

```
In [1]: %pylab inline
import os
import keras
import metrics
import numpy as np
import pandas as pd
import keras.backend as K

from time import time

from keras import callbacks
from keras.models import Model
from keras.optimizers import SGD
from keras.layers import Dense, Input
from keras.initializers import VarianceScaling
from tensorflow.keras.layers import Layer, InputSpec
#from keras.engine.topology import Layer, InputSpec

from matplotlib.pyplot import imread
from sklearn import mixture
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, normalized_mutual_info_score
```

%pylab is deprecated, use %matplotlib inline and import the required libraries.
 Populating the interactive namespace from numpy and matplotlib

2022-06-06 12:25:34.059092: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerro
 r: libcudart.so.11.0: cannot open shared object file: No such file or direc
 tory; LD_LIBRARY_PATH: /usr/local/cuda-11.6/lib64:
 2022-06-06 12:25:34.059122: I tensorflow/stream_executor/cuda/cudart_stub.c
 c:29] Ignore above cudart dlerror if you do not have a GPU set up on your m
 achine.

Globus Data

```
In [2]: from pyriemann.estimation import Covariances
a1 = np.load('/media/ubuntu/HDD_P1/shubh/IITR/riem_total.npz')
test_data = a1['ts_total']
print(np.shape(test_data))

(900, 300)
```

```
In [3]: # this is our input placeholder
input_data = Input(shape=(300,))
```

```
In [8]: encoded = Dense(units=100, activation='relu')(input_data)
encoded = Dense(units=50, activation='relu')(encoded)
encoded = Dense(units=25, activation='relu')(encoded)
encoded = Dense(units=10, activation='relu')(encoded)
encoded = Dense(units=5, activation='relu')(encoded)
encoded = Dense(units=1, activation='relu')(encoded)

# The decoder
decoded = Dense(units=5, activation='relu')(encoded)
decoded = Dense(units=10, activation='relu')(decoded)
decoded = Dense(units=25, activation='relu')(decoded)
decoded = Dense(units=50, activation='relu')(decoded)
decoded = Dense(units=100, activation='relu')(decoded)
decoded = Dense(units=300, activation='sigmoid')(decoded)
```

```
# The autoencoder
autoencoder = Model(input_data, decoded)

# Compile the autoencoder
autoencoder.compile(optimizer='adam', loss='binary_crossentropy', metrics=[''])
# Train the autoencoder
autoencoder.fit(test_data, test_data, epochs=20, batch_size=256, shuffle=True)

# Save the model
autoencoder.save('autoencoder.h5')
```

```

Epoch 1/20
4/4 [=====] - 1s 6ms/step - loss: 0.6923 - accurac
y: 0.0067
Epoch 2/20
4/4 [=====] - 0s 6ms/step - loss: 0.6891 - accurac
y: 0.0000e+00
Epoch 3/20
4/4 [=====] - 0s 6ms/step - loss: 0.6835 - accurac
y: 0.0000e+00
Epoch 4/20
4/4 [=====] - 0s 6ms/step - loss: 0.6733 - accurac
y: 0.0000e+00
Epoch 5/20
4/4 [=====] - 0s 6ms/step - loss: 0.6539 - accurac
y: 0.0000e+00
Epoch 6/20
4/4 [=====] - 0s 6ms/step - loss: 0.6172 - accurac
y: 0.0000e+00
Epoch 7/20
4/4 [=====] - 0s 6ms/step - loss: 0.5479 - accurac
y: 0.0000e+00
Epoch 8/20
4/4 [=====] - 0s 6ms/step - loss: 0.4331 - accurac
y: 0.0000e+00
Epoch 9/20
4/4 [=====] - 0s 6ms/step - loss: 0.2701 - accurac
y: 0.0000e+00
Epoch 10/20
4/4 [=====] - 0s 5ms/step - loss: 0.1146 - accurac
y: 0.0000e+00
Epoch 11/20
4/4 [=====] - 0s 6ms/step - loss: 0.0057 - accurac
y: 0.0000e+00
Epoch 12/20
4/4 [=====] - 0s 6ms/step - loss: -0.0933 - accura
cy: 0.0000e+00
Epoch 13/20
4/4 [=====] - 0s 5ms/step - loss: -0.2672 - accura
cy: 0.0000e+00
Epoch 14/20
4/4 [=====] - 0s 5ms/step - loss: -0.6951 - accura
cy: 0.0000e+00
Epoch 15/20
4/4 [=====] - 0s 6ms/step - loss: -1.5810 - accura
cy: 0.0022
Epoch 16/20
4/4 [=====] - 0s 6ms/step - loss: -3.3996 - accura
cy: 0.0000e+00
Epoch 17/20
4/4 [=====] - 0s 6ms/step - loss: -6.8578 - accura
cy: 0.0011
Epoch 18/20
4/4 [=====] - 0s 6ms/step - loss: -13.5286 - accur
acy: 0.0011
Epoch 19/20
4/4 [=====] - 0s 6ms/step - loss: -26.3057 - accur
acy: 0.0011
Epoch 20/20
4/4 [=====] - 0s 5ms/step - loss: -47.7853 - accur
acy: 0.0056

```

```
In [9]: autoencoder.summary()
```

Model: "model_3"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 300)]	0
dense_36 (Dense)	(None, 100)	30100
dense_37 (Dense)	(None, 50)	5050
dense_38 (Dense)	(None, 25)	1275
dense_39 (Dense)	(None, 10)	260
dense_40 (Dense)	(None, 5)	55
dense_41 (Dense)	(None, 1)	6
dense_42 (Dense)	(None, 5)	10
dense_43 (Dense)	(None, 10)	60
dense_44 (Dense)	(None, 25)	275
dense_45 (Dense)	(None, 50)	1300
dense_46 (Dense)	(None, 100)	5100
dense_47 (Dense)	(None, 300)	30300

```

=====
Total params: 73,791
Trainable params: 73,791
Non-trainable params: 0
=====

```

```
In [10]: # this model maps an input to its encoded representation
encoder = Model(input_data, encoded)
```

```
In [11]: pred_test = encoder.predict(test_data)
```

```
29/29 [=====] - 0s 862us/step
```

```
In [12]: km = KMeans(n_clusters=2, n_init=20)
#km.fit(pred_auto_train)
km.fit(pred_test)
pred2 = km.predict(pred_test)
```

```
In [13]: #test data from globus dataset
normalized_mutual_info_score(test_label, pred2)
```

```
Unexpected exception formatting exception. Falling back to standard exception
```

```

Traceback (most recent call last):
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/interactiveshell.py", line 3397, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "/tmp/ipykernel_89765/4192888014.py", line 2, in <cell line: 2>
    normalized_mutual_info_score(test_label, pred2)
NameError: name 'test_label' is not defined

```

During handling of the above exception, another exception occurred:

```

Traceback (most recent call last):
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/interactiveshell.py", line 1992, in showtraceback
    stb = self.InteractiveTB.structured_traceback(
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/ultratb.py", line 1118, in structured_traceback
    return FormattedTB.structured_traceback(
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/ultratb.py", line 1012, in structured_traceback
    return VerboseTB.structured_traceback(
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/ultratb.py", line 865, in structured_traceback
    formatted_exception = self.format_exception_as_a_whole(etype, evalue, e
tb, number_of_lines_of_context,
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/ultratb.py", line 799, in format_exception_as_a_whole
    self.get_records(etb, number_of_lines_of_context, tb_offset) if etb els
e []
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/IPython/core/ultratb.py", line 845, in get_records
    style = stack_data.style_with_executing_node(style, "bg:ansiyellow")
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/stack_data/core.py", line 424, in style_with_executing_node
    class NewStyle(style):
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/pygments/style.py", line 91, in __new__
    ndef[4] = colorformat(styledef[3:])
  File "/home/ubuntu/anaconda3/envs/rapids-22.04/lib/python3.8/site-packages/pygments/style.py", line 58, in colorformat
    assert False, "wrong color format %r" % text
AssertionError: wrong color format 'ansiyellow'

```

```
In [14]: print(pred2)
```

```
In [16]: #counting final label for each subject based on maximum number of occurrence
from collections import Counter
m2 = []
for i in range(36):
    l1 = pred2[((26*i)-(i-1))-1:(25*(i+1)+1)-1]
    #print((range(((26*i)-(i-1))-1,(25*(i+1)+1)-1)))
    #print(l1.shape)
    most_common1,num_most_common1 = Counter(l1).most_common(1)[0] # 4, 6 ti
    #print(most_common, num_most_common)
    m2.append(most_common1)
tf1 = pd.DataFrame(m2).to_csv('/media/ubuntu/HDD_P2/MDD/final2/NN2.csv')
```

Out[17]: 28

Out[18]: 8

In []:

```
In [20]: df = pd.read_csv('/media/ubuntu/HDD_P2/MDD/gmm.csv')
print(df.head())
df1 = df['B'].values
print(df1[0])
```

```

print(len(df1))
from collections import Counter
m2 = []
for i in range(36):
    l1 = df1[((26*i)-(i-1))-1:(25*(i+1)+1)-1]
    #print((range((26*i)-(i-1))-1,(25*(i+1)+1)-1)))
    #print(l.shape)
    most_common1,num_most_common1 = Counter(l1).most_common(1)[0] # 4, 6 ti
    #print(most_common, num_most_common)
    m2.append(most_common1)
tf1 = pd.DataFrame(m2).to_csv('/media/ubuntu/HDD_P2/MDD/final2/gmm.csv')
df2 = pd.read_excel('/media/ubuntu/HDD_P2/MDD/final2/gmm2.xlsx')
print(df2.head())
df3 = df2['GMM'].values
print(df3[0])
print(len(df3))

```

```

      A  B
0  0  1
1  1  1
2  2  1
3  3  1
4  4  1
1
900
      Id  GMM
0     1     0
1     2     1
2     3     0
3     4     1
4     5     1
0
36

```

In [24]: `list(df3).count(0)`

Out[24]: 14

In [25]: `list(df3).count(1)`

Out[25]: 22

14 HC 22 MDD

In []: