neural_net

July 6, 2021

[146]:
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

from sklearn import model_selection
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential, load_model
from keras.layers import LSTM, Dense, Dropout
```

[147]:
```
data = pd.read_csv("birth_data.csv")
data
```

[147]:

	age	st_toc	parity	st_mod
0	16	unbooked	1	SVD
1	20	unbooked	1	SVD
2	24	Booked	3	SVD
3	40	Booked	5	SVD
4	28	Booked	3	SVD
…	…	…	…	…
1120	30	Booked	3	SVD
1121	38	Booked	5	SVD
1122	23	Booked	2	SVD
1123	21	Booked	1	SVD
1124	15	Booked	1	SVD

[1125 rows x 4 columns]

[148]:
```
data["st_toc"] = data["st_toc"].str.lower()
data["st_mod"] = data["st_mod"].str.lower()

l_toc = LabelEncoder()
l_mod = LabelEncoder()
```

```
data['toc'] = l_toc.fit_transform(data['st_toc'])
data['mod'] = l_mod.fit_transform(data['st_mod'])

data = data.drop(['st_toc','st_mod'], axis='columns')
data
```

[148]:

	age	parity	toc	mod
0	16	1	1	1
1	20	1	1	1
2	24	3	0	1
3	40	5	0	1
4	28	3	0	1
…	…	…	…	…
1120	30	3	0	1
1121	38	5	0	1
1122	23	2	0	1
1123	21	1	0	1
1124	15	1	0	1

[1125 rows x 4 columns]

[149]:
```
data = data['mod'].values
data = data.reshape(-1, 1)
data
data[:5]
```

[149]:
```
array([[1],
       [1],
       [1],
       [1],
       [1]])
```

[150]:
```
data_train = np.array(data[:int(data.shape[0]*0.8)])
data_test = np.array(data[:int(data.shape[0]*0.8)-500:])
print(data_train.shape)
print(data_test.shape)
```

```
(900, 1)
(400, 1)
```

[151]:
```
scaler = MinMaxScaler(feature_range=(0,1))
data_train = scaler.fit_transform(data_train)
data_test = scaler.fit_transform(data_test)
print(data_train[:3])
print(data_train[:3])
```

```
[[1.]
```

```
 [1.]
 [1.]]
[[1.]
 [1.]
 [1.]]
```

[152]:
```
def create_dataset(data):
    x = []
    y = []
    for i in range(50, data.shape[0]):
        x.append(data[i-50:i,0])
        y.append(data[i, 0])
    x = np.array(x)
    y = np.array(y)
    return x,y
```

[153]:
```
x_train, y_train = create_dataset(data_train)
x_train[:1]
```

[153]:
```
array([[1., 1., 1., 1., 1., 1., 1., 0., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1.]])
```

[154]:
```
x_test, y_test = create_dataset(data_test)
x_test[:1]
```

[154]:
```
array([[1., 1., 1., 1., 1., 1., 1., 0., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1.]])
```

[155]:
```
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

[156]:
```
model = Sequential()
model.add(LSTM(units=96, return_sequences=True, input_shape=(x_train.shape[1],␣
↪1)))
model.add(Dropout(0.2))
model.add(LSTM(units=96, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=96, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=96))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

[166]:
```
adam = Adam(lr=0.001)
model.compile(loss='categorical_crossentropy', optimizer='rmsprop',␣
↪metrics=['accuracy'])
```

[167]:
```
history = model.fit(x_train, y_train, validation_data=(x_test, y_test),␣
↪epochs=50, batch_size=32)
```

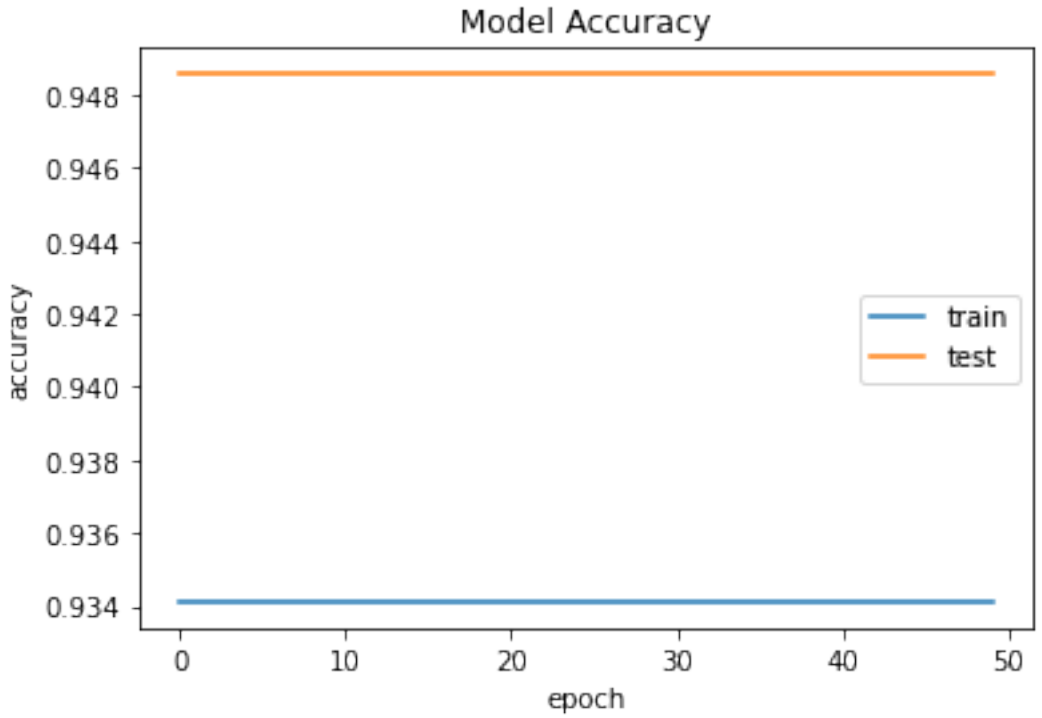
```
Epoch 1/50
27/27 [==============================] - 14s 259ms/step - loss: 1.1120e-07 -
accuracy: 0.9329 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 2/50
27/27 [==============================] - 5s 187ms/step - loss: 1.1020e-07 -
accuracy: 0.9244 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 3/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1122e-07 -
accuracy: 0.9329 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 4/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1304e-07 -
accuracy: 0.9483 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 5/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1050e-07 -
accuracy: 0.9270 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 6/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1094e-07 -
accuracy: 0.9306 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 7/50
27/27 [==============================] - 7s 276ms/step - loss: 1.1099e-07 -
accuracy: 0.9310 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 8/50
27/27 [==============================] - 5s 198ms/step - loss: 1.1380e-07 -
accuracy: 0.9546 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 9/50
27/27 [==============================] - 5s 191ms/step - loss: 1.1137e-07 -
accuracy: 0.9343 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 10/50
27/27 [==============================] - 5s 191ms/step - loss: 1.1106e-07 -
accuracy: 0.9317 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 11/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1195e-07 -
accuracy: 0.9391 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 12/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1259e-07 -
accuracy: 0.9445 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 13/50
27/27 [==============================] - 5s 191ms/step - loss: 1.1168e-07 -
accuracy: 0.9369 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 14/50
```

```
27/27 [==============================] - 5s 189ms/step - loss: 1.1205e-07 -
accuracy: 0.9400 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 15/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1329e-07 -
accuracy: 0.9503 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 16/50
27/27 [==============================] - 5s 192ms/step - loss: 1.1068e-07 -
accuracy: 0.9285 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 17/50
27/27 [==============================] - 5s 199ms/step - loss: 1.1250e-07 -
accuracy: 0.9437 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 18/50
27/27 [==============================] - 5s 193ms/step - loss: 1.1002e-07 -
accuracy: 0.9229 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 19/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1196e-07 -
accuracy: 0.9392 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 20/50
27/27 [==============================] - 5s 203ms/step - loss: 1.1242e-07 -
accuracy: 0.9431 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 21/50
27/27 [==============================] - 5s 192ms/step - loss: 1.1306e-07 -
accuracy: 0.9484 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 22/50
27/27 [==============================] - 5s 192ms/step - loss: 1.1136e-07 -
accuracy: 0.9342 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 23/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1228e-07 -
accuracy: 0.9418 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 24/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1152e-07 -
accuracy: 0.9355 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 25/50
27/27 [==============================] - 5s 191ms/step - loss: 1.0877e-07 -
accuracy: 0.9124 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 26/50
27/27 [==============================] - 5s 191ms/step - loss: 1.1277e-07 -
accuracy: 0.9460 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 27/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1103e-07 -
accuracy: 0.9314 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 28/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1241e-07 -
accuracy: 0.9429 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 29/50
27/27 [==============================] - 5s 190ms/step - loss: 1.0939e-07 -
accuracy: 0.9176 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 30/50
```

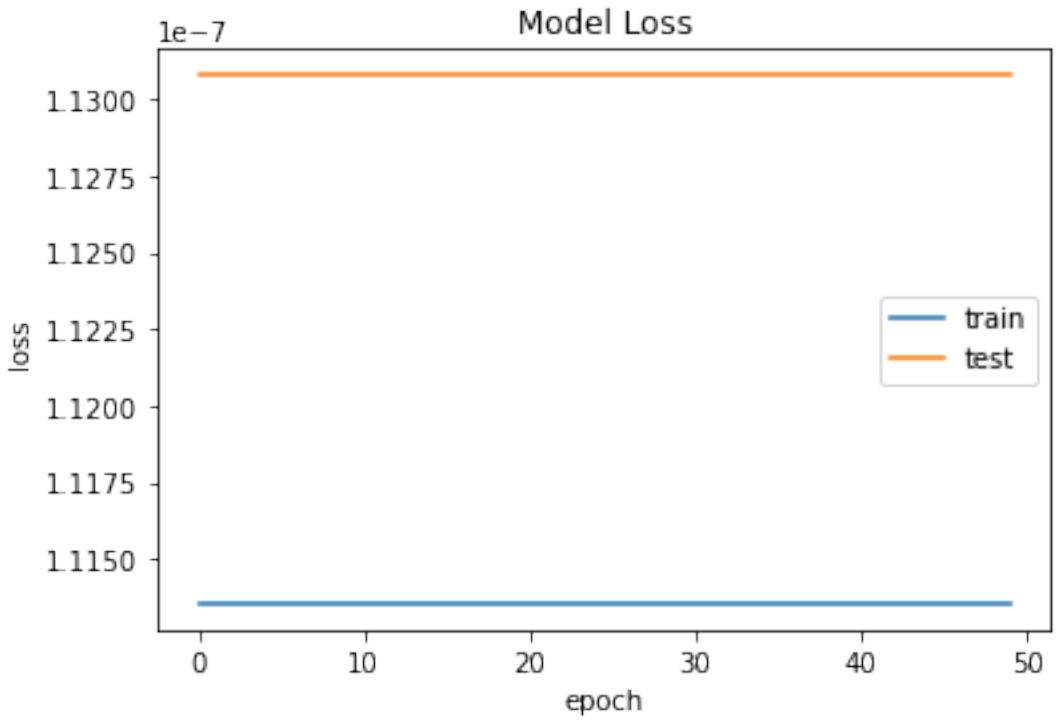
```
27/27 [==============================] - 8s 288ms/step - loss: 1.1216e-07 -
accuracy: 0.9409 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 31/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1185e-07 -
accuracy: 0.9383 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 32/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1162e-07 -
accuracy: 0.9363 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 33/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1050e-07 -
accuracy: 0.9269 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 34/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1100e-07 -
accuracy: 0.9311 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 35/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1135e-07 -
accuracy: 0.9341 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 36/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1106e-07 -
accuracy: 0.9317 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 37/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1160e-07 -
accuracy: 0.9362 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 38/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1184e-07 -
accuracy: 0.9382 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 39/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1202e-07 -
accuracy: 0.9397 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 40/50
27/27 [==============================] - 5s 190ms/step - loss: 1.0957e-07 -
accuracy: 0.9191 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 41/50
27/27 [==============================] - 5s 189ms/step - loss: 1.1030e-07 -
accuracy: 0.9253 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 42/50
27/27 [==============================] - 5s 200ms/step - loss: 1.1319e-07 -
accuracy: 0.9495 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 43/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1158e-07 -
accuracy: 0.9360 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 44/50
27/27 [==============================] - 5s 191ms/step - loss: 1.1022e-07 -
accuracy: 0.9246 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 45/50
27/27 [==============================] - 5s 190ms/step - loss: 1.0968e-07 -
accuracy: 0.9200 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 46/50
```

```
27/27 [==============================] - 5s 190ms/step - loss: 1.1285e-07 -
accuracy: 0.9467 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 47/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1086e-07 -
accuracy: 0.9300 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 48/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1113e-07 -
accuracy: 0.9322 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 49/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1185e-07 -
accuracy: 0.9383 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
Epoch 50/50
27/27 [==============================] - 5s 190ms/step - loss: 1.1147e-07 -
accuracy: 0.9351 - val_loss: 1.1308e-07 - val_accuracy: 0.9486
```

[168]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'])
plt.show()
```



[169]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'])
plt.show()
```



[172]:
```
from sklearn.metrics import classification_report, accuracy_score

score_pred = np.argmax(model.predict(x_test), axis=1)

print('Prediction Score')
print(accuracy_score(y_test, score_pred))
print(classification_report(y_test, score_pred))
```

```
Prediction Score
0.05142857142857143
              precision    recall  f1-score   support

         0.0       0.05      1.00      0.10        18
         1.0       0.00      0.00      0.00       332

    accuracy                           0.05       350
```

```
   macro avg       0.03      0.50      0.05       350
weighted avg       0.00      0.05      0.01       350

/home/el-sunais/anaconda3/lib/python3.8/site-
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/el-sunais/anaconda3/lib/python3.8/site-
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/el-sunais/anaconda3/lib/python3.8/site-
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[ ]:

```

```