A01706648 Wenguang Hu

Analyze and Report on the performance of model.

We are trying to create a model to predicate the class of the wine, based on wine's composition.

So, the input data will be wine's composition, the target data is wine's class.

We read the data from csv file, then randomly select 20% of the data set as Test Data Set.

The reset of the data will be used for Training. Meanwhile Training, we take 15% of the Training Data for the Validation, in order to check the model training is correct or not.

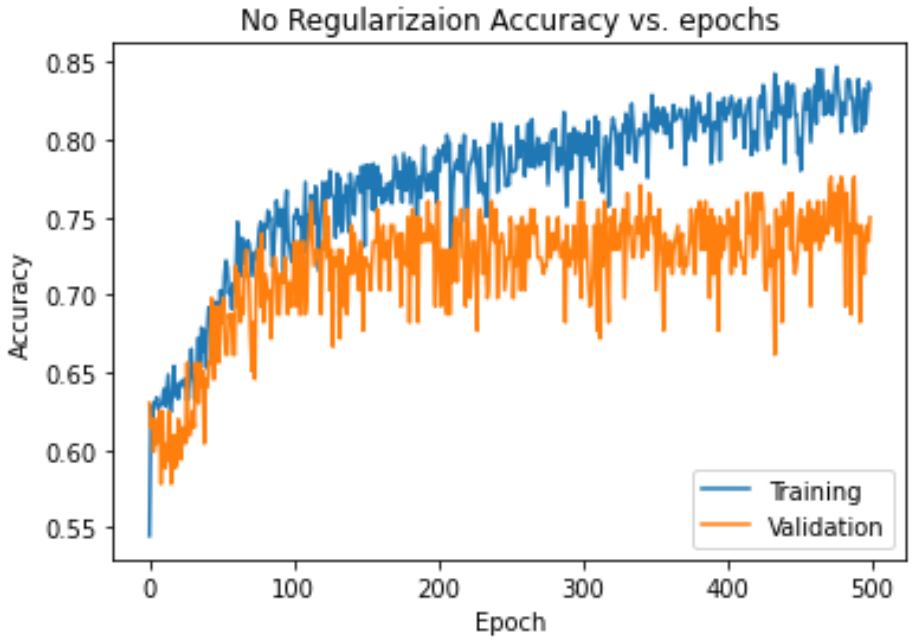
In the beginning, we created a complex model in order to be able to handle most of the complexity of the system. It is like below

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_30 (Dense)            (None, 64)                768

 dense_31 (Dense)            (None, 128)               8320

 dense_32 (Dense)            (None, 128)               16512

 dense_33 (Dense)            (None, 128)               16512

 dense_34 (Dense)            (None, 128)               16512

 dense_35 (Dense)            (None, 64)                8256

 dense_36 (Dense)            (None, 64)                4160

 dense_37 (Dense)            (None, 64)                4160

 dense_38 (Dense)            (None, 64)                4160

 dense_39 (Dense)            (None, 1)                 65

=================================================================
Total params: 79,425
Trainable params: 79,425
Non-trainable params: 0
_________________________________________________________________
```

A01706648 Wenguang Hu

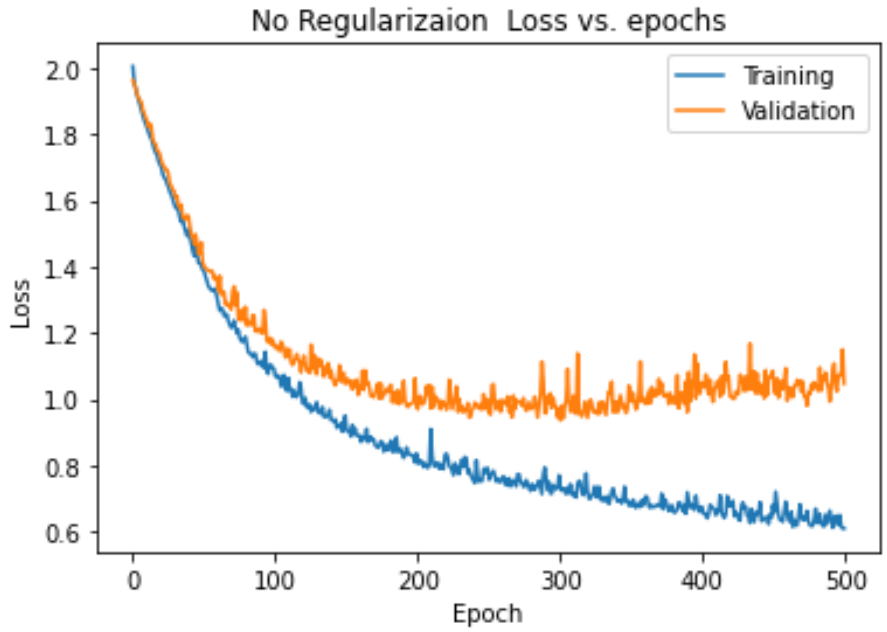
After feed the training data and validation data to train for 500 epochs

We are able to get the chart about Train Data and Validation Data's Accuracy.



Before 100 Epochs, Training and Validation's accuracy are almost same, in this stage, the model is improving to trying to fix generally.

After 100 Epochs, we notice that the Training's accuracy is higher and higher than Validation's accuracy, that means the model is trying to specifically fit the Training Data, and not fit the for general anymore.



This situation also appeared in the chart for Loss.

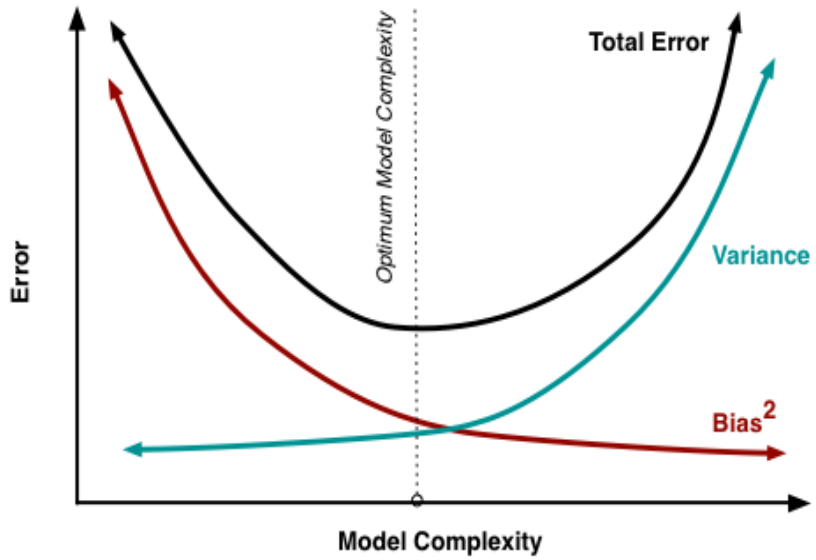
The loss of both Train and Validation decreased almost in the same rate before 100 Epochs, it means the system is fitting to general situation, to reduce the Error from the Expected result.

But after 100, the loss Training is continuously decrease, but the Validation's loss starts to increase. That indicates the system is trying to fit the Training Data specifically, not to fit the general situation.

This situation of model fitting the training data too much, not for general, and it is a too Complex system, we consider it as an OVER FIT

Since we have a very High Accuracy of both Train and Validation, it means we have a Low Error Rate, it means we have LOW BIAS.

Because we have an OVER FIT system, and it is a too Complex System, so we are having a HIGH VARIANCE.



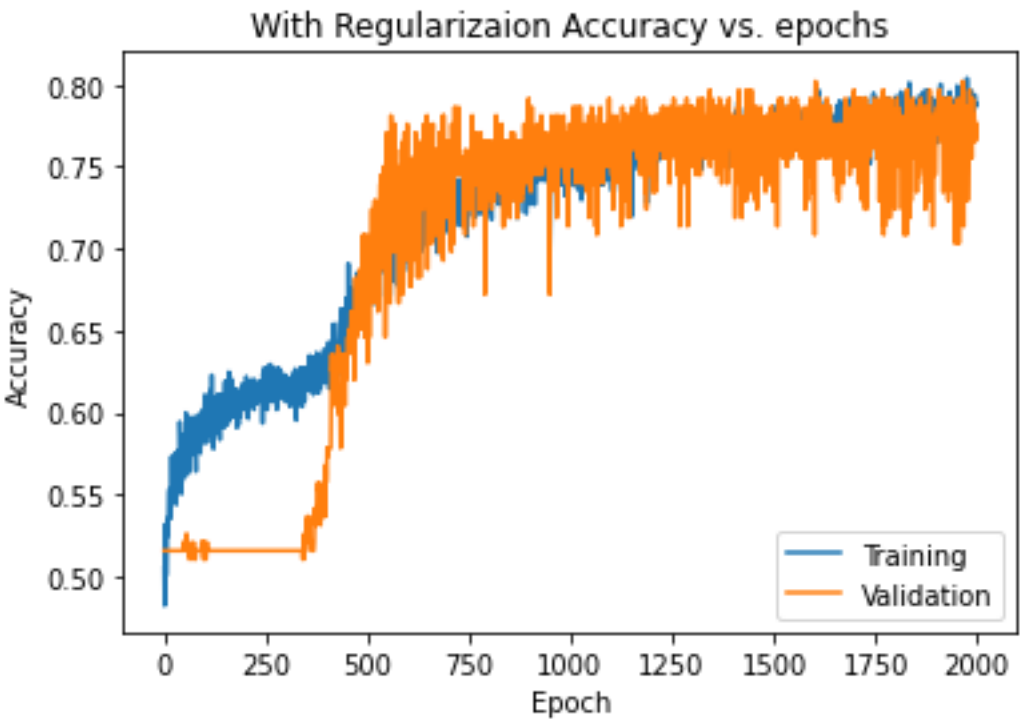
Because we have an OVER FIT system, so we add addition of two dropout layers, weight decay (l2 kernel regularization), and a batch normalization layer, to make a regularized model.

A01706648 Wenguang Hu

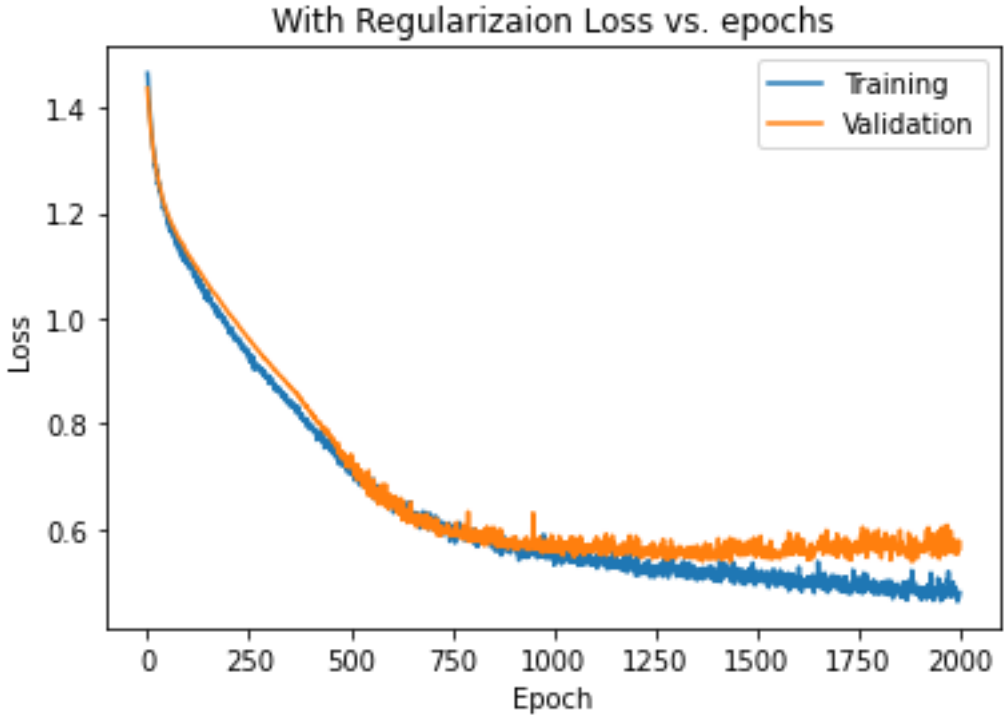
Layer (type)	Output Shape	Param #
dense_50 (Dense)	(None, 64)	768
dense_51 (Dense)	(None, 128)	8320
dense_52 (Dense)	(None, 128)	16512
dropout_4 (Dropout)	(None, 128)	0
dense_53 (Dense)	(None, 128)	16512
dense_54 (Dense)	(None, 128)	16512
batch_normalization_2 (BatchNormalization)	(None, 128)	512
dense_55 (Dense)	(None, 64)	8256
dense_56 (Dense)	(None, 64)	4160
dropout_5 (Dropout)	(None, 64)	0
dense_57 (Dense)	(None, 64)	4160
dense_58 (Dense)	(None, 64)	4160
dense_59 (Dense)	(None, 1)	65

After training for 1000 epochs,

We can get the chart about Train Data and Validation Data's Accuracy.



Before 300 Epochs, Training's accuracy is more than Validation's it was overfit, but after this stage, due to the dropout layer start to drop out the input unit, the model is simpler, and start to be fitted. We can see the Accuracy of Training is almost the same as validation's accuracy.



A01706648 Wenguang Hu

From the loss chart, we can see the before 1500, most of the time, the loss of Train is same as loos of Validation, at that time, the model covers both Train Data and Validation Data.

And in the very end, the loss of Training is slightly lower than the Validation, so the system has slightly overfit, but basically, it is FITTED.

Since we have a very High Accuracy of both Train and Validation, it means we have a Low Error Rate, it means we have LOW BIAS.

Because we have a FITTED system, and it is Normal System, so we are having a MIDIEAN VARIANCE.

The evaluation of system without regularization,

```
Test loss: 0.963
Test accuracy: 74.69%
```

The evaluation of system with regularization,

```
Test loss: 0.530
Test accuracy: 76.56%
```

So we slight increase the accuracy by 1.9%, and decrease the Test loss by 0.37