Optimisation for training deep neural networks

```
#Building a Sequential Feed Forward Network in Keras

from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(256, activation='relu', input_shape=(new_dimension,)))
model.add(Dense(128, activation='relu'))
model.add(Dense(no_labels, activation='softmax'))

model.compile(optimizer='adam', loss=tf.keras.losses.categorical_crossentropy,
              metrics=['accuracy'])
history = model.fit(x_train, y_train, validation_data=(x_val, y_val), epochs=20, batch_size=1

test_loss, test_accuracy = model.evaluate(test_images, y_test)
print('Test loss: {}'.format(test_loss))
print('Test accuracy: {}'.format(test_accuracy))
```

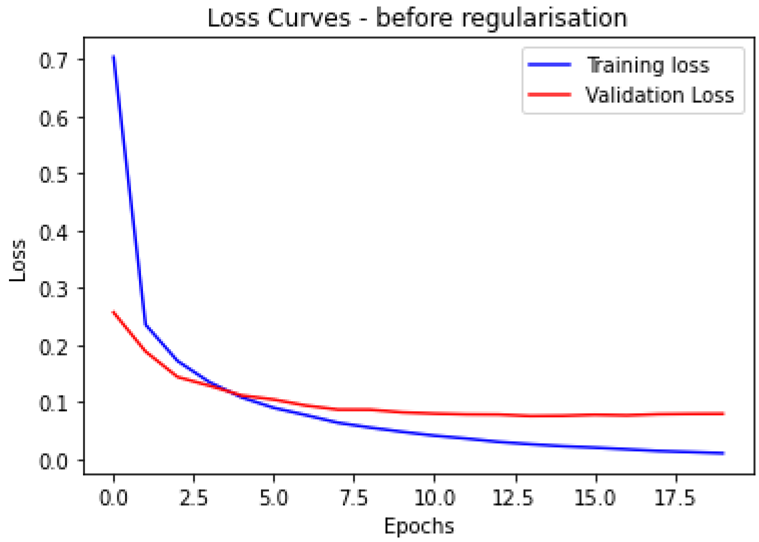
```
Train on 50000 samples, validate on 10000 samples
Epoch 1/20
50000/50000 [==============================] - 2s 32us/sample - loss: 0.7043 - acc: 0.8
Epoch 2/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.2356 - acc: 0.9
Epoch 3/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.1718 - acc: 0.9
Epoch 4/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.1346 - acc: 0.9
Epoch 5/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.1080 - acc: 0.9
Epoch 6/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0898 - acc: 0.9
Epoch 7/20
50000/50000 [==============================] - 1s 30us/sample - loss: 0.0770 - acc: 0.9
Epoch 8/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.0636 - acc: 0.9
Epoch 9/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0551 - acc: 0.9
Epoch 10/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.0476 - acc: 0.9
Epoch 11/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.0409 - acc: 0.9
Epoch 12/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0358 - acc: 0.9
Epoch 13/20
50000/50000 [==============================] - 2s 30us/sample - loss: 0.0300 - acc: 0.9
Epoch 14/20
50000/50000 [==============================] - 1s 30us/sample - loss: 0.0259 - acc: 0.9
Epoch 15/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0224 - acc: 0.9
Epoch 16/20
```

```
50000/50000 [==============================] - 2s 30us/sample - loss: 0.0200 - acc: 0.99
Epoch 17/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0170 - acc: 0.99
Epoch 18/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0140 - acc: 0.99
Epoch 19/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0121 - acc: 0.99
Epoch 20/20
50000/50000 [==============================] - 2s 31us/sample - loss: 0.0101 - acc: 0.99
10000/10000 [==============================] - 1s 59us/sample - loss: 0.0718 - acc: 0.97
Test loss: 0.0718050493865041
Test accuracy: 0.9799000024795532
```

Visualize Result

```
plt.figure()
plt.plot(history.history['loss'], 'blue')
plt.plot(history.history['val_loss'], 'red')
plt.legend(['Training loss', 'Validation Loss'])
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curves - before regularisation')
```

```
Text(0.5, 1.0, 'Loss Curves - before regularisation')
```

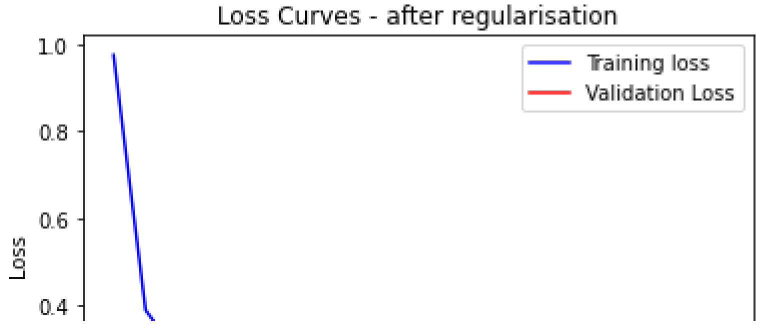


Train Again

```
#Although the validation and training loss seem great, we can see that the validation #loss i
#This identifies overfitting in our network. How do we proceed? #Introduce regularisation to

from tensorflow.keras.layers import Dropout
reg_model = Sequential()
reg_model.add(Dense(256, activation='relu', input_shape=(new_dimension,)))
reg_model.add(Dropout(0.4))
reg_model.add(Dense(128, activation='relu'))
reg_model.add(Dropout(0.4))
reg_model.add(Dense(no_labels, activation='softmax'))

reg_model.compile(optimizer='adam', loss=tf.keras.losses.categorical_crossentropy,
              metrics=['accuracy'])

reg_history = reg_model.fit(x_train, y_train, validation_data=(x_val, y_val),
                            epochs=20, batch_size=1000)
test_loss, test_accuracy = reg_model.evaluate(test_images, y_test)
print('Test loss: {}'.format(test_loss))
print('Test accuracy: {}'.format(test_accuracy))

test_loss, test_accuracy = reg_model.evaluate(test_images, y_test)
print('Test loss: {}'.format(test_loss))
print('Test accuracy: {}'.format(test_accuracy))

plt.figure()
plt.plot(reg_history.history['loss'], 'blue')
plt.plot(reg_history.history['val_loss'], 'red')
plt.legend(['Training loss', 'Validation Loss'])
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curves - after regularisation')
```

```
Train on 50000 samples, validate on 10000 samples
Epoch 1/20
50000/50000 [==============================] - 2s 39us/sample - loss: 0.9748 - acc: 0.69
Epoch 2/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.3899 - acc: 0.88
Epoch 3/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.2900 - acc: 0.91
Epoch 4/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.2396 - acc: 0.92
Epoch 5/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.2066 - acc: 0.93
Epoch 6/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1767 - acc: 0.94
Epoch 7/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1622 - acc: 0.95
Epoch 8/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1474 - acc: 0.95
Epoch 9/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1325 - acc: 0.96
Epoch 10/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1233 - acc: 0.96
Epoch 11/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1132 - acc: 0.96
Epoch 12/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.1052 - acc: 0.96
Epoch 13/20
50000/50000 [==============================] - 2s 37us/sample - loss: 0.0994 - acc: 0.97
Epoch 14/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0939 - acc: 0.97
Epoch 15/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0855 - acc: 0.97
Epoch 16/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0831 - acc: 0.97
Epoch 17/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0786 - acc: 0.97
Epoch 18/20
```

```
Epoch 19/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0691 - acc: 0.97
Epoch 20/20
50000/50000 [==============================] - 2s 36us/sample - loss: 0.0668 - acc: 0.97
10000/10000 [==============================] - 1s 60us/sample - loss: 0.0692 - acc: 0.97
Test loss: 0.06924701468222774
Test accuracy: 0.9789000153541565
10000/10000 [==============================] - 1s 63us/sample - loss: 0.0692 - acc: 0.97
Test loss: 0.06924701468222774
Test accuracy: 0.9789000153541565
Text(0.5, 1.0, 'Loss Curves - after regularisation')
```


Loss Curves - after regularisation
Training loss
Validation Loss
Loss