universe

July 14, 2024

[ ]:
```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM, Dense
```

[ ]:
```
data = pd.read_csv('/home/darkcover/Documentos/Out/dados/Parte2/matrix.csv')
data.head()
```

[ ]:
```
   Unnamed: 0                                             Listas
0           0  [1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, …
1           1  [1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, …
2           2  [1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, …
3           3  [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, …
4           4  [1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, …
```

[ ]:
```
data = data.drop(columns=['Unnamed: 0'])
data.head()
```

[ ]:
```
                                              Listas
0  [1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, …
1  [1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, …
2  [1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, …
3  [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, …
4  [1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, …
```

[ ]:
```
data = data.drop(data.index[-1])
```

[ ]:
```
data.describe
```

[ ]:
```
<bound method NDFrame.describe of
Listas
0    [1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, …
1    [1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, …
2    [1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, …
3    [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, …
4    [1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, …
..                                                 …
```

```
410  [1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, …
411  [1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, …
412  [1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, …
413  [1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, …
414  [1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, …

[415 rows x 1 columns]>
```

[ ]:

```python
# Supondo que seus dados estão em um DataFrame chamado 'df'
# Convertendo a coluna de listas para um array numpy
data = data['Listas'].apply(lambda x: np.array(eval(x))).values
data = np.array([list(item) for item in data])
```

[ ]:

```python
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import LSTM, Dense


# Número de entradas iniciais a serem usadas
k = 160

# Dividindo dados em entradas (X) e saídas (y)
X = data[:, :k]
y = data[:, k:]

# Reformatar X para [samples, timesteps, features]
X = X.reshape((X.shape[0], X.shape[1], 1))

# Construindo o modelo LSTM
model = Sequential()
model.add(LSTM(640, activation='relu', input_shape=(k, 1)))
model.add(Dense(y.shape[1], activation='sigmoid'))  # Sigmoid para saída binária
model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])

# Treinando o modelo
model.fit(X, y, epochs=50, verbose=1, batch_size=640)

# Fazendo predições
predictions = (model.predict(X) > 0.5).astype(int)  # Convertendo␣
 ↪probabilidades em 0s e 1s

# Avaliando a precisão
accuracy = np.mean(predictions == y)
print(f"Accuracy: {accuracy:.2f}")
```

```
/home/darkcover/Documentos/Out/venv/lib/python3.10/site-
packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)
```

```
Epoch 1/50
1/1               24s 24s/step -
accuracy: 0.0000e+00 - loss: 0.6932
Epoch 2/50
1/1               21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6917
Epoch 3/50
1/1               22s 22s/step -
accuracy: 0.0000e+00 - loss: 0.6898
Epoch 4/50
1/1               35s 35s/step -
accuracy: 0.0000e+00 - loss: 0.6870
Epoch 5/50
1/1               20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6813
Epoch 6/50
1/1               21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6429
Epoch 7/50
1/1               16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6752
Epoch 8/50
1/1               16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6789
Epoch 9/50
1/1               21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6802
Epoch 10/50
1/1               19s 19s/step -
accuracy: 0.0000e+00 - loss: 0.6806
Epoch 11/50
1/1               20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6806
Epoch 12/50
1/1               15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6802
Epoch 13/50
1/1               14s 14s/step -
accuracy: 0.0000e+00 - loss: 0.6797
Epoch 14/50
1/1               14s 14s/step -
accuracy: 0.0000e+00 - loss: 0.6789
```

```
Epoch 15/50
1/1          21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6778
Epoch 16/50
1/1          15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6763
Epoch 17/50
1/1          15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6741
Epoch 18/50
1/1          20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6710
Epoch 19/50
1/1          15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6662
Epoch 20/50
1/1          15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6572
Epoch 21/50
1/1          19s 19s/step -
accuracy: 0.0000e+00 - loss: 0.6305
Epoch 22/50
1/1          14s 14s/step -
accuracy: 0.0000e+00 - loss: 0.6461
Epoch 23/50
1/1          21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6474
Epoch 24/50
1/1          15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6447
Epoch 25/50
1/1          16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6387
Epoch 26/50
1/1          20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6289
Epoch 27/50
1/1          21s 21s/step -
accuracy: 0.0000e+00 - loss: 1.0566
Epoch 28/50
1/1          20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6394
Epoch 29/50
1/1          20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6527
Epoch 30/50
1/1          19s 19s/step -
accuracy: 0.0000e+00 - loss: 0.6600
```

```
Epoch 31/50
1/1            21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6645
Epoch 32/50
1/1            15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6675
Epoch 33/50
1/1            17s 17s/step -
accuracy: 0.0000e+00 - loss: 0.6696
Epoch 34/50
1/1            19s 19s/step -
accuracy: 0.0000e+00 - loss: 0.6711
Epoch 35/50
1/1            16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6722
Epoch 36/50
1/1            20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6731
Epoch 37/50
1/1            21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6737
Epoch 38/50
1/1            19s 19s/step -
accuracy: 0.0000e+00 - loss: 0.6741
Epoch 39/50
1/1            22s 22s/step -
accuracy: 0.0000e+00 - loss: 0.6745
Epoch 40/50
1/1            20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6747
Epoch 41/50
1/1            21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6748
Epoch 42/50
1/1            21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6749
Epoch 43/50
1/1            20s 20s/step -
accuracy: 0.0000e+00 - loss: 0.6749
Epoch 44/50
1/1            15s 15s/step -
accuracy: 0.0000e+00 - loss: 0.6748
Epoch 45/50
1/1            16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6746
Epoch 46/50
1/1            18s 18s/step -
accuracy: 0.0000e+00 - loss: 0.6744
```

```
Epoch 47/50
1/1             12s 12s/step -
accuracy: 0.0000e+00 - loss: 0.6741
Epoch 48/50
1/1             24s 24s/step -
accuracy: 0.0000e+00 - loss: 0.6736
Epoch 49/50
1/1             16s 16s/step -
accuracy: 0.0000e+00 - loss: 0.6731
Epoch 50/50
1/1             21s 21s/step -
accuracy: 0.0000e+00 - loss: 0.6724
13/13             10s 734ms/step
Accuracy: 0.68
```

[ ]:

```
model.summary
```

[ ]:

```
<bound method Model.summary of <Sequential name=sequential, built=True>>
```