

Assignment 1

Team # 4

Member: Wenjun song, Yanjun lyu, Haimin zhang

Problem 2 :

The parameters of Conv-2D layer include: filters, kernel_size, strides, padding, data_format, dilation_rate, activation function, use_bias, initializer (kernel_initializer, bias_initializer) regularizer (kernel_regularizer, bias_regularizer, activity_regularizer), constraint functions (kernel_constraint, bias_constraint). We focus on the main parameters, such as kernel_size, filters, padding and activation function, and we use the regularizer parameters to reduce the impact of overfitting. We leave some secondary parameters being default value, such as kernel initializer and bias_initializer which are rarely used in the network designing process.

We built 20 models to check how different parameters changing will impact the model. We set the model with base code as the base model. Based on that, we modified the parameters. Some models were just adjusted by single parameter, and some models were modified by two or more parameters, Appendix A is the parameters details of different models.

Considering the time for computing, firstly we selected the proper batch size being 512, and we set a model with epochs=100, which can help us to observe the convergence of the base model. From Figure 1, we selected epochs=30 to build most of the trial models.

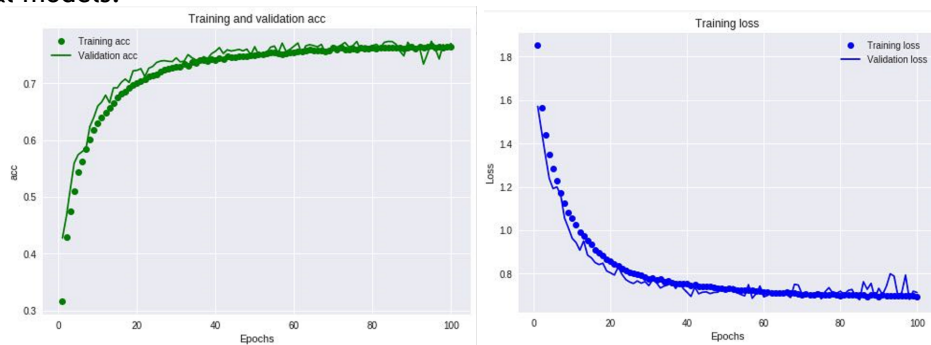


Figure 1: Base model with epochs=100

We changed the value of padding to be "Valid" which means not to detect the images' border. Obviously, it will influence the accuracy of this case and the trial result proved this, the accuracy dropped to 0.4359 from 0.6921. And we changed the value of pooling size to be 4, which dragged down the accuracy a little bit. And we compared the model with pooling_size=3, the accuracy is nearly but lower than the base model. So the pooling size can influence the model, and the more the value of pooling size is, the more information of images will lose, which will influence the quality of the model. It depends on the realistic problem, we should make trade-off to decide the value of pooling model.

We modified the value of filters for different convolutions, and we built three models to test the impact of filters. From state 3, state 4, and state 5, we can see the all the values of accuracy are lower than base model. The number of filters can determine how well the features we can get from the convolutional process. More features come along with more filters, but it can influence the complexity of models. Also, we should make trade-off to choose the filters. Then we altered the number of convolutional layers, we tried 6 and 10 convolutional layers to observe the impact. From state 7 to state 9, we can get the results of accuracy being much lower than base model, which means the more convolutional layers will influence the model quality a lot, so we should select the layers number carefully. And the state 9 with 10 layers generated accuracy being 0.1, which means the model is not suitable for this case, the learning process will be broken.

Besides, the kernel size is another important factor for the models, so we made state 6 with the kernel size of 4*4 which give out a litter better accuracy and loss value for the new model. When we want to use larger size for this case, such as 8, the result is

worse than 4×4 . So when we design a neural network, we should carefully select the kernel size as well. Not only the model quality it determines, but also it will influence the computing time.

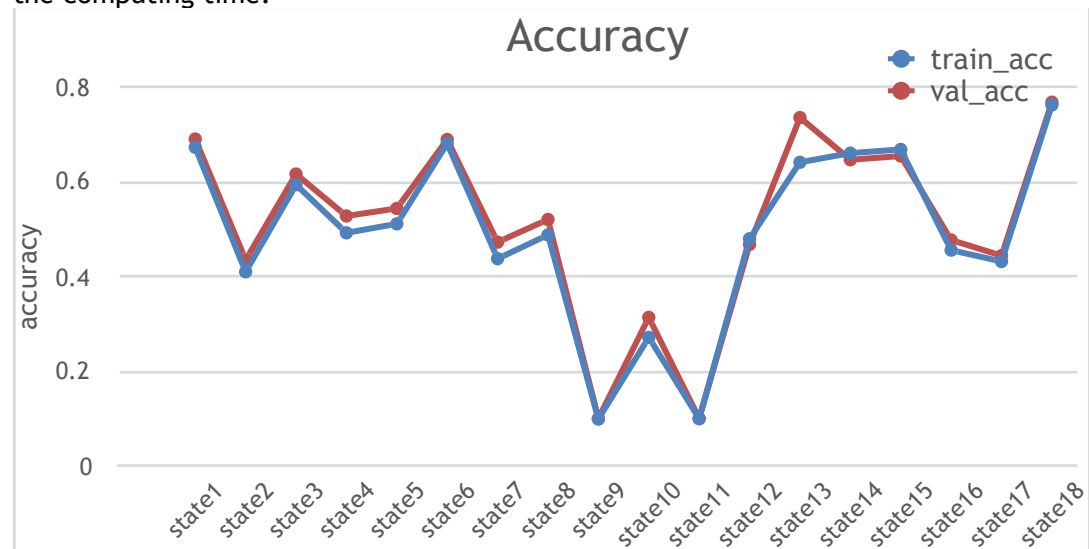


Figure 2: Accuracy for trials

Then we moved to the activation function, we put “softplus” functions being the convolutional layers functions, and we didn’t change the MLP parameters. The result shows us “relu” function is much better for this case. When we put activation=softplus and kernel_size=8 into one basket, the accuracy turned to be 0.1, which also means not suitable for this case.

After that, we chose the epochs=50 to do more trials, which gave out better output than epochs=30. Based on the base model, we got a higher value for accuracy(0.7376), and for other states that altered single parameter based on the original model, the results are obviously better than the models with epochs=30. The model of epochs=100 using double time than epochs=50 generated a better value of accuracy and loss, from 0.7376 to 0.7701. For the image classification problem, we think it worth the more cost of time and money.

At last, we analyzed the regularizers. We generated an overfitting model- state 19, and we modified the value of kernel_regularizer, bias_regularizer, activity_regularizer to reduce the impact of overfitting. The output proved that the parameters worked well for this job.