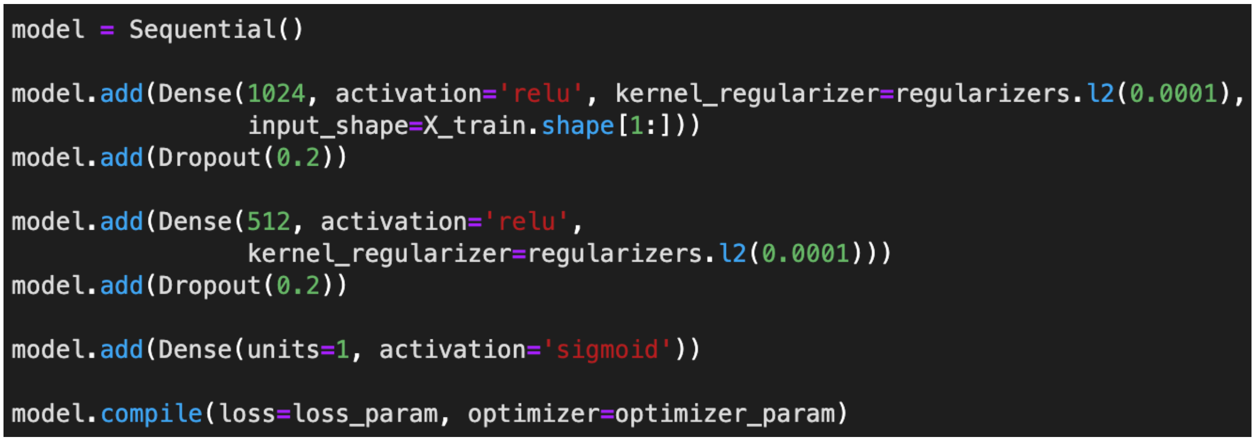
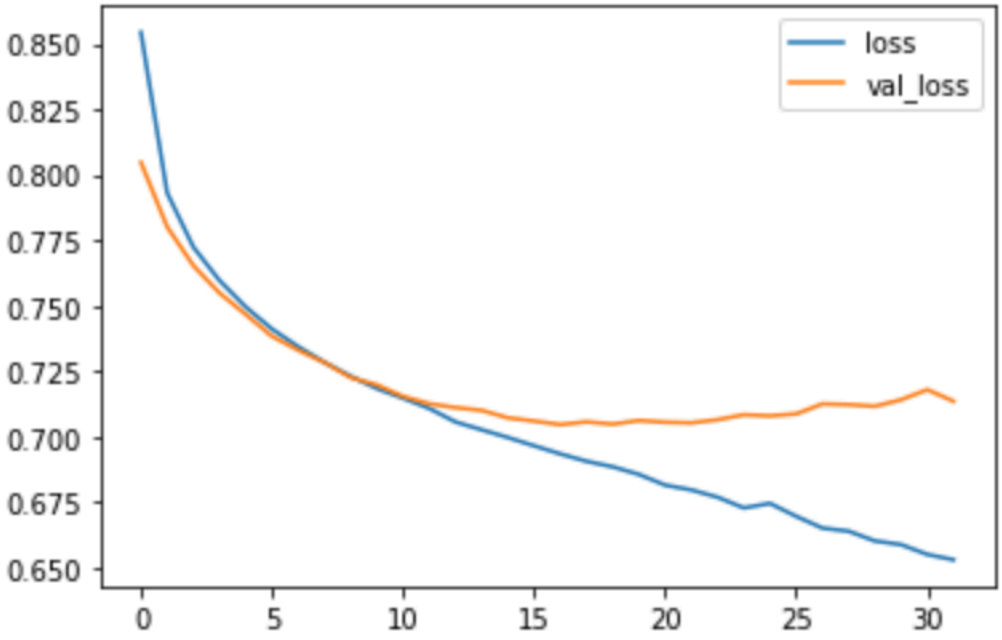
Model 1

Mean Loss:	0.705395
Mean Validation Loss:	0.721903
Batch Size:	5096

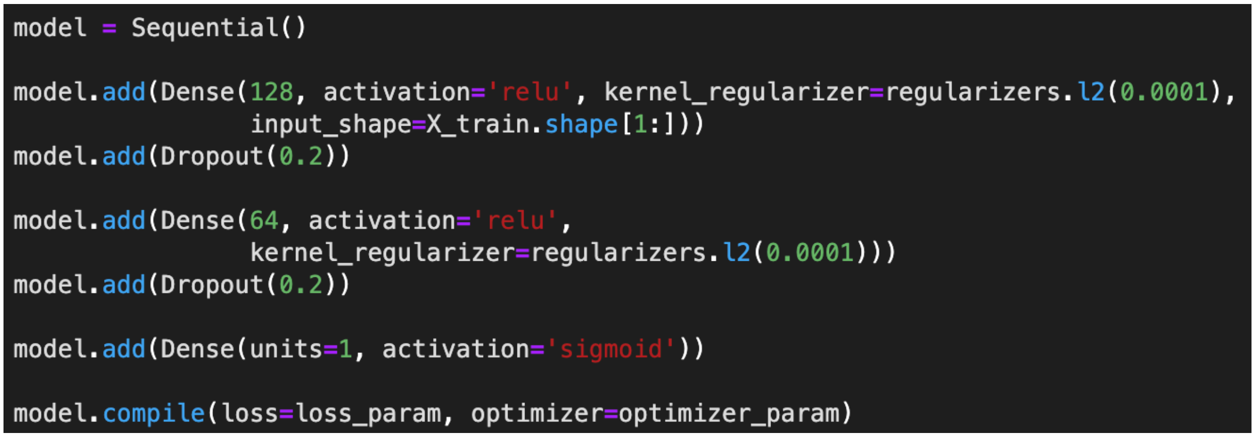


```
model = Sequential()

model.add(Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.2))

model.add(Dense(512, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.2))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

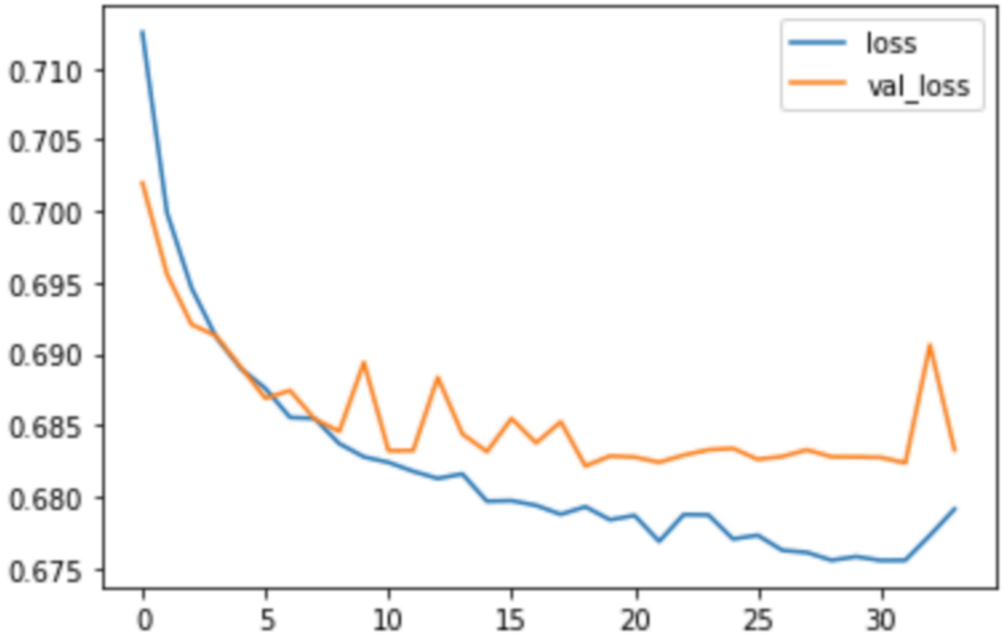


Model 2

Mean Loss: 0.682143
Mean Validation Loss: 0.690175
Batch Size: 2048

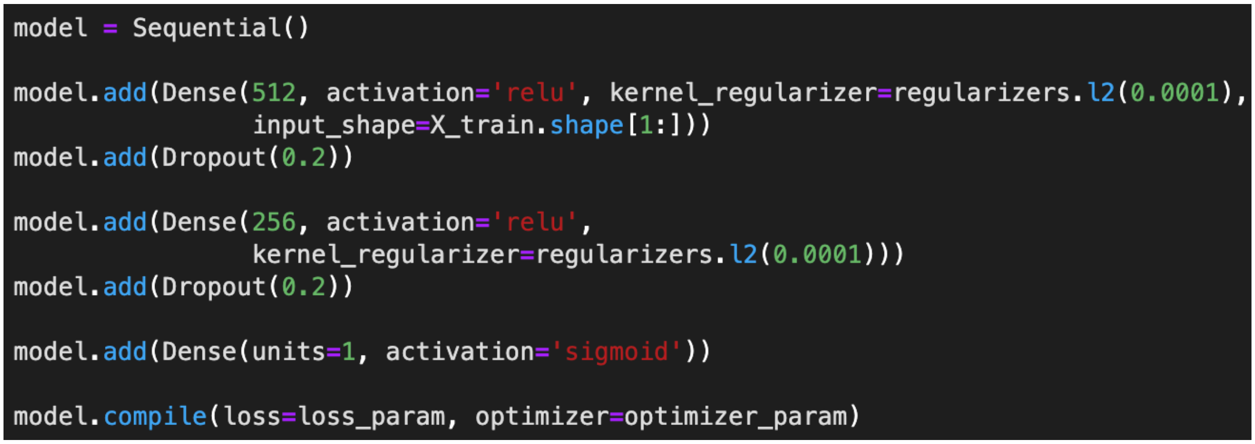


```
model = Sequential()

model.add(Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.2))

model.add(Dense(64, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.2))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

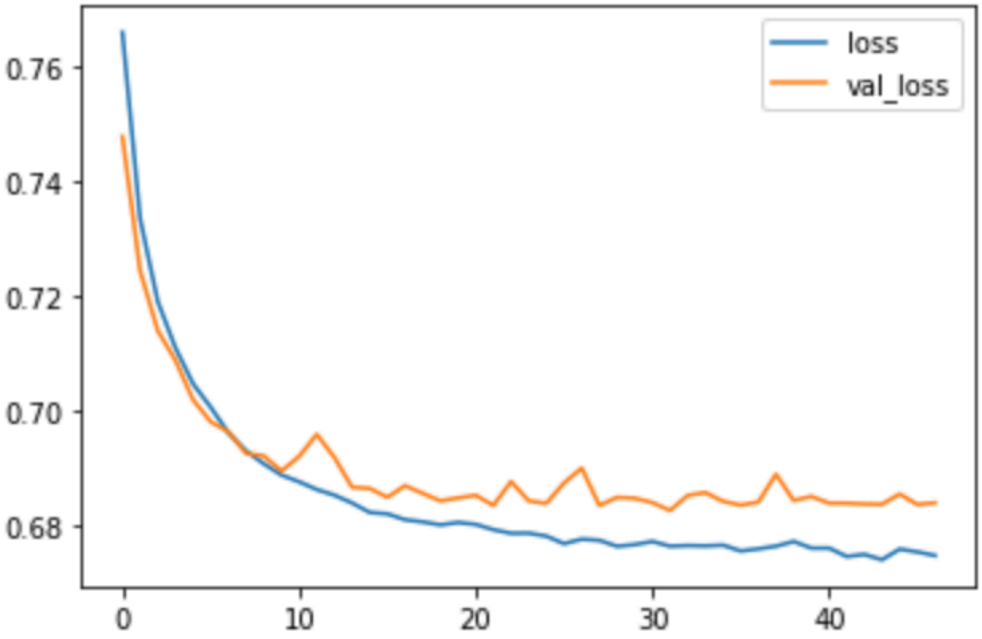


Model 3

Mean Loss:	0.685215
Mean Validation Loss:	0.690175
Batch Size:	2048

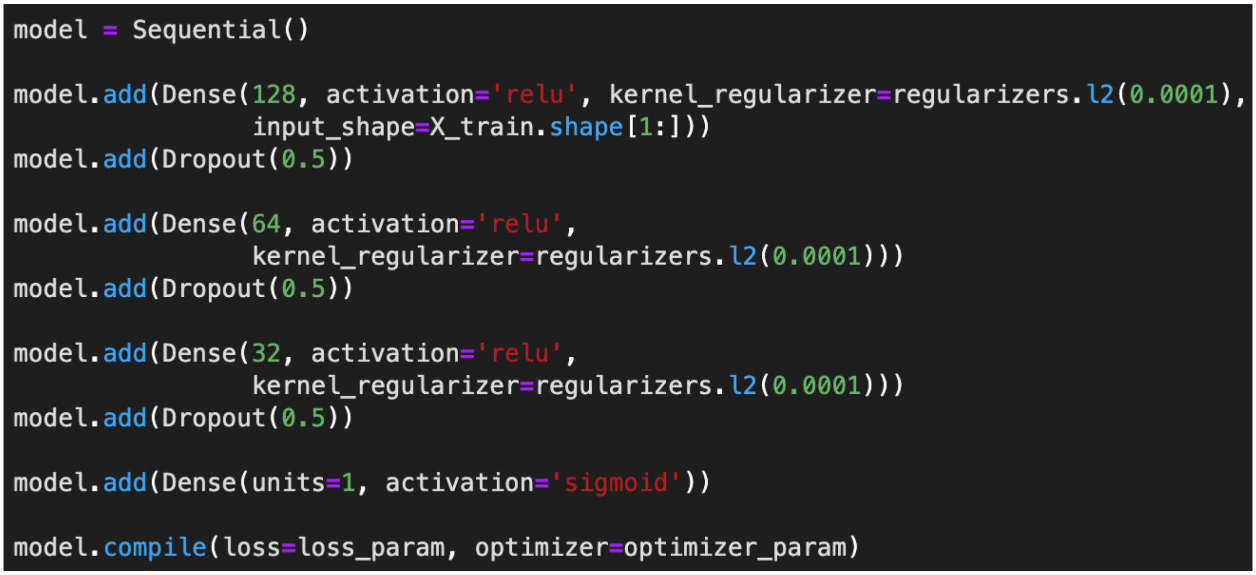


```
model = Sequential()

model.add(Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.2))

model.add(Dense(256, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.2))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

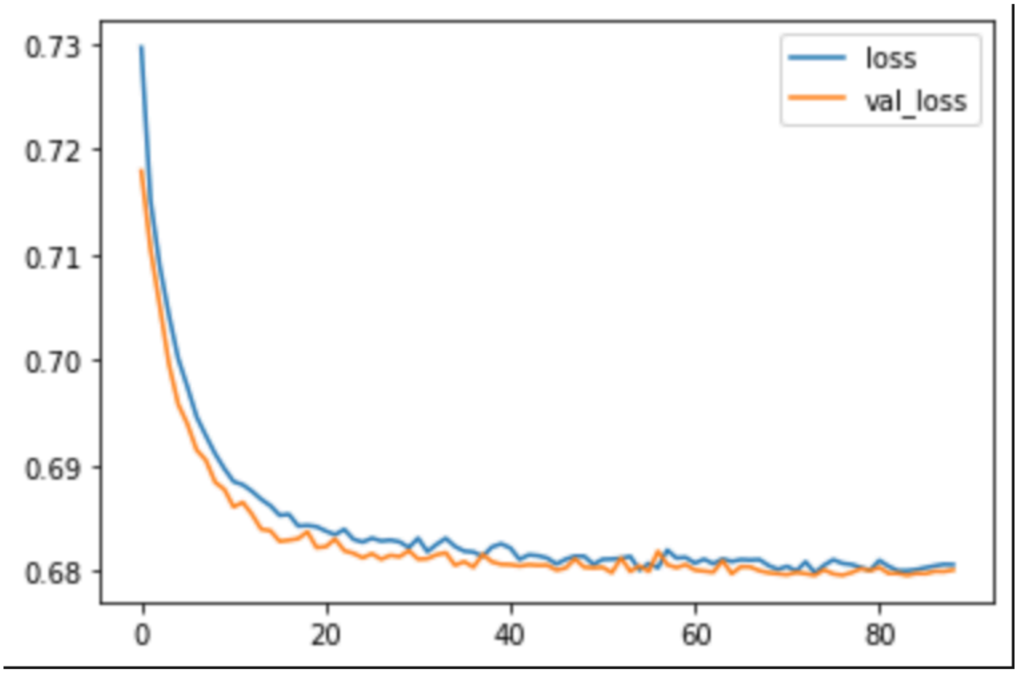


Model 4

Mean Loss:	0.684281
Mean Validation Loss:	0.682942
Batch Size:	2048

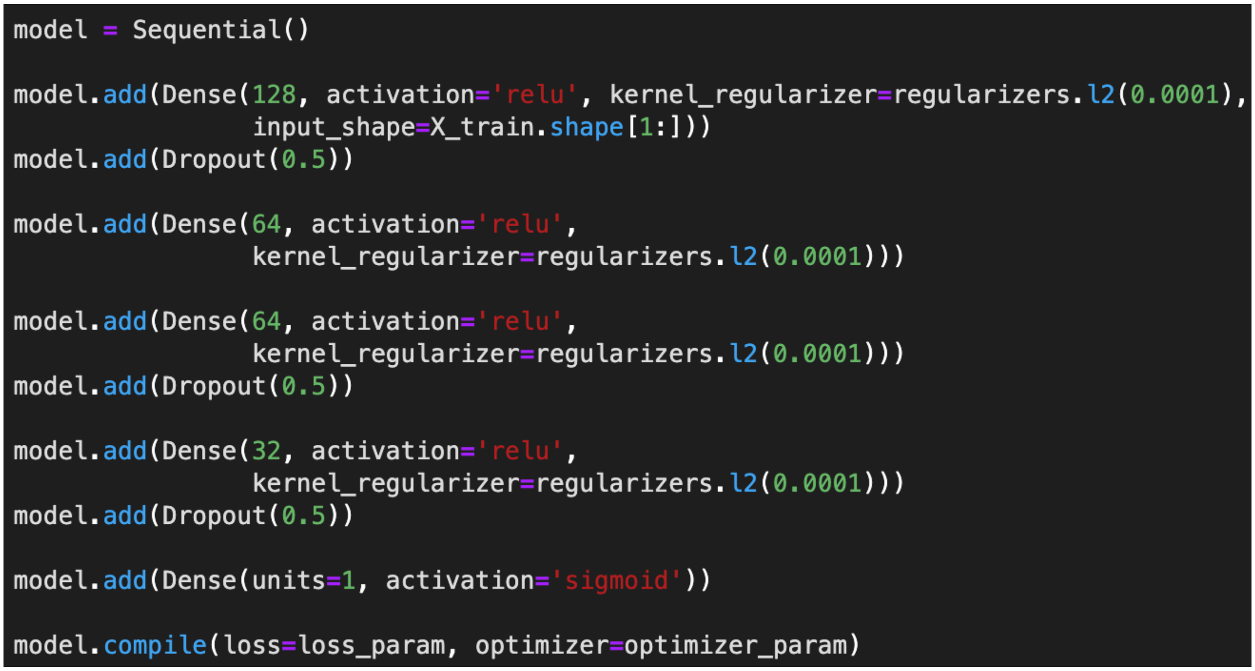


```
model = Sequential()

model.add(Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

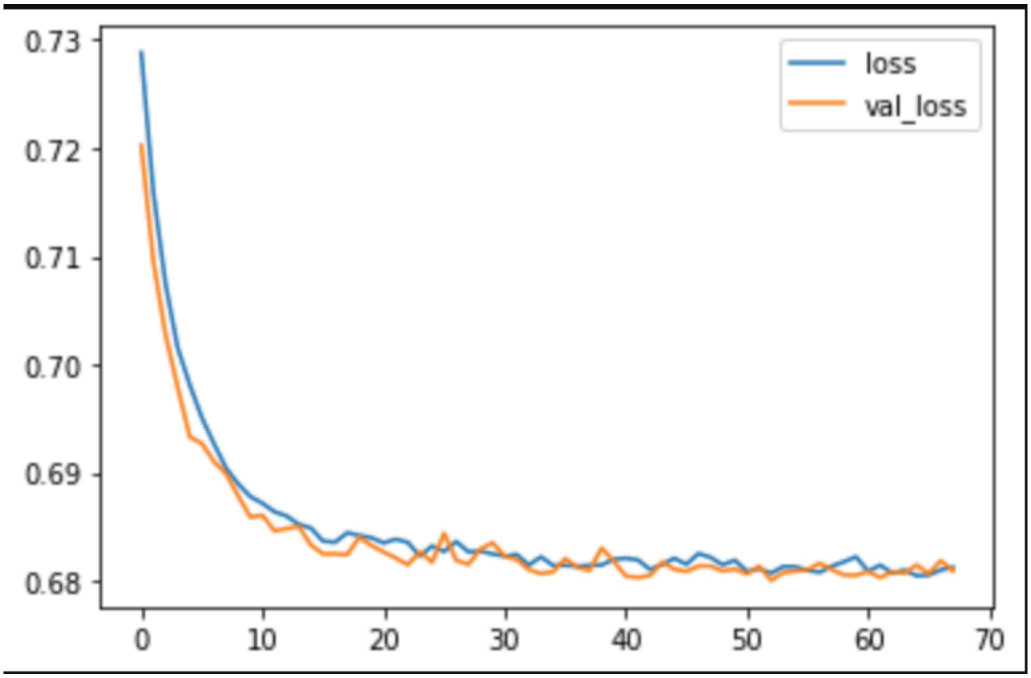


Model 5

Mean Loss:	0.685052
Mean Validation Loss:	0.68408
Batch Size:	1024



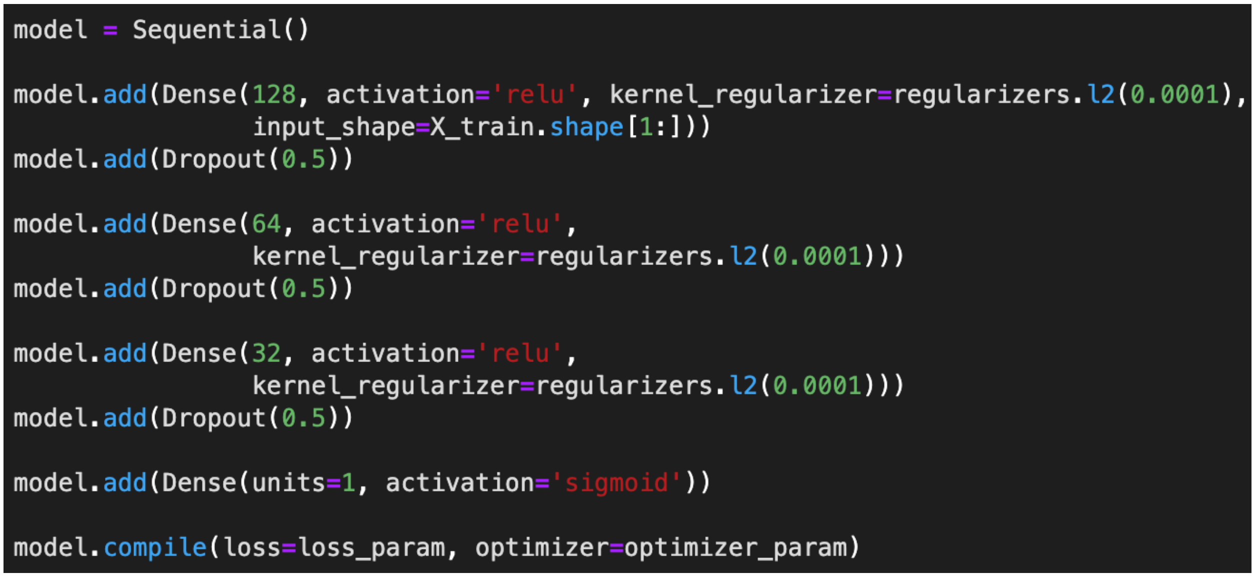
```
model = Sequential()

model.add(Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))

model.add(Dense(64, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```



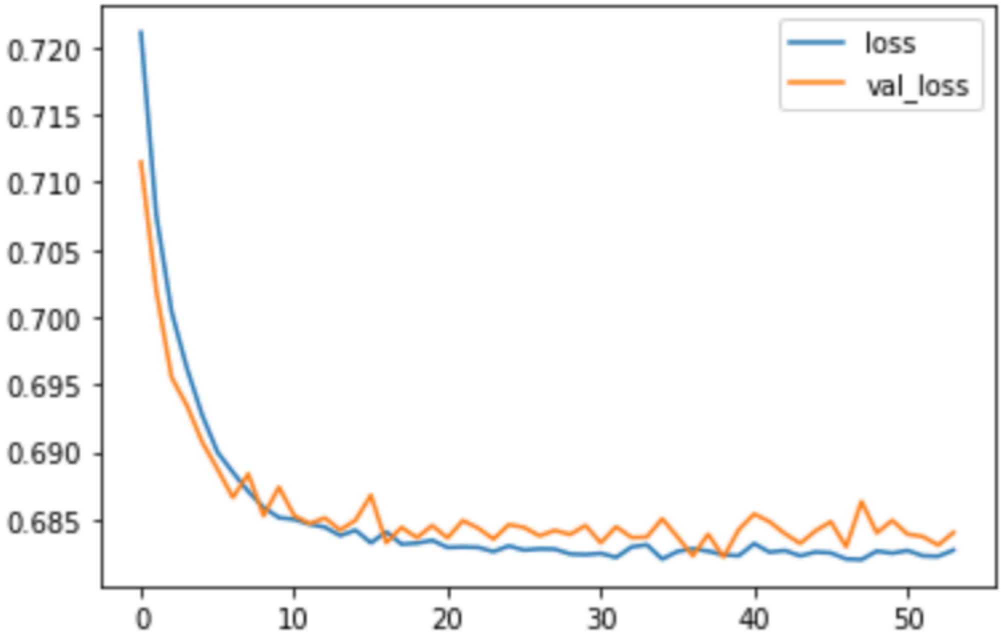
Model 6

Mean Loss: 0.685211
Mean Validation Loss: 0.685804
Batch Size: 1024

Scaling with MinMax normalization from now on

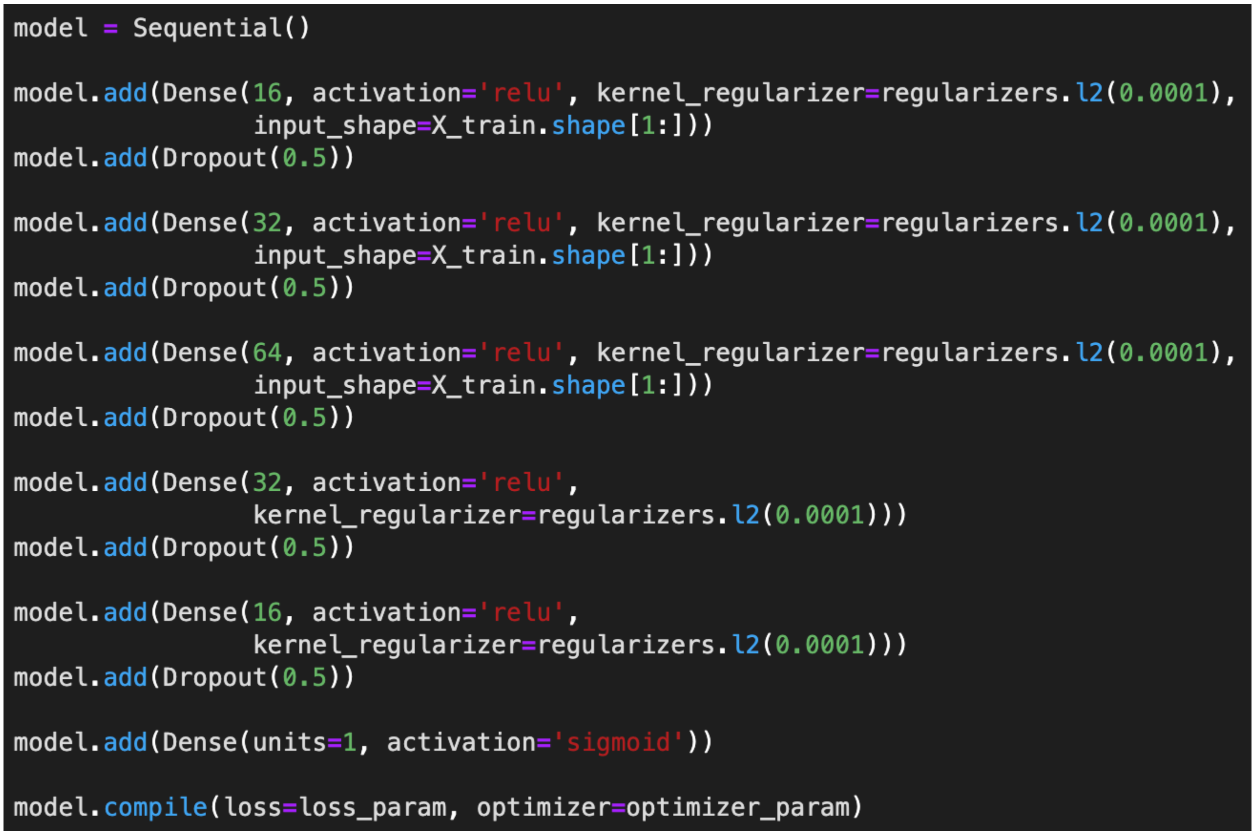


```
model = Sequential()

model.add(Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

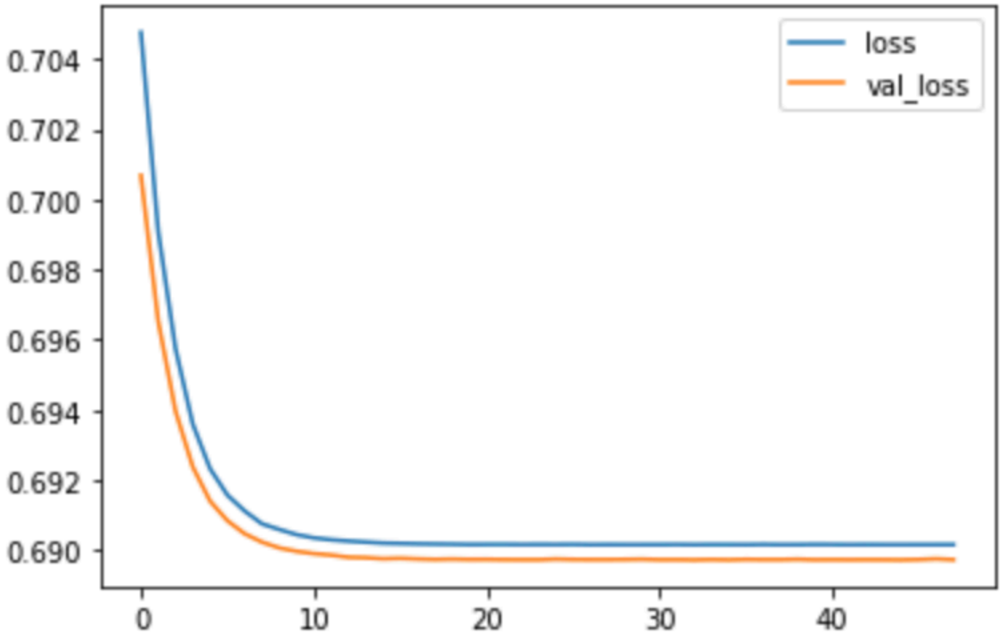


Model 7

Mean Loss:	0.690971
Mean Validation Loss:	0.69035
Batch Size:	2048



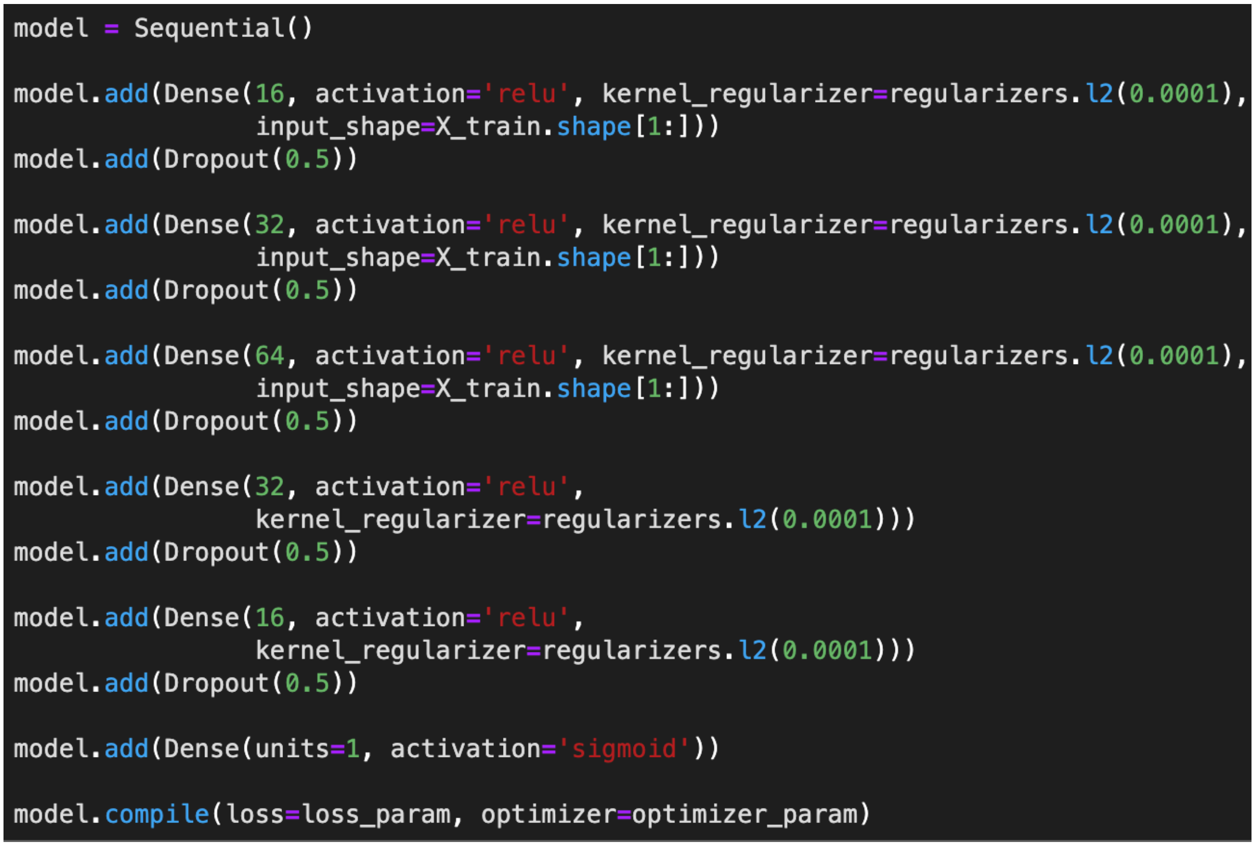
```
model = Sequential()

model.add(Dense(16, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(16, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```



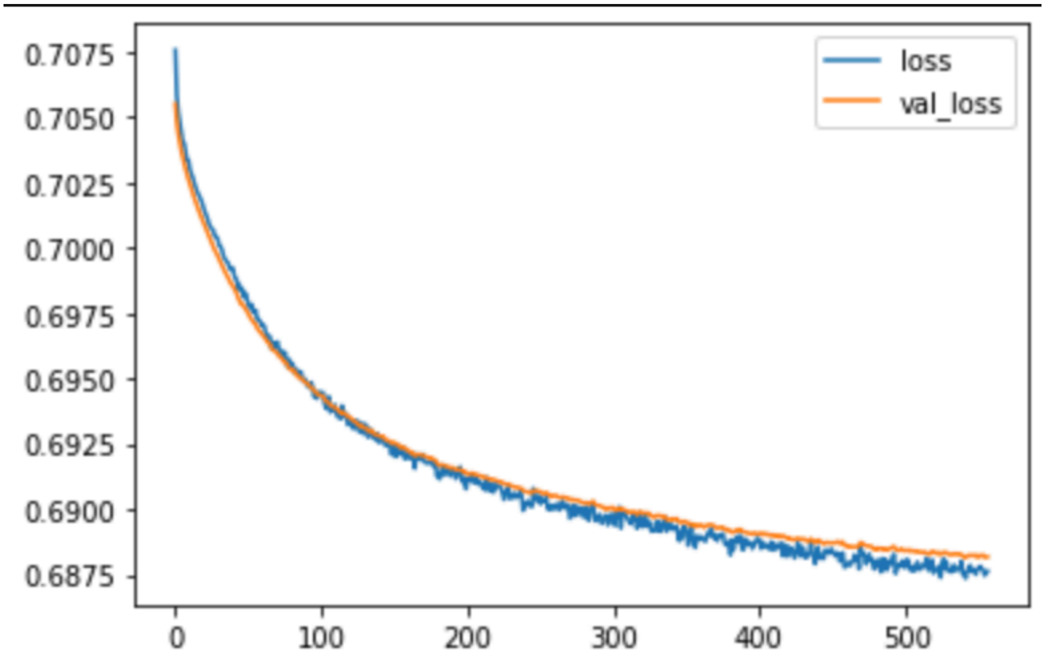
Model 7

Mean Loss:	0.691329
Mean Validation Loss:	0.691562
Batch Size:	32

Adagrad optimization

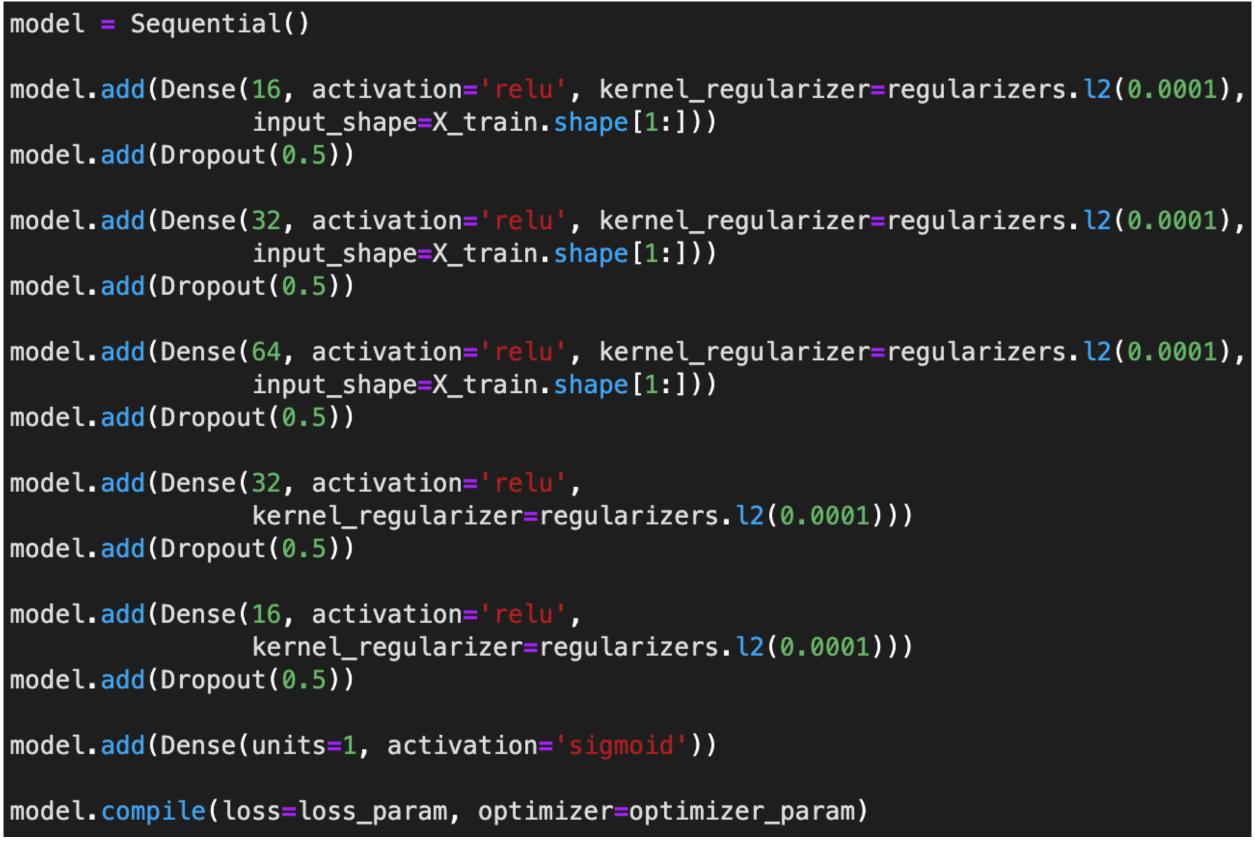


```
model = Sequential()

model.add(Dense(16, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(16, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```

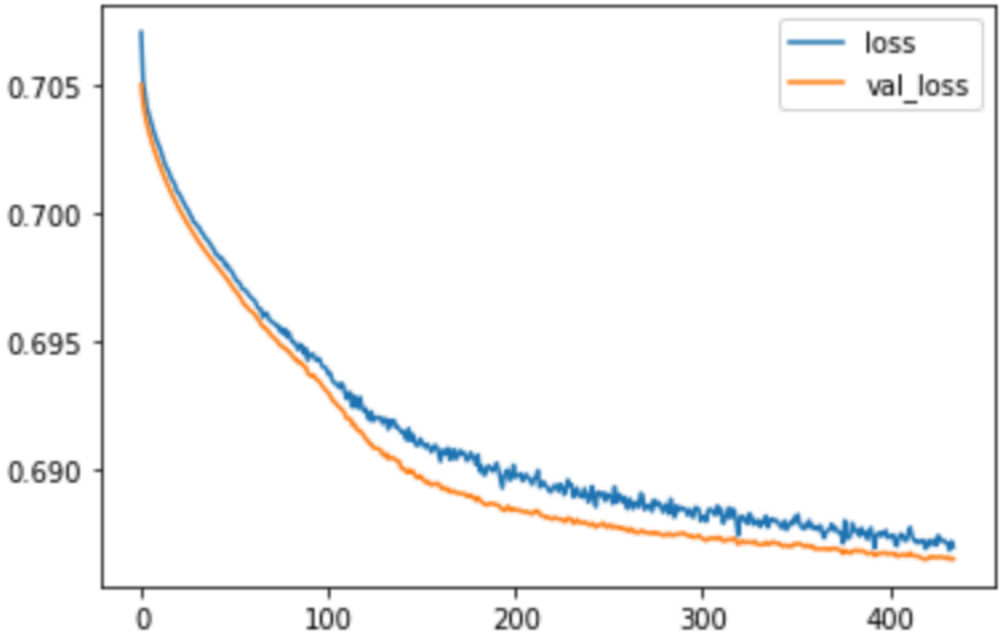


Model 8

Mean Loss:	0.691172
Mean Validation Loss:	0.69024
Batch Size:	16
Adagrad optimization	



```
model = Sequential()

model.add(Dense(16, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.0001),
                input_shape=X_train.shape[1:]))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(16, activation='relu',
                kernel_regularizer=regularizers.l2(0.0001)))
model.add(Dropout(0.5))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param)
```


loss
val_loss