TJ Keras Neural Net

November 21, 2020

1 Tommy John Neural Network

[200]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

[201]:
```
df_train = pd.read_csv('/Users/timmorales/Desktop/STAT 6341/Tommy John Project/
↪Data/SMOTETRAIN.csv')
df_test = pd.read_csv('/Users/timmorales/Desktop/STAT 6341/Tommy John Project/
↪Data/testset.csv')

df_test = df_test.drop(df_test.columns[0], axis=1)
df_train = df_train.drop(df_train.columns[0], axis=1)
```

[202]:
```
from copy import deepcopy
df_train = deepcopy(df_train)
df_test = deepcopy(df_test)
```

[203]:
```
df_train.head()
```

[203]:
```
   player_age    p_game       xba      xslg     xwoba      xobp      xiso  \
0    0.449343  0.271899 -0.236873 -0.433236 -0.346471 -0.221791 -0.481580
1   -0.366322 -1.198887  2.683901  2.403073  2.954865  3.126211  1.774630
2   -0.366322 -1.151442  1.147790  0.499249  1.177223  1.071755 -0.010346
3    2.624452 -1.056553  1.364144  2.558488  2.555803  2.498461  2.859896
4    1.536897 -0.961663  0.282376 -0.180688 -0.001826  0.025505 -0.453021

   exit_velocity_avg  launch_angle_avg  barrel_batted_rate  …  bats_L  \
0          -0.455052         -0.622586           -0.331504  …       0
1           0.644949          0.078895            1.192009  …       0
2           1.103283         -2.676924           -0.625515  …       0
3          -0.363385          1.565368           -0.090950  …       0
4           0.599116          0.245915           -0.625515  …       1

   bats_R  throws_L  throws_R  throws_S  TJ_Yes  made_postseason_X0  \
0       1         0         1         0       0                   0
1       1         0         1         0       0                   1
```

```
2        1         0         1         0       0                  1
3        1         0         1         0       0                  1
4        0         1         0         0       0                  1

   made_postseason_X1  warm_birth_place_X0  warm_birth_place_X1
0                   1                    1                    0
1                   0                    1                    0
2                   0                    1                    0
3                   0                    0                    1
4                   0                    0                    1

[5 rows x 337 columns]
```

[204]:
```
#setting up my targets
y_train = df_train['TJ_Yes'].values.astype(np.int)
y_test = df_test['TJ_Yes'].values.astype(np.int)
```

1.1 Model Work

With the data succesfully and properly loaded, I bring in keras and tensorflow.

[205]:
```
from sklearn import metrics as mt
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Input
from tensorflow.keras.layers import Embedding, Flatten, Concatenate
from tensorflow.keras.models import Model
```

[206]:
```
# combine the features into a single large matrix
X_train = df_train.drop(["TJ_Yes"],axis=1).to_numpy()
X_test = df_test.drop(["TJ_Yes"],axis=1).to_numpy()
```

Input size will always be the same. I start off with the most basic model possible.

[207]:
```
#id number of features
num_features = X_train.shape[1]
#set up input
input_tensor = Input(shape=(num_features,))
```

[208]:
```
from keras.models import Sequential
from keras.layers import Dropout
from keras.layers import Dense
import tensorflow as tf
from sklearn import metrics as mt
```

Start out extremely simple with no relu, dropout or penalization to see how it fits.

[209]:
```
model = Sequential()
model.add(Dense(10, input_dim=num_features, activation='sigmoid'))
model.add(Dense(5, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
```

Im using adam to hopefully find the minima quickly without bouncing around too much and using recall as our metric of choice. Binary Cross is a pretty standard use for this case. 50 epochs and small validation set to start.

[210]:
```
recall = tf.keras.metrics.Recall()

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit([X_train],
                    y_train,
                    epochs=50,
                    batch_size=16,
                    verbose=1,
                    validation_split = .2)

model.summary()
```

```
Epoch 1/50
157/157 [==============================] - 0s 2ms/step - loss: 0.5376 -
accuracy: 0.8401 - val_loss: 1.1647 - val_accuracy: 0.0000e+00
Epoch 2/50
157/157 [==============================] - 0s 1ms/step - loss: 0.3559 -
accuracy: 0.9512 - val_loss: 1.5097 - val_accuracy: 0.0000e+00
Epoch 3/50
157/157 [==============================] - 0s 1ms/step - loss: 0.2817 -
accuracy: 0.9512 - val_loss: 1.8093 - val_accuracy: 0.0000e+00
Epoch 4/50
157/157 [==============================] - 0s 2ms/step - loss: 0.2426 -
accuracy: 0.9512 - val_loss: 2.0613 - val_accuracy: 0.0000e+00
Epoch 5/50
157/157 [==============================] - 0s 2ms/step - loss: 0.2210 -
accuracy: 0.9512 - val_loss: 2.2817 - val_accuracy: 0.0000e+00
Epoch 6/50
157/157 [==============================] - 0s 2ms/step - loss: 0.2087 -
accuracy: 0.9512 - val_loss: 2.4593 - val_accuracy: 0.0000e+00
Epoch 7/50
157/157 [==============================] - 0s 1ms/step - loss: 0.2018 -
accuracy: 0.9512 - val_loss: 2.5880 - val_accuracy: 0.0000e+00
Epoch 8/50
157/157 [==============================] - 0s 998us/step - loss: 0.1925 -
```

```
accuracy: 0.9512 - val_loss: 2.4821 - val_accuracy: 0.0000e+00
Epoch 9/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1744 -
accuracy: 0.9512 - val_loss: 2.3900 - val_accuracy: 0.0000e+00
Epoch 10/50
157/157 [==============================] - 0s 1000us/step - loss: 0.1626 -
accuracy: 0.9512 - val_loss: 2.3797 - val_accuracy: 0.0000e+00
Epoch 11/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1540 -
accuracy: 0.9512 - val_loss: 2.2849 - val_accuracy: 0.0000e+00
Epoch 12/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1474 -
accuracy: 0.9512 - val_loss: 2.1753 - val_accuracy: 0.0000e+00
Epoch 13/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1420 -
accuracy: 0.9512 - val_loss: 2.1722 - val_accuracy: 0.0000e+00
Epoch 14/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1374 -
accuracy: 0.9512 - val_loss: 2.1077 - val_accuracy: 0.0000e+00
Epoch 15/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1332 -
accuracy: 0.9512 - val_loss: 2.2367 - val_accuracy: 0.0000e+00
Epoch 16/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1299 -
accuracy: 0.9516 - val_loss: 1.8244 - val_accuracy: 0.0016
Epoch 17/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1259 -
accuracy: 0.9636 - val_loss: 2.0664 - val_accuracy: 0.0447
Epoch 18/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1224 -
accuracy: 0.9716 - val_loss: 2.0699 - val_accuracy: 0.0942
Epoch 19/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1185 -
accuracy: 0.9732 - val_loss: 2.0512 - val_accuracy: 0.1581
Epoch 20/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1158 -
accuracy: 0.9736 - val_loss: 1.9033 - val_accuracy: 0.2396
Epoch 21/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1133 -
accuracy: 0.9736 - val_loss: 1.7463 - val_accuracy: 0.3227
Epoch 22/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1103 -
accuracy: 0.9740 - val_loss: 1.6785 - val_accuracy: 0.3578
Epoch 23/50
157/157 [==============================] - 0s 2ms/step - loss: 0.1081 -
accuracy: 0.9756 - val_loss: 1.7344 - val_accuracy: 0.3482
Epoch 24/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1050 -
```

```
accuracy: 0.9752 - val_loss: 1.7404 - val_accuracy: 0.3466
Epoch 25/50
157/157 [==============================] - 0s 1ms/step - loss: 0.1028 -
accuracy: 0.9760 - val_loss: 1.7064 - val_accuracy: 0.3690
Epoch 26/50
157/157 [==============================] - 0s 2ms/step - loss: 0.1001 -
accuracy: 0.9768 - val_loss: 1.6879 - val_accuracy: 0.3882
Epoch 27/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0977 -
accuracy: 0.9764 - val_loss: 1.5690 - val_accuracy: 0.4345
Epoch 28/50
157/157 [==============================] - 0s 2ms/step - loss: 0.0954 -
accuracy: 0.9772 - val_loss: 1.5714 - val_accuracy: 0.4361
Epoch 29/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0927 -
accuracy: 0.9776 - val_loss: 1.6742 - val_accuracy: 0.4121
Epoch 30/50
157/157 [==============================] - 0s 2ms/step - loss: 0.0908 -
accuracy: 0.9780 - val_loss: 1.6720 - val_accuracy: 0.4121
Epoch 31/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0883 -
accuracy: 0.9792 - val_loss: 1.7514 - val_accuracy: 0.3898
Epoch 32/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0869 -
accuracy: 0.9796 - val_loss: 1.6834 - val_accuracy: 0.4185
Epoch 33/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0846 -
accuracy: 0.9816 - val_loss: 1.5704 - val_accuracy: 0.4489
Epoch 34/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0825 -
accuracy: 0.9824 - val_loss: 1.6768 - val_accuracy: 0.4201
Epoch 35/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0806 -
accuracy: 0.9824 - val_loss: 1.6613 - val_accuracy: 0.4265
Epoch 36/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0784 -
accuracy: 0.9832 - val_loss: 1.6674 - val_accuracy: 0.4409
Epoch 37/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0770 -
accuracy: 0.9832 - val_loss: 1.7335 - val_accuracy: 0.4105
Epoch 38/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0746 -
accuracy: 0.9844 - val_loss: 1.6795 - val_accuracy: 0.4377
Epoch 39/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0728 -
accuracy: 0.9836 - val_loss: 1.7563 - val_accuracy: 0.4105
Epoch 40/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0705 -
```

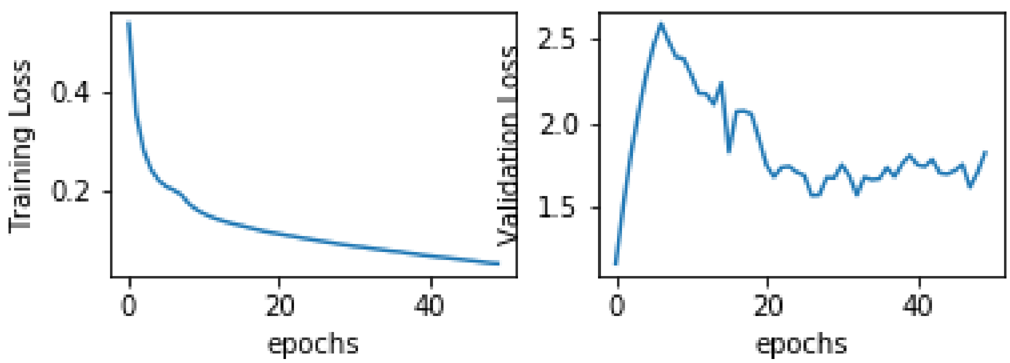
```
accuracy: 0.9852 - val_loss: 1.8054 - val_accuracy: 0.4026
Epoch 41/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0688 -
accuracy: 0.9852 - val_loss: 1.7502 - val_accuracy: 0.4201
Epoch 42/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0668 -
accuracy: 0.9856 - val_loss: 1.7394 - val_accuracy: 0.4345
Epoch 43/50
157/157 [==============================] - 0s 2ms/step - loss: 0.0655 -
accuracy: 0.9864 - val_loss: 1.7809 - val_accuracy: 0.4217
Epoch 44/50
157/157 [==============================] - 0s 2ms/step - loss: 0.0637 -
accuracy: 0.9868 - val_loss: 1.7009 - val_accuracy: 0.4505
Epoch 45/50
157/157 [==============================] - 0s 2ms/step - loss: 0.0619 -
accuracy: 0.9880 - val_loss: 1.6950 - val_accuracy: 0.4569
Epoch 46/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0601 -
accuracy: 0.9868 - val_loss: 1.7160 - val_accuracy: 0.4585
Epoch 47/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0584 -
accuracy: 0.9876 - val_loss: 1.7516 - val_accuracy: 0.4441
Epoch 48/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0563 -
accuracy: 0.9880 - val_loss: 1.6174 - val_accuracy: 0.4856
Epoch 49/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0545 -
accuracy: 0.9888 - val_loss: 1.7046 - val_accuracy: 0.4585
Epoch 50/50
157/157 [==============================] - 0s 1ms/step - loss: 0.0536 -
accuracy: 0.9884 - val_loss: 1.8209 - val_accuracy: 0.4249
Model: "sequential_36"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_105 (Dense)            (None, 10)                3370
_________________________________________________________________
dense_106 (Dense)            (None, 5)                 55
_________________________________________________________________
dense_107 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[211]:
```
# now lets see how well the model performed
yhat_proba = model.predict(X_test)
yhat = np.where(yhat_proba > 0.5, 1, 0)
print(mt.confusion_matrix(y_test,yhat))
print(mt.classification_report(y_test,yhat))
```

```
[[1042   22]
 [  28    1]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      1064
           1       0.04      0.03      0.04        29

    accuracy                           0.95      1093
   macro avg       0.51      0.51      0.51      1093
weighted avg       0.95      0.95      0.95      1093
```

[212]:
```
plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[212]: Text(0.5, 0, 'epochs')



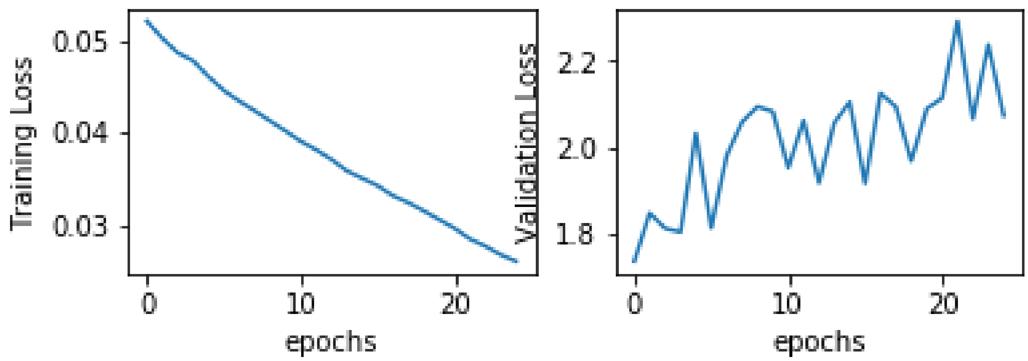
Model doesnt seem to converge fully, will add more epochs, but looks like we may be starting to overfit.

[213]:
```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit([X_train],
                    y_train,
                    epochs=25,
                    batch_size=16,
                    verbose=0,
                    validation_split = .2)

model.summary()
```

```
Model: "sequential_36"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_105 (Dense)            (None, 10)                3370
_________________________________________________________________
dense_106 (Dense)            (None, 5)                 55
_________________________________________________________________
dense_107 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[214]:
```
plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[214]:
```
Text(0.5, 0, 'epochs')
```



Looks like adding the 25 actually resulted in some overfitting when comparing validation and training loss. We therefore will fit for 70 total epochs, but include some drop out. I also include some regularization on the hidden layer to aid in the overfitting.

Since I see no issue with the model training I dont change activiation function.

[215]:
```
model = Sequential()
model.add(Dense(10, input_dim=num_features, activation='sigmoid'))
model.add(Dropout(0.25))
model.add(Dense(5, activation='sigmoid',kernel_regularizer=tf.keras.
 ↪regularizers.l2(1e-4)))
model.add(Dropout(0.1))
model.add(Dense(1, activation='sigmoid'))
```

[216]:
```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit([X_train],
                    y_train,
                    epochs=70,
                    batch_size=16,
                    verbose=1,
                    validation_split = .2)

model.summary()
```

```
Epoch 1/70
157/157 [==============================] - 0s 2ms/step - loss: 0.3087 -
accuracy: 0.9464 - val_loss: 2.1051 - val_accuracy: 0.0000e+00
Epoch 2/70
157/157 [==============================] - 0s 1ms/step - loss: 0.2250 -
accuracy: 0.9508 - val_loss: 2.5909 - val_accuracy: 0.0000e+00
```

```
Epoch 3/70
157/157 [==============================] - 0s 1ms/step - loss: 0.2008 -
accuracy: 0.9512 - val_loss: 2.6669 - val_accuracy: 0.0000e+00
Epoch 4/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1891 -
accuracy: 0.9512 - val_loss: 2.6470 - val_accuracy: 0.0000e+00
Epoch 5/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1734 -
accuracy: 0.9512 - val_loss: 2.5104 - val_accuracy: 0.0000e+00
Epoch 6/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1676 -
accuracy: 0.9512 - val_loss: 2.5101 - val_accuracy: 0.0000e+00
Epoch 7/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1613 -
accuracy: 0.9512 - val_loss: 2.3670 - val_accuracy: 0.0000e+00
Epoch 8/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1560 -
accuracy: 0.9512 - val_loss: 2.3374 - val_accuracy: 0.0000e+00
Epoch 9/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1528 -
accuracy: 0.9512 - val_loss: 2.2518 - val_accuracy: 0.0000e+00
Epoch 10/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1426 -
accuracy: 0.9516 - val_loss: 2.0788 - val_accuracy: 0.0000e+00
Epoch 11/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1419 -
accuracy: 0.9564 - val_loss: 2.2219 - val_accuracy: 0.0000e+00
Epoch 12/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1403 -
accuracy: 0.9584 - val_loss: 2.1352 - val_accuracy: 0.0000e+00
Epoch 13/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1373 -
accuracy: 0.9604 - val_loss: 2.0941 - val_accuracy: 0.0000e+00
Epoch 14/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1378 -
accuracy: 0.9592 - val_loss: 1.9955 - val_accuracy: 0.0048
Epoch 15/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1307 -
accuracy: 0.9612 - val_loss: 1.9488 - val_accuracy: 0.0224
Epoch 16/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1348 -
accuracy: 0.9632 - val_loss: 1.8590 - val_accuracy: 0.0719
Epoch 17/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1255 -
accuracy: 0.9668 - val_loss: 1.9578 - val_accuracy: 0.1006
Epoch 18/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1317 -
accuracy: 0.9656 - val_loss: 1.8788 - val_accuracy: 0.1454
```

```
Epoch 19/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1305 -
accuracy: 0.9668 - val_loss: 1.8386 - val_accuracy: 0.1725
Epoch 20/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1231 -
accuracy: 0.9688 - val_loss: 1.7317 - val_accuracy: 0.2524
Epoch 21/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1227 -
accuracy: 0.9696 - val_loss: 1.8374 - val_accuracy: 0.2300
Epoch 22/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1175 -
accuracy: 0.9688 - val_loss: 1.8807 - val_accuracy: 0.2300
Epoch 23/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1180 -
accuracy: 0.9708 - val_loss: 1.8948 - val_accuracy: 0.2428
Epoch 24/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1176 -
accuracy: 0.9680 - val_loss: 1.9699 - val_accuracy: 0.2380
Epoch 25/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1095 -
accuracy: 0.9712 - val_loss: 1.8254 - val_accuracy: 0.3003
Epoch 26/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1135 -
accuracy: 0.9680 - val_loss: 1.7853 - val_accuracy: 0.3163
Epoch 27/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1094 -
accuracy: 0.9716 - val_loss: 1.7864 - val_accuracy: 0.3307
Epoch 28/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1105 -
accuracy: 0.9744 - val_loss: 1.8322 - val_accuracy: 0.3259
Epoch 29/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1144 -
accuracy: 0.9684 - val_loss: 1.7290 - val_accuracy: 0.3674
Epoch 30/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1126 -
accuracy: 0.9744 - val_loss: 1.7534 - val_accuracy: 0.3610
Epoch 31/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1113 -
accuracy: 0.9724 - val_loss: 1.8818 - val_accuracy: 0.3243
Epoch 32/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1069 -
accuracy: 0.9732 - val_loss: 1.7270 - val_accuracy: 0.3770
Epoch 33/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1043 -
accuracy: 0.9732 - val_loss: 1.7665 - val_accuracy: 0.3674
Epoch 34/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1075 -
accuracy: 0.9744 - val_loss: 1.8006 - val_accuracy: 0.3626
```

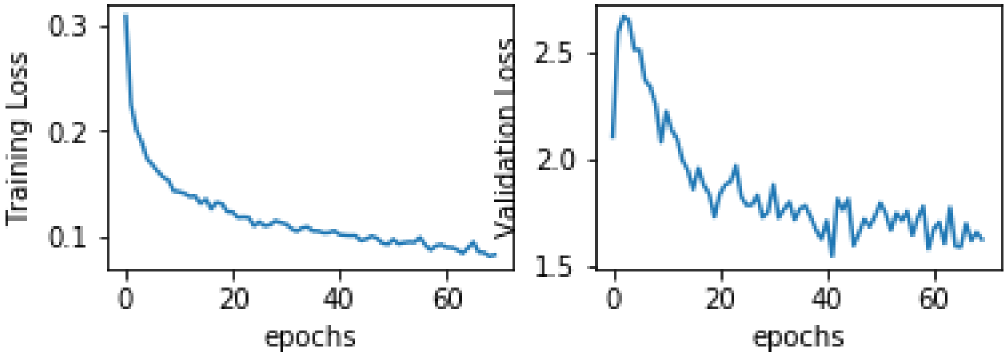
```
Epoch 35/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1087 -
accuracy: 0.9724 - val_loss: 1.7185 - val_accuracy: 0.3962
Epoch 36/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1046 -
accuracy: 0.9752 - val_loss: 1.7690 - val_accuracy: 0.3914
Epoch 37/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1046 -
accuracy: 0.9744 - val_loss: 1.7832 - val_accuracy: 0.3866
Epoch 38/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1028 -
accuracy: 0.9772 - val_loss: 1.7297 - val_accuracy: 0.4089
Epoch 39/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1031 -
accuracy: 0.9760 - val_loss: 1.6698 - val_accuracy: 0.4361
Epoch 40/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1048 -
accuracy: 0.9760 - val_loss: 1.6268 - val_accuracy: 0.4473
Epoch 41/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1013 -
accuracy: 0.9744 - val_loss: 1.7136 - val_accuracy: 0.4217
Epoch 42/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1006 -
accuracy: 0.9788 - val_loss: 1.5470 - val_accuracy: 0.4808
Epoch 43/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1002 -
accuracy: 0.9752 - val_loss: 1.8153 - val_accuracy: 0.3978
Epoch 44/70
157/157 [==============================] - 0s 1ms/step - loss: 0.1004 -
accuracy: 0.9756 - val_loss: 1.7643 - val_accuracy: 0.4121
Epoch 45/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0954 -
accuracy: 0.9780 - val_loss: 1.8138 - val_accuracy: 0.4058
Epoch 46/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0969 -
accuracy: 0.9760 - val_loss: 1.5965 - val_accuracy: 0.4744
Epoch 47/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0998 -
accuracy: 0.9760 - val_loss: 1.6546 - val_accuracy: 0.4633
Epoch 48/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0977 -
accuracy: 0.9764 - val_loss: 1.7214 - val_accuracy: 0.4329
Epoch 49/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0930 -
accuracy: 0.9768 - val_loss: 1.6842 - val_accuracy: 0.4489
Epoch 50/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0919 -
accuracy: 0.9736 - val_loss: 1.7338 - val_accuracy: 0.4329
```

```
Epoch 51/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0968 -
accuracy: 0.9756 - val_loss: 1.7959 - val_accuracy: 0.4073
Epoch 52/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0924 -
accuracy: 0.9792 - val_loss: 1.7483 - val_accuracy: 0.4217
Epoch 53/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0936 -
accuracy: 0.9788 - val_loss: 1.6698 - val_accuracy: 0.4505
Epoch 54/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0942 -
accuracy: 0.9776 - val_loss: 1.7485 - val_accuracy: 0.4297
Epoch 55/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0939 -
accuracy: 0.9764 - val_loss: 1.7138 - val_accuracy: 0.4409
Epoch 56/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0985 -
accuracy: 0.9744 - val_loss: 1.7578 - val_accuracy: 0.4217
Epoch 57/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0914 -
accuracy: 0.9776 - val_loss: 1.6458 - val_accuracy: 0.4728
Epoch 58/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0866 -
accuracy: 0.9792 - val_loss: 1.7283 - val_accuracy: 0.4441
Epoch 59/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0905 -
accuracy: 0.9768 - val_loss: 1.7801 - val_accuracy: 0.4249
Epoch 60/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0918 -
accuracy: 0.9748 - val_loss: 1.5805 - val_accuracy: 0.5032
Epoch 61/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0890 -
accuracy: 0.9780 - val_loss: 1.6761 - val_accuracy: 0.4649
Epoch 62/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0889 -
accuracy: 0.9788 - val_loss: 1.7070 - val_accuracy: 0.4617
Epoch 63/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0868 -
accuracy: 0.9792 - val_loss: 1.6066 - val_accuracy: 0.5032
Epoch 64/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0833 -
accuracy: 0.9788 - val_loss: 1.7748 - val_accuracy: 0.4457
Epoch 65/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0885 -
accuracy: 0.9776 - val_loss: 1.5944 - val_accuracy: 0.4968
Epoch 66/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0942 -
accuracy: 0.9760 - val_loss: 1.5903 - val_accuracy: 0.5080
```

```
Epoch 67/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0845 -
accuracy: 0.9812 - val_loss: 1.7004 - val_accuracy: 0.4792
Epoch 68/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0840 -
accuracy: 0.9804 - val_loss: 1.6196 - val_accuracy: 0.5064
Epoch 69/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0805 -
accuracy: 0.9820 - val_loss: 1.6595 - val_accuracy: 0.4936
Epoch 70/70
157/157 [==============================] - 0s 1ms/step - loss: 0.0819 -
accuracy: 0.9792 - val_loss: 1.6255 - val_accuracy: 0.5112
Model: "sequential_37"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_108 (Dense)            (None, 10)                3370
_________________________________________________________________
dropout_63 (Dropout)         (None, 10)                0
_________________________________________________________________
dense_109 (Dense)            (None, 5)                 55
_________________________________________________________________
dropout_64 (Dropout)         (None, 5)                 0
_________________________________________________________________
dense_110 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[217]:
```
plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[217]:
```
Text(0.5, 0, 'epochs')
```



[218]:
```
# now lets see how well the model performed
yhat_proba = model.predict(X_test)
yhat = np.where(yhat_proba > 0.5, 1, 0)
print(mt.confusion_matrix(y_test,yhat))
print(mt.classification_report(y_test,yhat))
```

```
[[1041   23]
 [  28    1]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      1064
           1       0.04      0.03      0.04        29

    accuracy                           0.95      1093
   macro avg       0.51      0.51      0.51      1093
weighted avg       0.95      0.95      0.95      1093
```

No improvement in the results. Since it did not seem to be training very well after a few epochs, I will use relu instead of sigmoid to address gradient issue potentially.

[219]:
```
model = Sequential()
model.add(Dense(10, input_dim=num_features, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(5, activation='relu',kernel_regularizer=tf.keras.regularizers.
 ↪l2(1e-4)))
model.add(Dropout(0.1))
model.add(Dense(1, activation='relu'))
```

[220]:
```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```python
history = model.fit([X_train],
                    y_train,
                    epochs=70,
                    batch_size=16,
                    verbose=1,
                    validation_split = .2)

model.summary()
```

```
Epoch 1/70
157/157 [==============================] - 0s 2ms/step - loss: 0.7605 -
accuracy: 0.9504 - val_loss: 15.4255 - val_accuracy: 0.0000e+00
Epoch 2/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7527 -
accuracy: 0.9512 - val_loss: 15.4254 - val_accuracy: 0.0000e+00
Epoch 3/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7531 -
accuracy: 0.9508 - val_loss: 15.4253 - val_accuracy: 0.0000e+00
Epoch 4/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7525 -
accuracy: 0.9512 - val_loss: 15.4253 - val_accuracy: 0.0000e+00
Epoch 5/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7526 -
accuracy: 0.9512 - val_loss: 15.4252 - val_accuracy: 0.0000e+00
Epoch 6/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7524 -
accuracy: 0.9512 - val_loss: 15.4252 - val_accuracy: 0.0000e+00
Epoch 7/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7524 -
accuracy: 0.9512 - val_loss: 15.4252 - val_accuracy: 0.0000e+00
Epoch 8/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7525 -
accuracy: 0.9512 - val_loss: 15.4252 - val_accuracy: 0.0000e+00
Epoch 9/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4252 - val_accuracy: 0.0000e+00
Epoch 10/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 11/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 12/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 13/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
```

```
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 14/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 15/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 16/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 17/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 18/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 19/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4251 - val_accuracy: 0.0000e+00
Epoch 20/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 21/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 22/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 23/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 24/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 25/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 26/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 27/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 28/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 29/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
```

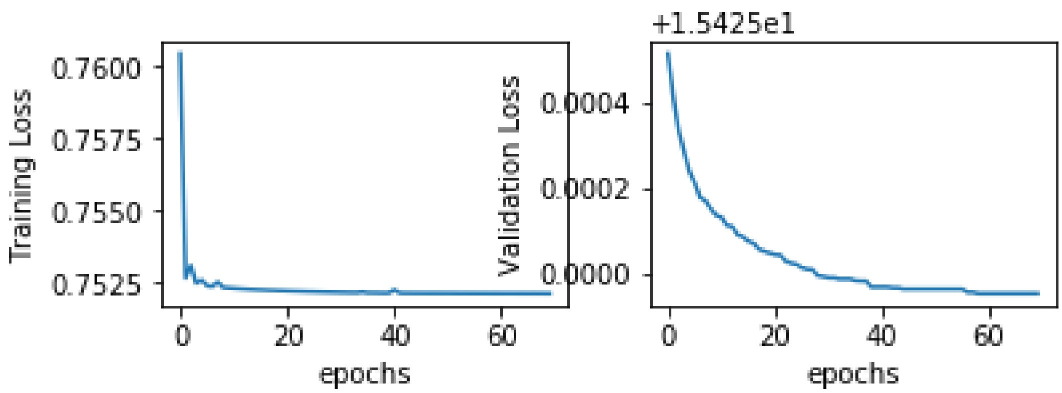
```
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 30/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 31/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 32/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 33/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 34/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 35/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 36/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 37/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 38/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 39/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 40/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 41/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7523 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 42/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 43/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 44/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 45/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
```

```
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 46/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 47/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 48/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 49/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7522 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 50/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 51/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 52/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 53/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 54/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 55/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 56/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 57/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 58/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 59/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 60/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 61/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
```

```
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 62/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 63/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 64/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 65/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 66/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 67/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 68/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 69/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Epoch 70/70
157/157 [==============================] - 0s 1ms/step - loss: 0.7521 -
accuracy: 0.9512 - val_loss: 15.4250 - val_accuracy: 0.0000e+00
Model: "sequential_38"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_111 (Dense)            (None, 10)                3370
_________________________________________________________________
dropout_65 (Dropout)         (None, 10)                0
_________________________________________________________________
dense_112 (Dense)            (None, 5)                 55
_________________________________________________________________
dropout_66 (Dropout)         (None, 5)                 0
_________________________________________________________________
dense_113 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[221]:
```
%matplotlib inline

# plt.figure(figsize=(10,4))
# plt.subplot(2,2,1)
# plt.plot(history.history['recall'])

# plt.ylabel('Accuracy %')
# plt.title('Training')
# plt.subplot(2,2,2)
# plt.plot(history.history['val_accuracy'])
# plt.title('Validation')

plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[221]:
```
Text(0.5, 0, 'epochs')
```



Looks to be overfitting again, but now at around 40 epochs. The relu has helped to train much faster.

[222]:
```
model = Sequential()
model.add(Dense(10, input_dim=num_features, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(5, activation='relu',kernel_regularizer=tf.keras.regularizers.
 ↪l2(1e-4)))
model.add(Dropout(0.1))
```

```python
model.add(Dense(1, activation='relu'))

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit([X_train],
                    y_train,
                    epochs=40,
                    batch_size=16,
                    verbose=1,
                    validation_split = .2)

model.summary()
```

```
Epoch 1/40
157/157 [==============================] - 0s 2ms/step - loss: 0.6971 -
accuracy: 0.9452 - val_loss: 14.1004 - val_accuracy: 0.0000e+00
Epoch 2/40
157/157 [==============================] - 0s 1ms/step - loss: 0.6089 -
accuracy: 0.9552 - val_loss: 13.1706 - val_accuracy: 0.0016
Epoch 3/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5847 -
accuracy: 0.9588 - val_loss: 11.8667 - val_accuracy: 0.0304
Epoch 4/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5404 -
accuracy: 0.9616 - val_loss: 13.0796 - val_accuracy: 0.0048
Epoch 5/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5528 -
accuracy: 0.9636 - val_loss: 13.2361 - val_accuracy: 0.0064
Epoch 6/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5506 -
accuracy: 0.9628 - val_loss: 13.2077 - val_accuracy: 0.0096
Epoch 7/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5108 -
accuracy: 0.9656 - val_loss: 12.2625 - val_accuracy: 0.0256
Epoch 8/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5652 -
accuracy: 0.9620 - val_loss: 13.6683 - val_accuracy: 0.0080
Epoch 9/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5318 -
accuracy: 0.9656 - val_loss: 12.6389 - val_accuracy: 0.0240
Epoch 10/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5139 -
accuracy: 0.9660 - val_loss: 13.2466 - val_accuracy: 0.0112
Epoch 11/40
```

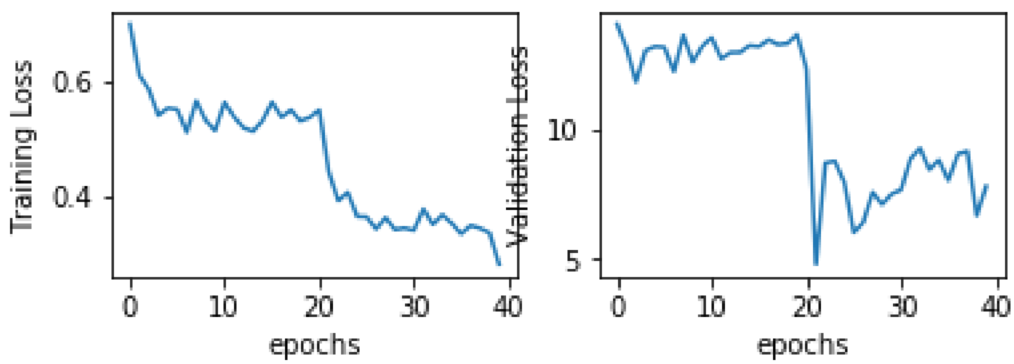
```
157/157 [==============================] - 0s 1ms/step - loss: 0.5627 -
accuracy: 0.9632 - val_loss: 13.5850 - val_accuracy: 0.0128
Epoch 12/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5377 -
accuracy: 0.9648 - val_loss: 12.7645 - val_accuracy: 0.0272
Epoch 13/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5191 -
accuracy: 0.9660 - val_loss: 13.0016 - val_accuracy: 0.0256
Epoch 14/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5126 -
accuracy: 0.9668 - val_loss: 13.0074 - val_accuracy: 0.0256
Epoch 15/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5313 -
accuracy: 0.9656 - val_loss: 13.2750 - val_accuracy: 0.0160
Epoch 16/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5633 -
accuracy: 0.9632 - val_loss: 13.2336 - val_accuracy: 0.0128
Epoch 17/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5372 -
accuracy: 0.9648 - val_loss: 13.4930 - val_accuracy: 0.0128
Epoch 18/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5494 -
accuracy: 0.9644 - val_loss: 13.3100 - val_accuracy: 0.0192
Epoch 19/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5309 -
accuracy: 0.9656 - val_loss: 13.3550 - val_accuracy: 0.0256
Epoch 20/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5369 -
accuracy: 0.9652 - val_loss: 13.7001 - val_accuracy: 0.0160
Epoch 21/40
157/157 [==============================] - 0s 1ms/step - loss: 0.5497 -
accuracy: 0.9632 - val_loss: 12.3731 - val_accuracy: 0.0511
Epoch 22/40
157/157 [==============================] - 0s 1ms/step - loss: 0.4430 -
accuracy: 0.9660 - val_loss: 4.7972 - val_accuracy: 0.1438
Epoch 23/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3935 -
accuracy: 0.9648 - val_loss: 8.7303 - val_accuracy: 0.1166
Epoch 24/40
157/157 [==============================] - 0s 1ms/step - loss: 0.4077 -
accuracy: 0.9636 - val_loss: 8.7951 - val_accuracy: 0.0687
Epoch 25/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3664 -
accuracy: 0.9664 - val_loss: 7.9730 - val_accuracy: 0.1118
Epoch 26/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3658 -
accuracy: 0.9640 - val_loss: 6.0295 - val_accuracy: 0.1725
Epoch 27/40
```

```
157/157 [==============================] - 0s 1ms/step - loss: 0.3449 -
accuracy: 0.9656 - val_loss: 6.4255 - val_accuracy: 0.1422
Epoch 28/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3644 -
accuracy: 0.9656 - val_loss: 7.5874 - val_accuracy: 0.1278
Epoch 29/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3439 -
accuracy: 0.9636 - val_loss: 7.1197 - val_accuracy: 0.1342
Epoch 30/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3464 -
accuracy: 0.9680 - val_loss: 7.5189 - val_accuracy: 0.1486
Epoch 31/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3427 -
accuracy: 0.9676 - val_loss: 7.6842 - val_accuracy: 0.1278
Epoch 32/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3793 -
accuracy: 0.9680 - val_loss: 8.8887 - val_accuracy: 0.1246
Epoch 33/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3527 -
accuracy: 0.9688 - val_loss: 9.2948 - val_accuracy: 0.1150
Epoch 34/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3691 -
accuracy: 0.9668 - val_loss: 8.4674 - val_accuracy: 0.1502
Epoch 35/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3548 -
accuracy: 0.9664 - val_loss: 8.8380 - val_accuracy: 0.1374
Epoch 36/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3366 -
accuracy: 0.9708 - val_loss: 8.0470 - val_accuracy: 0.1310
Epoch 37/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3511 -
accuracy: 0.9672 - val_loss: 9.0637 - val_accuracy: 0.1150
Epoch 38/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3461 -
accuracy: 0.9692 - val_loss: 9.1729 - val_accuracy: 0.1246
Epoch 39/40
157/157 [==============================] - 0s 1ms/step - loss: 0.3382 -
accuracy: 0.9680 - val_loss: 6.6909 - val_accuracy: 0.1326
Epoch 40/40
157/157 [==============================] - 0s 1ms/step - loss: 0.2840 -
accuracy: 0.9656 - val_loss: 7.8233 - val_accuracy: 0.1182
Model: "sequential_39"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_114 (Dense)            (None, 10)                3370
_________________________________________________________________
dropout_67 (Dropout)         (None, 10)                0
```

```
_________________________________________________________________
dense_115 (Dense)            (None, 5)                 55
_________________________________________________________________
dropout_68 (Dropout)         (None, 5)                 0
_________________________________________________________________
dense_116 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[223]:
```
plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[223]:
```
Text(0.5, 0, 'epochs')
```



[224]:
```
# now lets see how well the model performed
yhat_proba = model.predict(X_test)
yhat = np.where(yhat_proba > 0.5, 1, 0)
print(mt.confusion_matrix(y_test,yhat))
print(mt.classification_report(y_test,yhat))
```

```
[[1061    3]
 [  28    1]]
              precision    recall  f1-score   support
```

```
           0       0.97      1.00      0.99      1064
           1       0.25      0.03      0.06        29

    accuracy                           0.97      1093
   macro avg       0.61      0.52      0.52      1093
weighted avg       0.96      0.97      0.96      1093
```

It does look like the model really struggles to find the global minima and that is to be expected with small data. Last thing I will try is to increase the batch size. I keep in scale of 2.

[225]:
```
model = Sequential()
model.add(Dense(10, input_dim=num_features, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(5, activation='relu',kernel_regularizer=tf.keras.regularizers.
 ↪l2(1e-4)))
model.add(Dropout(0.1))
model.add(Dense(1, activation='relu'))

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit([X_train],
                    y_train,
                    epochs=40,
                    batch_size=24,
                    verbose=1,
                    validation_split = .2)

model.summary()
```

```
Epoch 1/40
105/105 [==============================] - 0s 2ms/step - loss: 0.7482 -
accuracy: 0.9496 - val_loss: 15.4039 - val_accuracy: 0.0000e+00
Epoch 2/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7608 -
accuracy: 0.9504 - val_loss: 15.4059 - val_accuracy: 0.0000e+00
Epoch 3/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7443 -
accuracy: 0.9508 - val_loss: 15.4258 - val_accuracy: 0.0000e+00
Epoch 4/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7442 -
accuracy: 0.9512 - val_loss: 15.4258 - val_accuracy: 0.0000e+00
Epoch 5/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7440 -
accuracy: 0.9512 - val_loss: 15.4258 - val_accuracy: 0.0000e+00
```

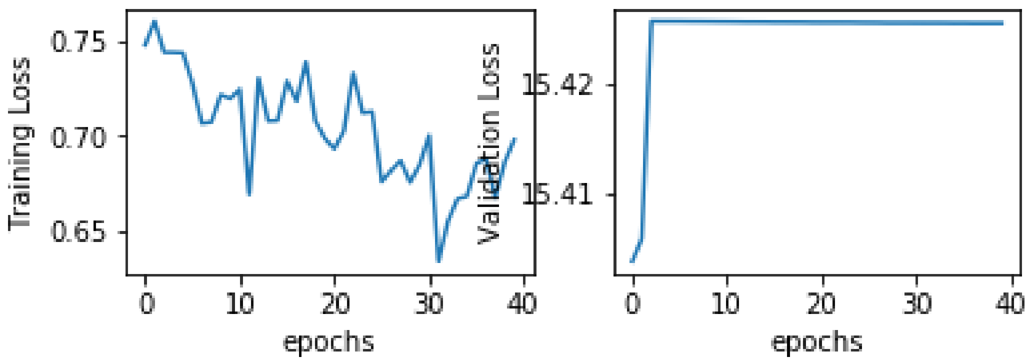
```
Epoch 6/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7277 -
accuracy: 0.9508 - val_loss: 15.4258 - val_accuracy: 0.0000e+00
Epoch 7/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7066 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 8/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7073 -
accuracy: 0.9508 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 9/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7218 -
accuracy: 0.9508 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 10/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7197 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 11/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7246 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 12/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6690 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 13/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7305 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 14/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7079 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 15/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7081 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 16/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7290 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 17/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7184 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 18/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7391 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 19/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7078 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 20/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6986 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 21/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6930 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
```

```
Epoch 22/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7024 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 23/40
105/105 [==============================] - 0s 2ms/step - loss: 0.7331 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 24/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7121 -
accuracy: 0.9512 - val_loss: 15.4257 - val_accuracy: 0.0000e+00
Epoch 25/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7128 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 26/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6756 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 27/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6817 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 28/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6868 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 29/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6753 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 30/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6847 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 31/40
105/105 [==============================] - 0s 1ms/step - loss: 0.7002 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 32/40
105/105 [==============================] - 0s 2ms/step - loss: 0.6334 -
accuracy: 0.9512 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 33/40
105/105 [==============================] - 0s 2ms/step - loss: 0.6548 -
accuracy: 0.9520 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 34/40
105/105 [==============================] - 0s 2ms/step - loss: 0.6666 -
accuracy: 0.9516 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 35/40
105/105 [==============================] - 0s 2ms/step - loss: 0.6679 -
accuracy: 0.9508 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 36/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6853 -
accuracy: 0.9532 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 37/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6880 -
accuracy: 0.9528 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
```

```
Epoch 38/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6672 -
accuracy: 0.9532 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 39/40
105/105 [==============================] - 0s 1ms/step - loss: 0.6862 -
accuracy: 0.9528 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Epoch 40/40
105/105 [==============================] - 0s 2ms/step - loss: 0.6979 -
accuracy: 0.9520 - val_loss: 15.4256 - val_accuracy: 0.0000e+00
Model: "sequential_40"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_117 (Dense)            (None, 10)                3370
_________________________________________________________________
dropout_69 (Dropout)         (None, 10)                0
_________________________________________________________________
dense_118 (Dense)            (None, 5)                 55
_________________________________________________________________
dropout_70 (Dropout)         (None, 5)                 0
_________________________________________________________________
dense_119 (Dense)            (None, 1)                 6
=================================================================
Total params: 3,431
Trainable params: 3,431
Non-trainable params: 0
_________________________________________________________________
```

[226]:
```
plt.subplot(2,2,3)
plt.plot(history.history['loss'])
plt.ylabel('Training Loss')
plt.xlabel('epochs')

plt.subplot(2,2,4)
plt.plot(history.history['val_loss'])
plt.ylabel('Validation Loss')
plt.xlabel('epochs')
```

[226]:
```
Text(0.5, 0, 'epochs')
```




```
[227]: # now lets see how well the model performed
yhat_proba = model.predict(X_test)
yhat = np.where(yhat_proba > 0.5, 1, 0)
print(mt.confusion_matrix(y_test,yhat))
print(mt.classification_report(y_test,yhat))
```

```
[[1064    0]
 [  29    0]]
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1064
           1       0.00      0.00      0.00        29

    accuracy                           0.97      1093
   macro avg       0.49      0.50      0.49      1093
weighted avg       0.95      0.97      0.96      1093
```

The model just predicts 0 for the entire dataset with larger batch size.

2 Conclusion

There does not seem to be anything really here in the data. It is tough to avoid phacking by continuing on and modifying and I do not really see any noticable improvement on a random guess. More data would most likely be helpful.