scale. The max accuracies both increases a lot.
The model is better. The main reason is the task is only number recognition.
The color information has no exact mapping with the number category and will make our model more complex and harder to fit in.
With gray scale image we can get the most useful information and make our model simpler and easier to fit.

Answer 1.3 (4p)

Loading model from file

Using l2 regularization for last two hidden layers can improve the performance a lot and better than using l2 for all hidden layers.
Adding batch normalization will make the model overfit much more.
In the learning curve accuracy and val_accuracy are closer which means less overfitting.

Answer 2.1 (7p)

Loading model from file

First make a vgg-like model with 6 conv layers which have 32,32,64,64,128,128 filters respectively.
When batchsize is 32 the max training accuracy is just 0.93 so chose to increase the batchsize to 128 then the max accuracy and val_accuracy are 0.986 and 0.932.

But with the learning curve model overfits heavily.

Begin regularization: first add droupot after each block with rate 0.2.The max val_accuracy comes to 0.940 and overfitting is lessened.

Gradually increase amount of droupout in our case doesn't help.

Then add batch normalization after each layer we can see the max val_accuracy comes to 0.946 but it overfits more so I decided not to add normalization.

Answer 2.2 (3p)

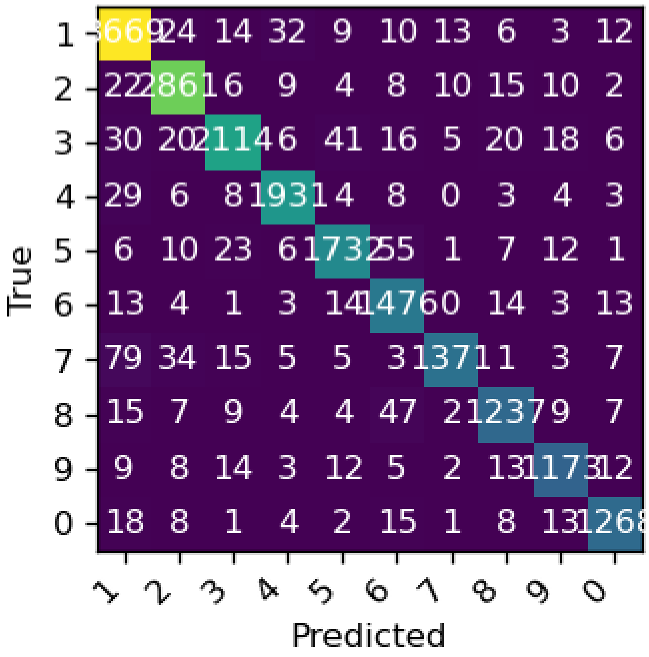
Loading model from file

Only slightly shift can increase our max val_accuracy. Other operations like rotation,sheer or zoom will make our model worse.

I think the main reason is that our images are quite low resolution and those operations will destroy much information.

Another thing is that in this model the val_accuracy is higher than training accuracy which may means this model has more ability of generalization.

Answer 3.1 (2p)



Predicted: 6,
Actual : 5

Predicted: 3,
Actual : 9

Predicted: 6,
Actual : 5

Predicted: 9,
Actual : 0

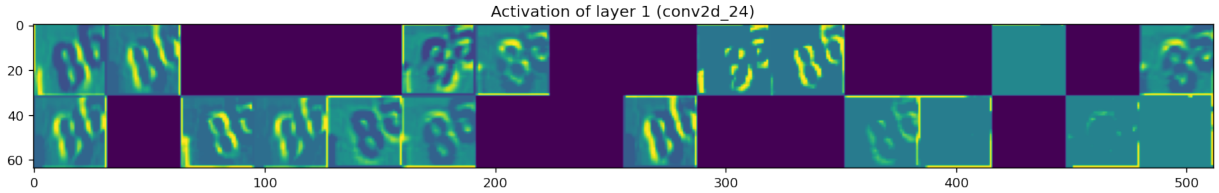
Predicted: 6,
Actual : 8

In the confused matrix we can find 1,3,6,8 are always confused.

With the images we can find noisiness is the main reason to error.

Answer 3.2 (4p)

```
D:\TUe\Q3_course\web\assignment-3-Yibin-Lei\assignment-3-Yibin-Lei\solution.py:671: RuntimeWarning:
 invalid value encountered in true_divide
```

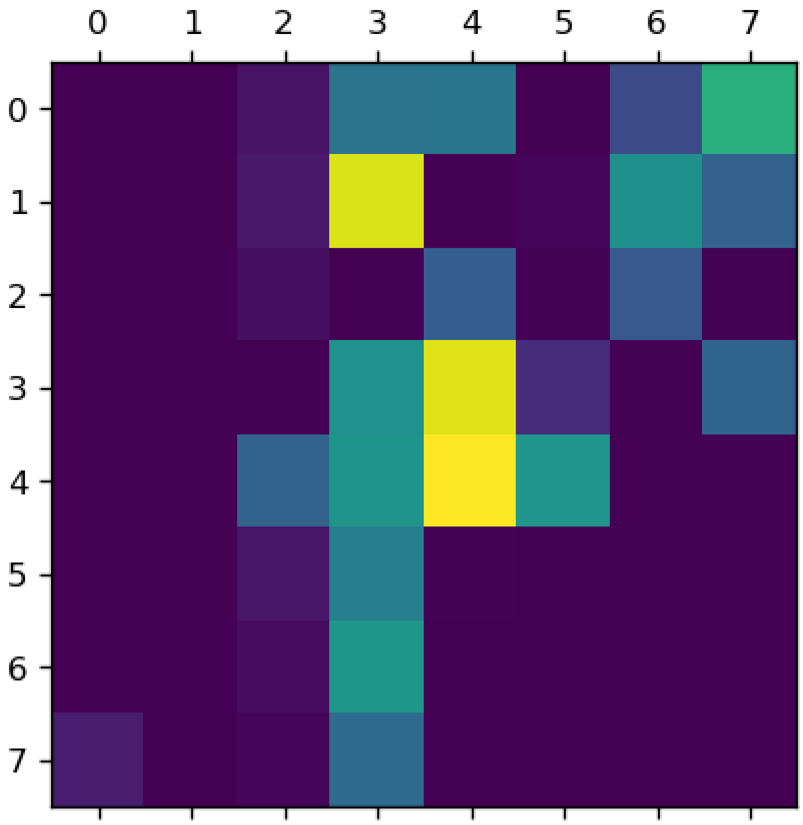


In the first two conv layers the model learns various edges of the number.

In the third and forth layers I find more abstract patterns and specific curves such as the circle of number '8'.

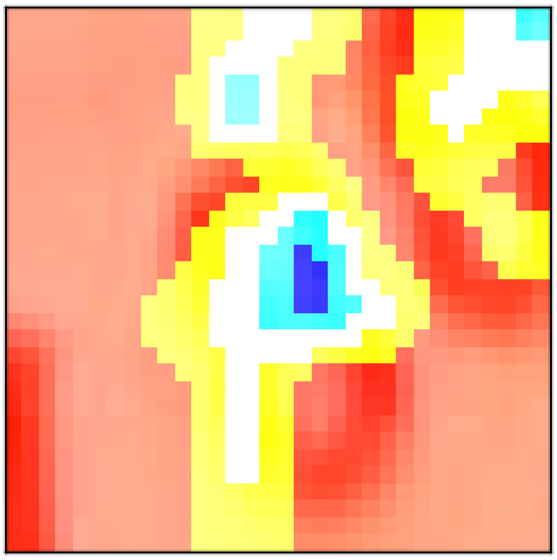
In the last two layers I see patterns becomes much more abstract. And empty filter activations means input does not have information that filter was interested in.

Answer 3.3 (4p)



```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

Class activation map



Answer 4.1 (5p)

Loading model from file

Freezing conv_base get a max val_accuracy 0.589.
Only unfreezing conv layers of block5 get a max val_accuracy 0.581 get a max val_accuracy 0.67. It improves a lot.
Then unfreezing conv lyers of block5 and block4,I get a max val_accuracy of 0.85 also increases a lot.
Then unfreezing conv layers of blocks 5, 4 and 3, the max val_accuracy comes to 0.918.
Finally unfreezing conv layers of blocks 5, 4, 3 and 2, the max val_accuracy drops to 0.58.
Hence unfreezing conv layers of blocks 5, 4 and 3 may be the best model.
In this way we can get a good fitted model using pretrained model.

Answer 4.2 (5p)

```
Pipeline(memory=None,
         steps=[('scaler',
                 StandardScaler(copy=True, with_mean=True, with_std=True)),
                ('reg',
                 LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                                  normalize=False))],
         verbose=False)
Evaluation: None

Running time: 32.25 seconds
Last modified: April 20, 2020
scikit-learn version: 0.22.2.post1
```

For this question verifying always gets error so I make two evaluate functions empty to run verify. The code I submit is not empty. Here below is my answer of this question.

For a linear regrssion model learned by our original data we coould only get a test accuracy for almost 0 but learned by embedding one I get a test accuracy of 0.815.
The embedding data has more useful information than original one so it performs really much better.
But our model can not beat neural networks I think the main reason is we have huge amount of parameters which can help us fit every nonlinear functions correctly which other non-deep learning methods can not reach.