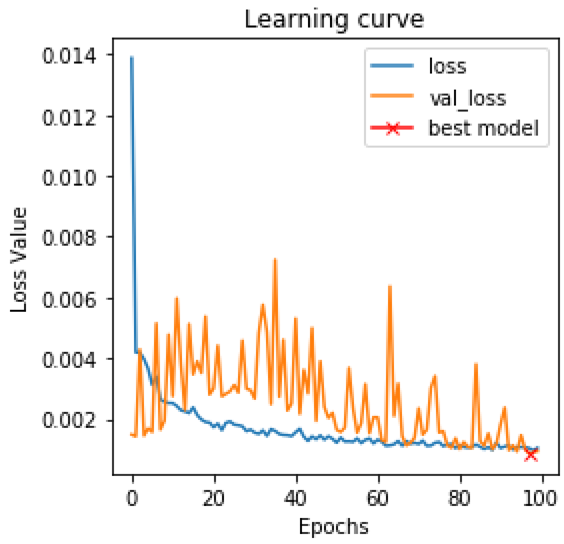
Students: Blanca Cabrera Gil & Laila Niazy
Course: CM2003 Deep Learning Methods for Medical Image Analysis

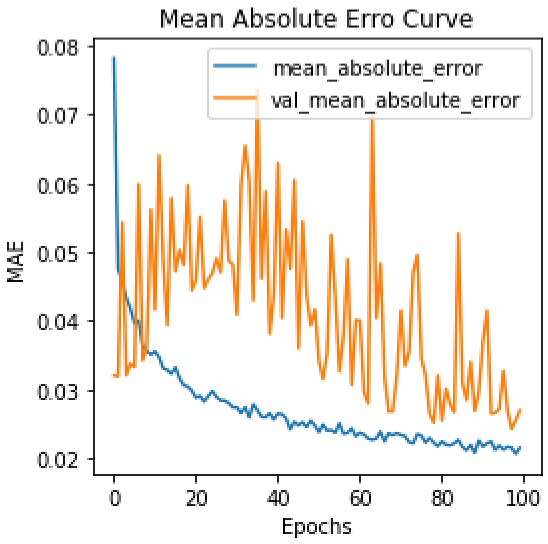
Laboratory Assignment 5

Task 1: Regression problem – prediction of stock prices

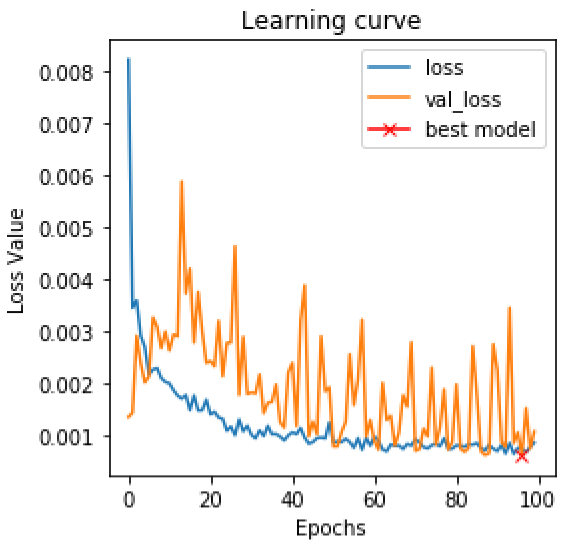
How good is your model? Can you observe any difference in performance if you increase (e.g. double) the number of output units of the LSTM layers?

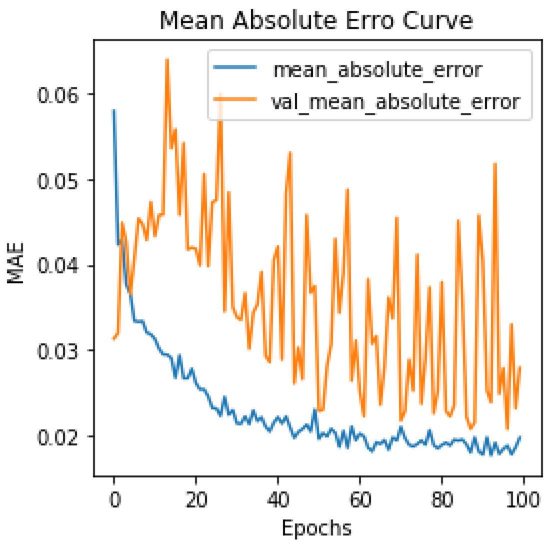
The first two figures seen are produced using 20 units of the LSTM layers. As we can see, the loss value is decreasing with each epoch for the training data, which tell us the model is learning and the same goes for the mean absolute error. However, for the validation data the loss curve is decreasing but is also oscillating a lot and the same goes for the mean absolute error curve. Since the values of the learning curve for both training and validation set are very close to each other, we can say the model is not over - or underfitting. Further, the model is performing really well since it reaches a very low loss value after 100 epochs and a very low mean absolute error.





Now for the next two figures below, they were produced with 40 units of the LSTM layers. We can observe, that the start value of the learning curve is much lower for this case compared to the figure with 20 units and reached a slightly lower loss value after 100 epochs. However, the learning curve is definitely not as smooth as the one above. The same observations are made with the mean absolute error curve.

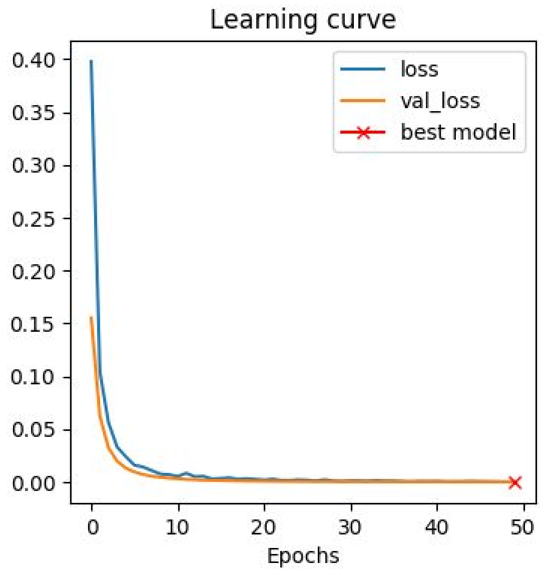


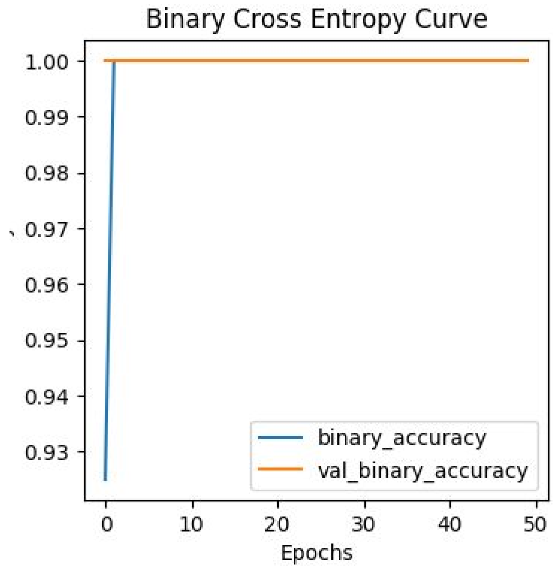


Task 2: Classification problem – fiber bundle classification

To perform the classification problem binary cross-entropy is used as cost function and binary accuracy as metric. After training the model for 50 epochs with 10 hidden units the obtained results are the following:

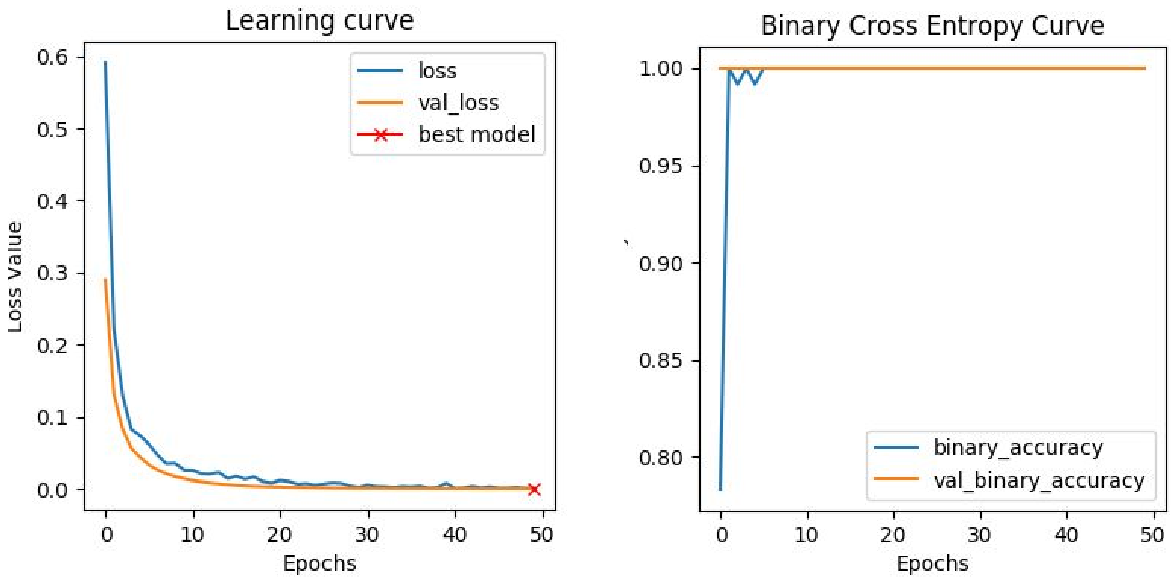
```
Epoch 50/50
120/120 [==============================] - 5s 45ms/step - loss: 2.0787e-04 -
binary_accuracy: 1.0000 - val_loss: 6.3461e-05- val_binary_accuracy: 1.0000
```





When decreasing the number of hidden units to 5 the following results are obtained:

```
Epoch 50/50
120/120 [==============================] - 6s 46ms/step - loss: 0.0012 -
binary_accuracy: 1.0000 - val_loss: 1.5258e-04 - val_binary_accuracy: 1.0000
```

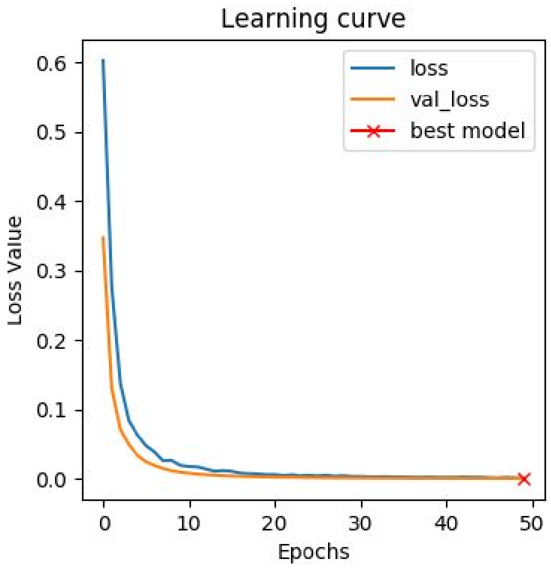


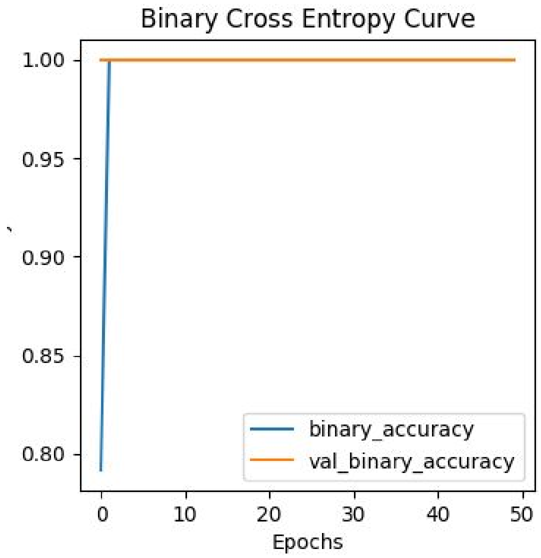
When decreasing the number of hidden units the model needs more training to reach the loss values achieved with the higher number of hidden units.

Bonus task 1:

After zero-padding the training and test input data and adding a mask layer with mask value 0, these are the obtained results after running for 50 epochs with bach size 2:

```
Epoch 50/50
60/60 [==============================] - 12s 204ms/step - loss: 9.4168e-04 -
binary_accuracy: 1.0000 - val_loss: 3.6465e-04 - val_binary_accuracy: 1.0000
```

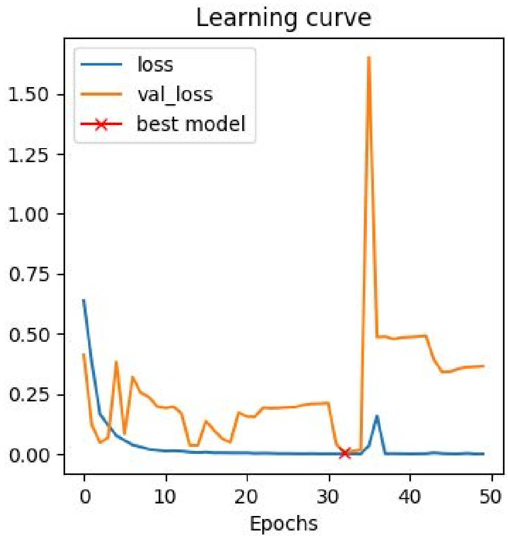


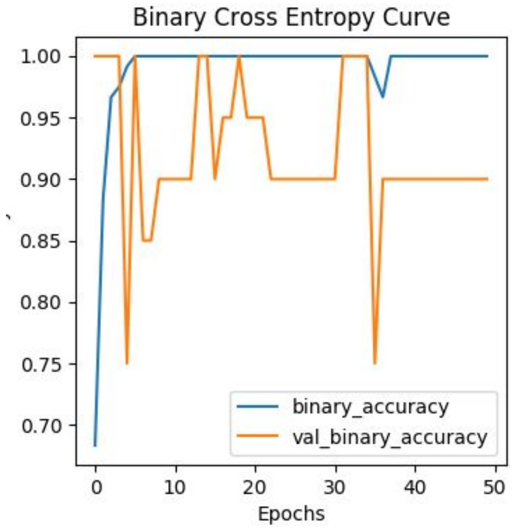


Bonus task 2:

When training the model again for Fornix bundle the following learning parameters are obtained:

```
Epoch 50/50
60/60 [==============================] - 7s 115ms/step - loss: 0.0015 -
binary_accuracy: 1.0000 - val_loss: 0.3663 - val_binary_accuracy: 0.9000
```



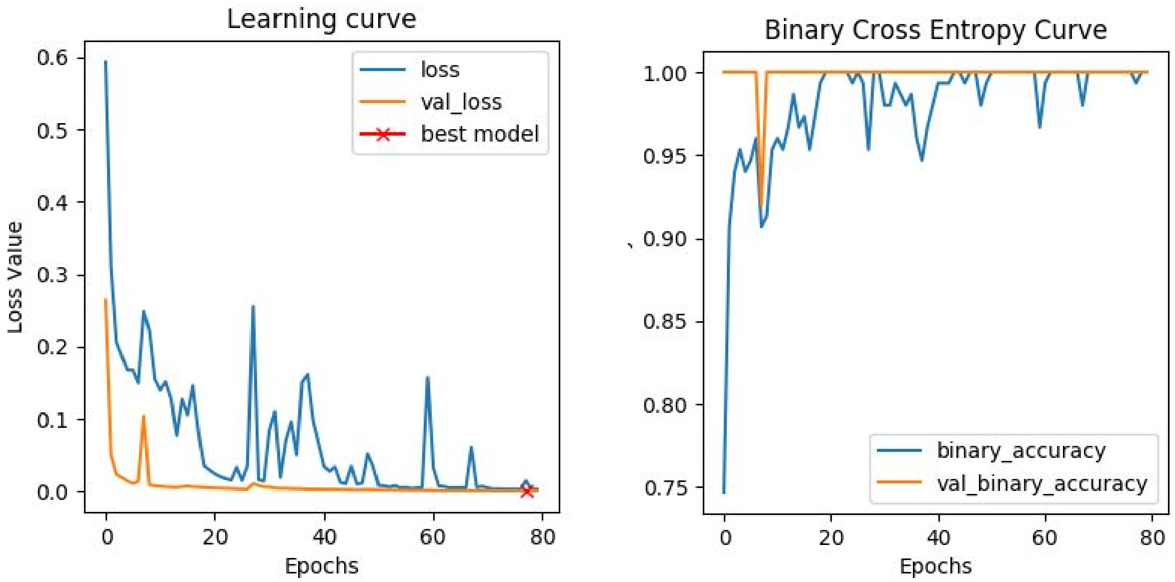


The model was trained a second time with different parameter setting to try find an optimal setting.

- #bundles =25
- Epochs = 80
- Units = 10

- Dropout rate = .02

```
Epoch 80/80
75/75 [==============================] - 9s 125ms/step - loss: 0.0026 -
binary_accuracy: 1.0000 - val_loss: 4.4810e-04 - val_binary_accuracy: 1.0000
```

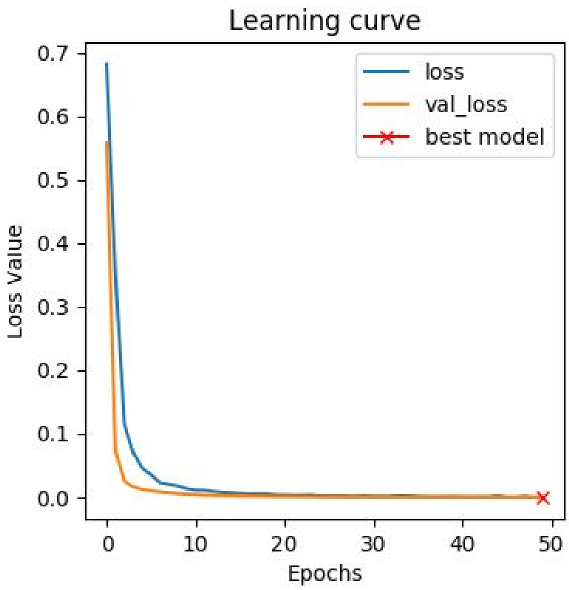


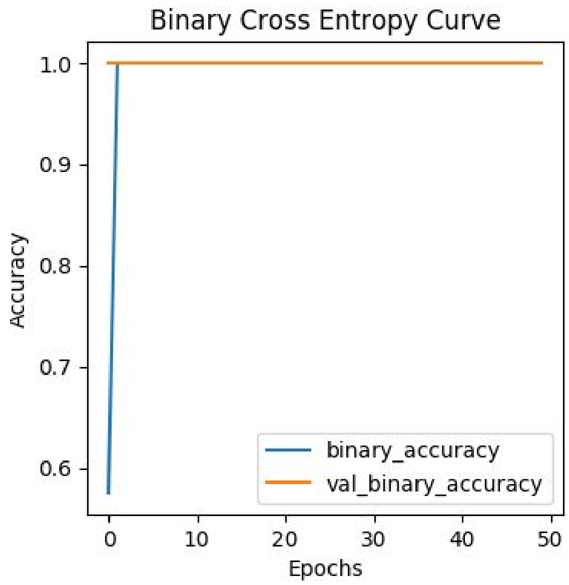
After training the model for Fornix bundle, it has been observed that the more bundles are considered, the more difficult is to train the model.

Finally, we train the model for Striato-fronto-orbital bundle. In this case, we train the model with the given configuration and we compare it to the training with other parameters to see the effects.

10 hidden units

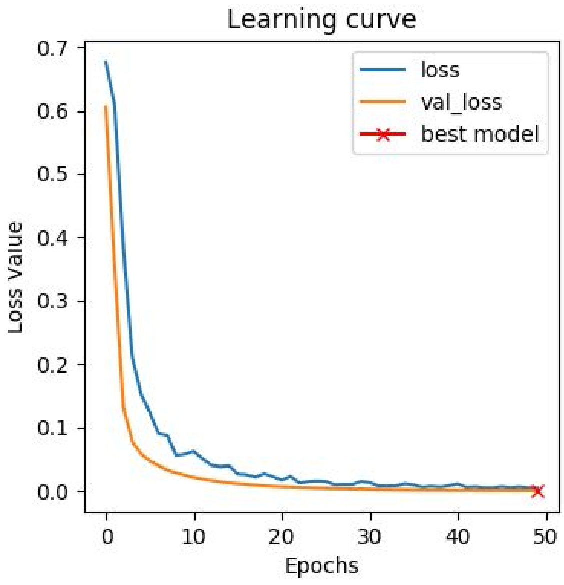
```
Epoch 50/50
60/60 [==============================] - 6s 107ms/step - loss: 8.0105e-04 -
binary_accuracy: 1.0000 - val_loss: 1.7330e-04 -val_binary_accuracy: 1.0000
```

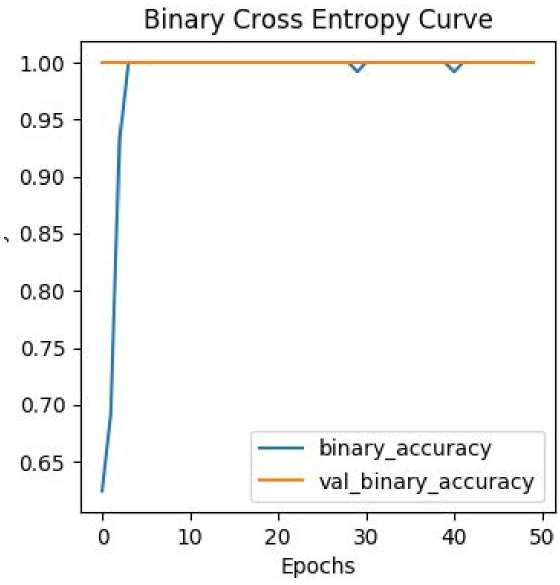




5 hidden units

```
Epoch 50/50
60/60 [==============================] - 7s 116ms/step - loss: 0.0046 - binary_accuracy: 1.0000 - val_loss: 8.0711e-04 - val_binary_accuracy: 1.0000
```





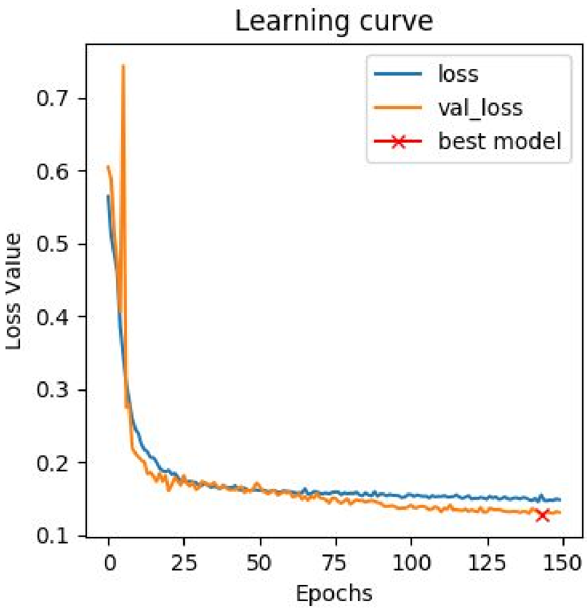
When testing different numbers of units, it can be seen that with a higher number of units the network learns faster. Yet, one must be careful of not overfitting the model.

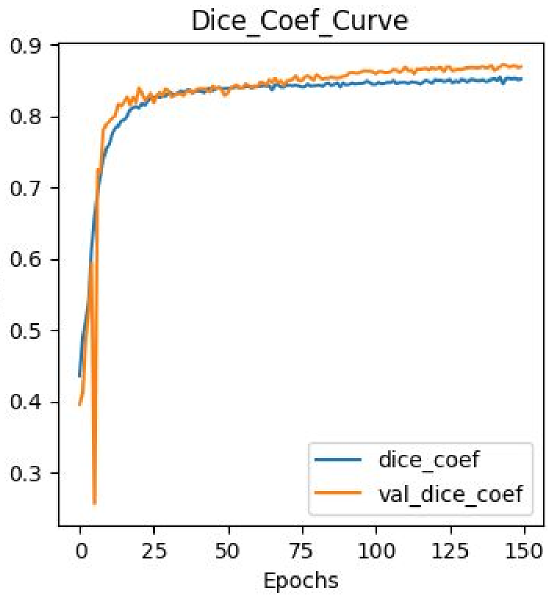
Task 3: Another U-Net extension

The following parameters were used to generate all figures below:

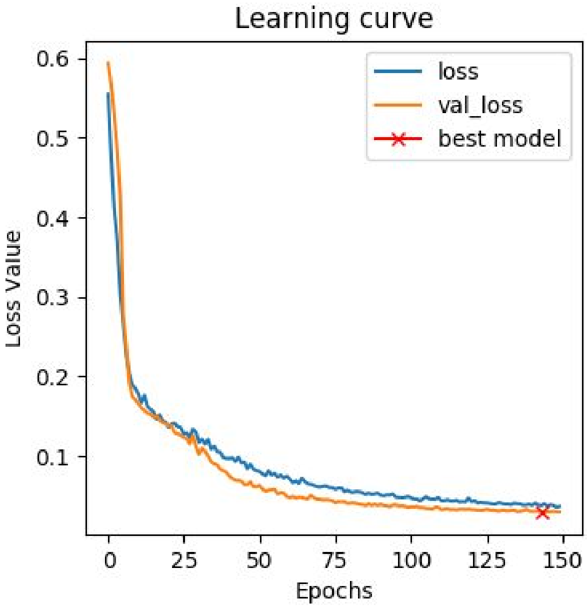
- image_size = 128 #both width and height of image are the same
- batch_size = 2
- LR = 0.0001
- SDRate = 0.5
- spatial_dropout = True
- epochs = 150
- base = 16
- batch_normalization = True

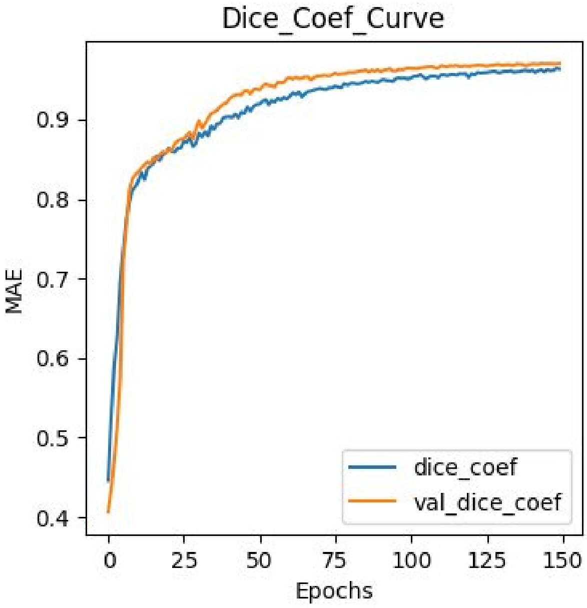
For the first two figures we used Data Augmentation and can seen the loss value is decreasing for both training and test set, which proves that the model is learning in each epoch.





In the other two figures below, we did not use Data Augmentation and we can see that the model is performing better, reaching almost 1.0 as a dice coefficient and almost 0.0 as a loss value after 150 epochs.


Learning curve
loss
val_loss
best model
Loss Value
Epochs


Dice_Coef_Curve
dice_coef
val_dice_coef
MAE
Epochs