Build Iteratively

March 30, 2020

[1]:
```
import pandas as pd
import numpy as np
import glob
import os
import random
import pickle
from sklearn.preprocessing import OneHotEncoder
from sklearn.utils import class_weight
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import datetime
import tensorflow.keras as keras
from tensorflow.keras import Sequential
from tensorflow.keras import regularizers
from tensorflow.keras.layers import LSTM, Dense, LSTM, Flatten,␣
↪BatchNormalization, Dropout
from keras.utils import to_categorical
import tensorflow as tf
# Load the TensorBoard notebook extension
%load_ext tensorboard
```

```
Using TensorFlow backend.
```

[2]:
```
countriesOfInterest = ["HK", "JP", 'ZA', 'TN', 'TR', 'GB', 'MX', 'US', 'CO',␣
↪'EC', 'AU', 'NZ']
countriesOfInterest = ["ZA", "EG", "TW", "JP", "DK", "FI", "US", "CA", "AU",␣
↪"NZ", "BR", "CO"]
train_n = 500
val_n = 20
Category = "Country"
w_length = 300
enc = OneHotEncoder()
enc.fit(np.array(countriesOfInterest).reshape(-1, 1))
```

[2]:
```
OneHotEncoder(categories='auto', drop=None, dtype=<class 'numpy.float64'>,
              handle_unknown='error', sparse=True)
```

[3]:
```python
def split(X, cat):
    X = X.reset_index()
    new_pos = list(X.track_id.index[X.track_id.shift(1) != X.track_id]) #␣
↪indices where the song changes
    new_pos.append(max(X.track_id.index) + 1) # add a new index to know where␣
↪the last song ends
    split_pos = []
    for i in range(len(new_pos)-1):
        split_pos = split_pos + list(range(new_pos[i], new_pos[i+1], w_length))
    split_pos = split_pos[1:]
    us_train = np.split(X.iloc[:,:24].to_numpy(), split_pos)
    labs = np.split(X[Category].to_numpy(), split_pos)
    # drop the short sequences
    short_seqs = []
    temp = []
    labels = []
    for i, value in enumerate(us_train):
        if value.shape[0] == w_length:
            temp.append(value)
            labels.append(labs[i][0])
    us_train = temp
    return np.stack(us_train), labels
```

[4]:
```python
def splitSeconds(n, country, t):
    data = pickle.load( open( "Raw Track Data\\" + country + "_" + t + ".p",␣
↪"rb" ) )
    tracks = data.track_id.unique()
    tracks = np.random.choice(tracks, size=n, replace=True)
    samples = []
    for track in tracks:
        try:
            trackFeats = data[data.track_id == track]
            FeatsLen =  trackFeats.shape[0]
            ind = random.randrange(1, FeatsLen - 10)
            feats = trackFeats.iloc[ind:ind+(w_length*10),6:30]
            dur = trackFeats.iloc[ind:ind+(w_length*10),1]
            example = np.array(feats.loc[feats.index.repeat(dur * 10)][-300:])
            if example.shape[0] == w_length:
                samples = samples + [example]
        except:
            continue
    samples = np.array(samples)
    return samples, np.repeat(np.array([country]), samples.shape[0])
```

[5]:
```python
def getSamples(train_n, val_n):
    train = pd.DataFrame()
    train_labels = pd.DataFrame()
```

```python
    val = pd.DataFrame()
    val_labels = pd.DataFrame()
    train_x = []
    train_labels = []
    val_x = []
    val_labels = []
    for country in countriesOfInterest:
        print("getting",country)
        x1, y1 = splitSeconds(train_n, country, "train")
        x2, y2 = splitSeconds(val_n, country, "val")
        train_x = train_x + x1.tolist()
        train_labels = train_labels + y1.tolist()
        val_x = val_x + x2.tolist()
        val_labels = val_labels + y2.tolist()
    #train_x = np.array(train_x)
    y = np.dstack(train_x)
    train_x = np.rollaxis(y,-1)
    train_labels = np.array(train_labels)
    #val_x = np.array(val_x)
    y = np.dstack(val_x)
    val_x = np.rollaxis(y,-1)
    val_labels = np.array(val_labels)
    class_weights = class_weight.compute_class_weight('balanced',
                                                      np.unique(train_labels),
                                                      list(train_labels))
    train_labels = enc.transform(np.array(train_labels).reshape(-1,1)).toarray()
    val_labels = enc.transform(np.array(val_labels).reshape(-1,1)).toarray()
    return train_x, train_labels, val_x, val_labels, class_weights
```

[6]:
```python
train_x, train_labels, val_x, val_labels, class_weights = getSamples(1, 1)
```

```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
```

0.0.1 Fit model

[7]:
```
enc.categories_
```

[7]:
```
[array(['AU', 'BR', 'CA', 'CO', 'DK', 'EG', 'FI', 'JP', 'NZ', 'TW', 'US',
        'ZA'], dtype='<U2')]
```

[8]:
```
train_x.shape
```

[8]:
```
(12, 300, 24)
```

[9]:
```
model = keras.Sequential()
model.add(LSTM(64,
               input_shape=(train_x.shape[1], train_x.shape[2]),
               return_sequences = False,
               kernel_regularizer=regularizers.l2(0.01)
              ))
model.add(Dropout(.5))
model.add(BatchNormalization())
model.add(Dense(len(enc.categories_[0]), activation= "softmax",␣
 ↪kernel_regularizer=regularizers.l2(0.01)))
adam = keras.optimizers.Adam(lr=0.001)
model.compile(loss = "categorical_crossentropy", optimizer= adam,␣
 ↪metrics=["acc"])
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 64)                22784
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
batch_normalization (BatchNo (None, 64)                256
_________________________________________________________________
dense (Dense)                (None, 12)                780
=================================================================
Total params: 23,820
Trainable params: 23,692
Non-trainable params: 128
_________________________________________________________________
None
```

[ ]:
```
desc = "64LSTMregularizedkernel_batchnorm_dropout_outputregularizedkernel"
log_dir = os.path.join(
    "logs",
    "iterative",
```

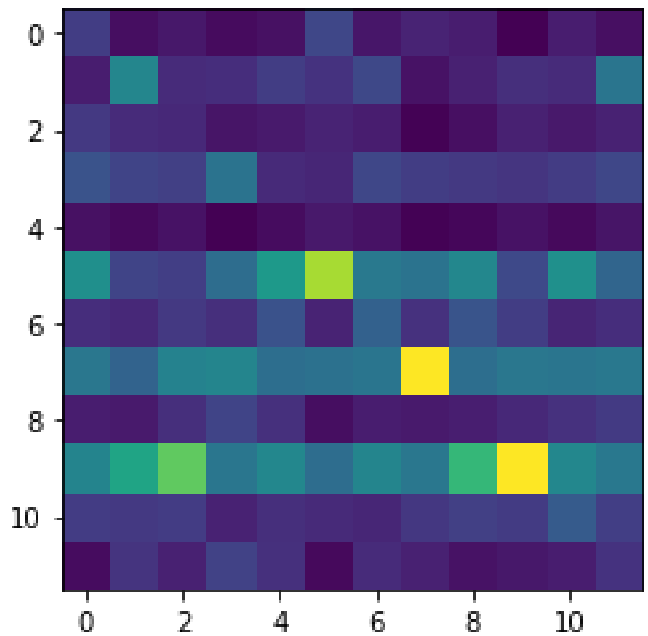
```
    desc
)
model_dir = os.path.join(
    "pickle",
    "save"
)

train_n = 5000
val_n = 1000
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,␣
 ↪histogram_freq=1)
epochs = 10
iterations = 10
learn_rate = 0.001
for i in range(iterations):
    adam = keras.optimizers.Adam(lr=learn_rate)
    model.compile(loss = "categorical_crossentropy", optimizer= adam,␣
 ↪metrics=["acc"])
    train_x, train_labels, val_x, val_labels, class_weights =␣
 ↪getSamples(train_n, val_n)
    print(np.sum(train_labels, axis = 0))
    model.fit(train_x, train_labels,
              epochs = i * epochs + epochs,
              initial_epoch = i * epochs,
              shuffle = True,
              validation_data = (val_x, val_labels),
              batch_size = 1024,
              class_weight = class_weights,
             callbacks=[tensorboard_callback],
             verbose = 1)
    model.save_weights(model_dir)
    if i%2 == 0:
        learn_rate = learn_rate/2
    if i % 1 == 0:
        preds = model.predict(val_x, batch_size = 1024, verbose = 1)
     #   print(np.sum(train_labels, axis = 0))
        plt.imshow(
            confusion_matrix(
                enc.inverse_transform(preds),
                enc.inverse_transform(val_labels),
              # normalize = "all"
            )
        )
        plt.pause(.5)
        plt.show()
        preds = model.predict(train_x, batch_size = 1024, verbose = 1)
        plt.imshow(
```

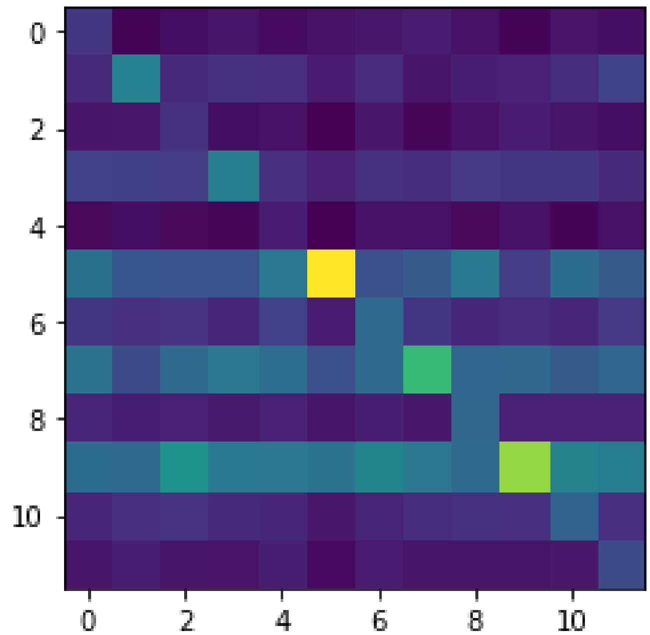
```
            confusion_matrix(
                enc.inverse_transform(preds),
                enc.inverse_transform(train_labels),
            #    normalize = "all"
            )
        )
        plt.pause(.5)
        plt.show()
```

```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4324. 4225. 4294. 4304. 4306. 4278. 4322. 4340. 4321. 4437. 4275. 4342.]
Train on 51768 samples, validate on 10376 samples
Epoch 1/10
51768/51768 [==============================] - 89s 2ms/sample - loss: 3.4267 -
acc: 0.0921 - val_loss: 2.9199 - val_acc: 0.1041
Epoch 2/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 3.0038 -
acc: 0.1079 - val_loss: 2.7752 - val_acc: 0.1259
Epoch 3/10
51768/51768 [==============================] - 79s 2ms/sample - loss: 2.8091 -
acc: 0.1188 - val_loss: 2.6966 - val_acc: 0.1252
Epoch 4/10
51768/51768 [==============================] - 82s 2ms/sample - loss: 2.6978 -
acc: 0.1279 - val_loss: 2.6488 - val_acc: 0.1292
Epoch 5/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 2.6310 -
acc: 0.1356 - val_loss: 2.6163 - val_acc: 0.1353
Epoch 6/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 2.5884 -
acc: 0.1450 - val_loss: 2.5901 - val_acc: 0.1411
Epoch 7/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 2.5580 -
acc: 0.1504 - val_loss: 2.5706 - val_acc: 0.1495
Epoch 8/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 2.5358 -
acc: 0.1546 - val_loss: 2.5570 - val_acc: 0.1470
```

```
Epoch 9/10
51768/51768 [==============================] - 84s 2ms/sample - loss: 2.5170 -
acc: 0.1630 - val_loss: 2.5458 - val_acc: 0.1510
Epoch 10/10
51768/51768 [==============================] - 83s 2ms/sample - loss: 2.5028 -
acc: 0.1626 - val_loss: 2.5365 - val_acc: 0.1557
10376/10376 [==============================] - 4s 411us/sample
```

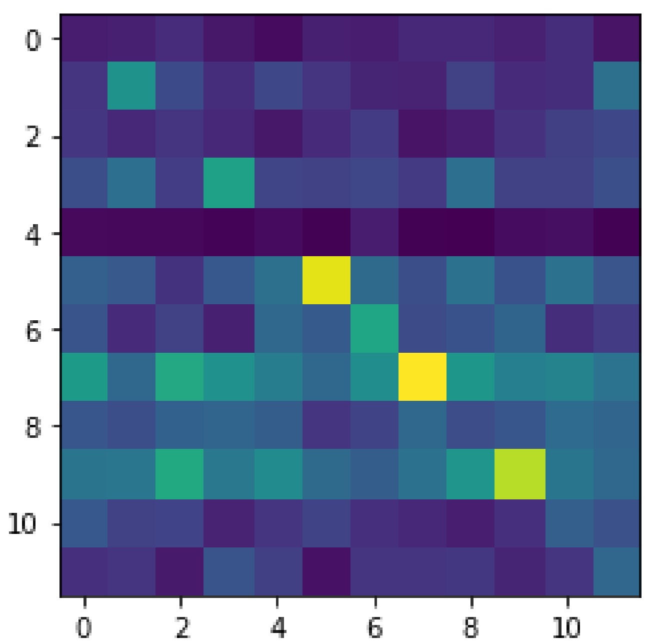



```
51768/51768 [==============================] - 20s 395us/sample
```

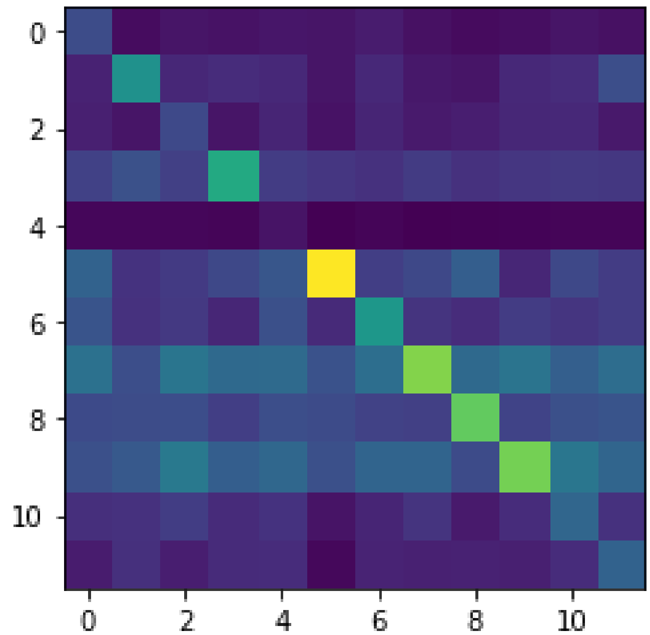



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4336. 4195. 4285. 4313. 4250. 4288. 4328. 4394. 4304. 4441. 4304. 4395.]
Train on 51833 samples, validate on 10405 samples
Epoch 11/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4992 -
acc: 0.1602 - val_loss: 2.5202 - val_acc: 0.1596
Epoch 12/20
51833/51833 [==============================] - 79s 2ms/sample - loss: 2.4791 -
acc: 0.1686 - val_loss: 2.5135 - val_acc: 0.1649
Epoch 13/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4705 -
acc: 0.1693 - val_loss: 2.5072 - val_acc: 0.1617
Epoch 14/20
51833/51833 [==============================] - 82s 2ms/sample - loss: 2.4597 -
acc: 0.1717 - val_loss: 2.5037 - val_acc: 0.1588
```

```
Epoch 15/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4503 -
acc: 0.1759 - val_loss: 2.4999 - val_acc: 0.1570
Epoch 16/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4432 -
acc: 0.1779 - val_loss: 2.4958 - val_acc: 0.1584
Epoch 17/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4334 -
acc: 0.1810 - val_loss: 2.4902 - val_acc: 0.1622
Epoch 18/20
51833/51833 [==============================] - 81s 2ms/sample - loss: 2.4246 -
acc: 0.1850 - val_loss: 2.4855 - val_acc: 0.1630
Epoch 19/20
51833/51833 [==============================] - 80s 2ms/sample - loss: 2.4192 -
acc: 0.1879 - val_loss: 2.4867 - val_acc: 0.1600
Epoch 20/20
51833/51833 [==============================] - 80s 2ms/sample - loss: 2.4117 -
acc: 0.1900 - val_loss: 2.4808 - val_acc: 0.1642
10405/10405 [==============================] - 4s 411us/sample
```

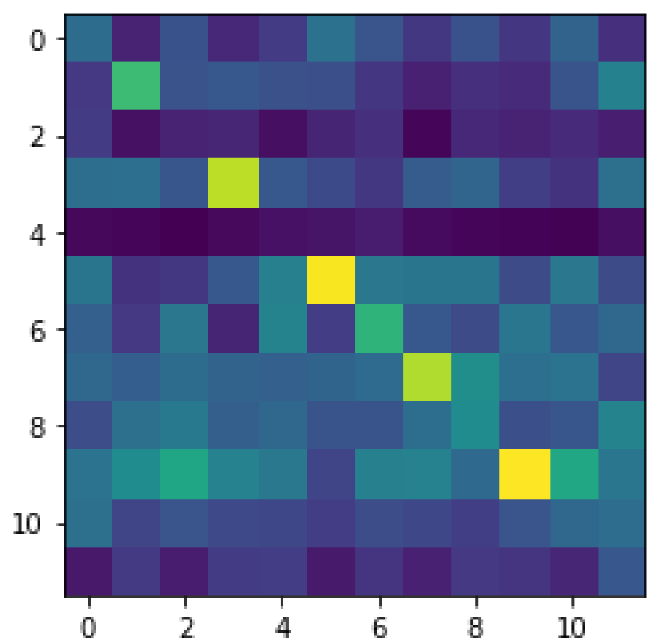



```
51833/51833 [==============================] - 20s 393us/sample
```

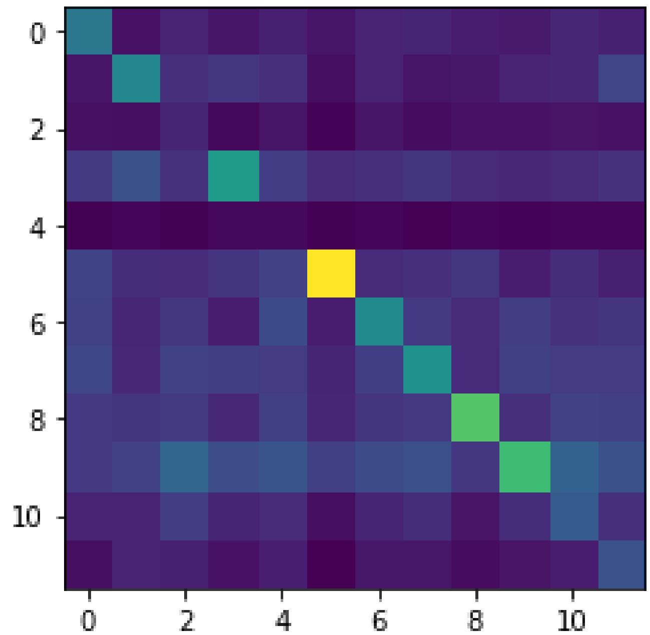



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4356. 4241. 4273. 4289. 4314. 4343. 4331. 4363. 4246. 4459. 4294. 4346.]
Train on 51855 samples, validate on 10337 samples
Epoch 21/30
51855/51855 [==============================] - 81s 2ms/sample - loss: 2.4141 -
acc: 0.1884 - val_loss: 2.4678 - val_acc: 0.1720
Epoch 22/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.4015 -
acc: 0.1948 - val_loss: 2.4638 - val_acc: 0.1736
Epoch 23/30
51855/51855 [==============================] - 79s 2ms/sample - loss: 2.3920 -
acc: 0.1990 - val_loss: 2.4620 - val_acc: 0.1667
Epoch 24/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3875 -
acc: 0.2004 - val_loss: 2.4629 - val_acc: 0.1705
```

```
Epoch 25/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3786 -
acc: 0.2013 - val_loss: 2.4628 - val_acc: 0.1722
Epoch 26/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3706 -
acc: 0.2082 - val_loss: 2.4590 - val_acc: 0.1706
Epoch 27/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3695 -
acc: 0.2076 - val_loss: 2.4574 - val_acc: 0.1756
Epoch 28/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3596 -
acc: 0.2121 - val_loss: 2.4590 - val_acc: 0.1728
Epoch 29/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3518 -
acc: 0.2151 - val_loss: 2.4559 - val_acc: 0.1736
Epoch 30/30
51855/51855 [==============================] - 80s 2ms/sample - loss: 2.3477 -
acc: 0.2169 - val_loss: 2.4521 - val_acc: 0.1721
10337/10337 [==============================] - 4s 431us/sample
```

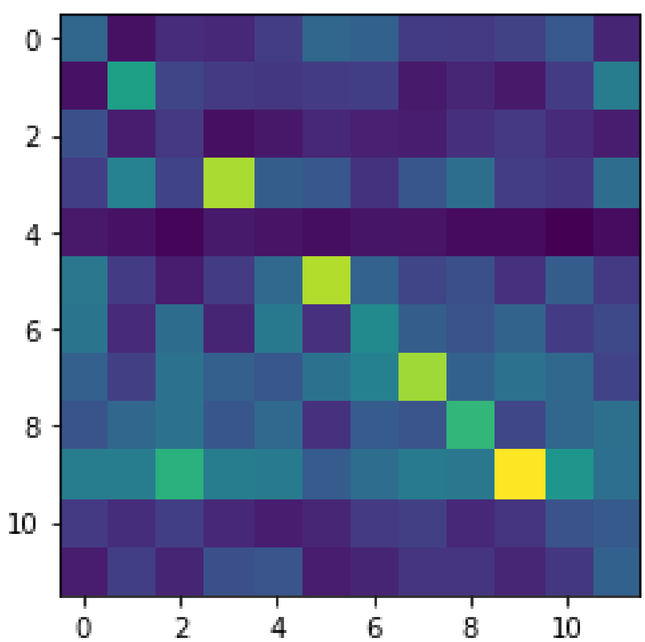



```
51855/51855 [==============================] - 21s 408us/sample
```

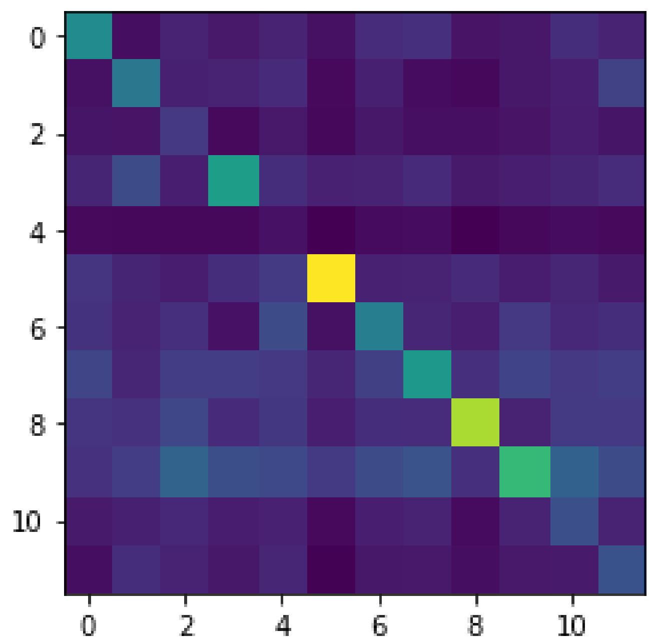



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4335. 4243. 4259. 4329. 4308. 4325. 4324. 4373. 4283. 4447. 4303. 4363.]
Train on 51892 samples, validate on 10289 samples
Epoch 31/40
51892/51892 [==============================] - 83s 2ms/sample - loss: 2.3542 -
acc: 0.2107 - val_loss: 2.4658 - val_acc: 0.1701
Epoch 32/40
51892/51892 [==============================] - 80s 2ms/sample - loss: 2.3430 -
acc: 0.2181 - val_loss: 2.4645 - val_acc: 0.1767
Epoch 33/40
51892/51892 [==============================] - 80s 2ms/sample - loss: 2.3337 -
acc: 0.2191 - val_loss: 2.4606 - val_acc: 0.1764
Epoch 34/40
51892/51892 [==============================] - 81s 2ms/sample - loss: 2.3326 -
acc: 0.2199 - val_loss: 2.4618 - val_acc: 0.1766
```

```
Epoch 35/40
51892/51892 [==============================] - 81s 2ms/sample - loss: 2.3249 -
acc: 0.2246 - val_loss: 2.4638 - val_acc: 0.1771
Epoch 36/40
51892/51892 [==============================] - 83s 2ms/sample - loss: 2.3202 -
acc: 0.2278 - val_loss: 2.4627 - val_acc: 0.1765
Epoch 37/40
51892/51892 [==============================] - 84s 2ms/sample - loss: 2.3206 -
acc: 0.2259 - val_loss: 2.4624 - val_acc: 0.1728
Epoch 38/40
51892/51892 [==============================] - 84s 2ms/sample - loss: 2.3174 -
acc: 0.2274 - val_loss: 2.4641 - val_acc: 0.1739
Epoch 39/40
51892/51892 [==============================] - 85s 2ms/sample - loss: 2.3121 -
acc: 0.2294 - val_loss: 2.4697 - val_acc: 0.1713
Epoch 40/40
51892/51892 [==============================] - 85s 2ms/sample - loss: 2.3071 -
acc: 0.2304 - val_loss: 2.4670 - val_acc: 0.1778
10289/10289 [==============================] - 4s 423us/sample
```

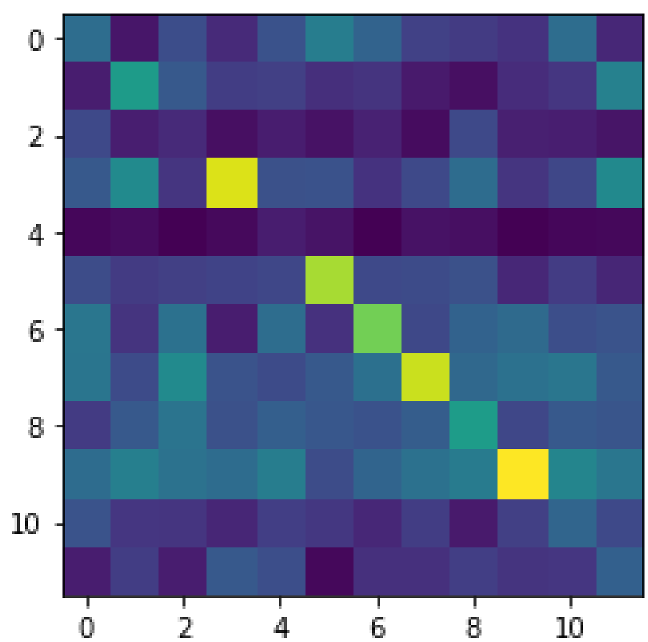



```
51892/51892 [==============================] - 21s 405us/sample
```

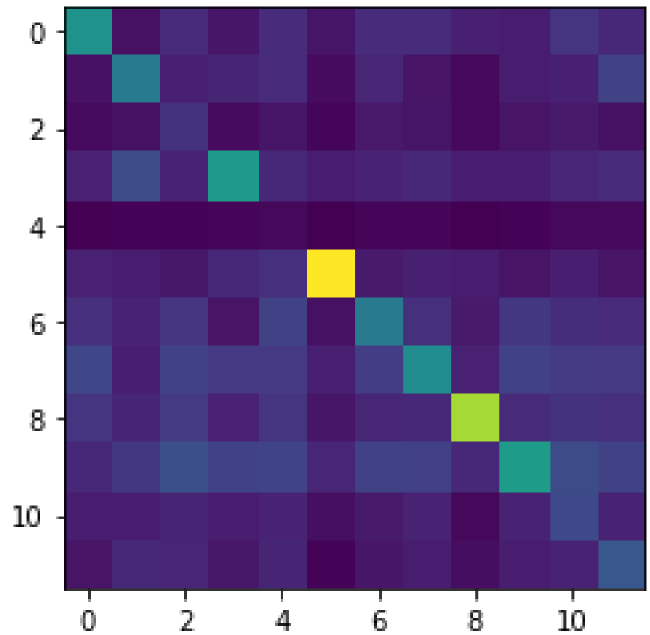



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4333. 4166. 4292. 4301. 4275. 4359. 4322. 4388. 4255. 4420. 4320. 4365.]
Train on 51796 samples, validate on 10403 samples
Epoch 41/50
51796/51796 [==============================] - 77s 1ms/sample - loss: 2.3214 -
acc: 0.2270 - val_loss: 2.4585 - val_acc: 0.1794
Epoch 42/50
51796/51796 [==============================] - 79s 2ms/sample - loss: 2.3080 -
acc: 0.2329 - val_loss: 2.4638 - val_acc: 0.1783
Epoch 43/50
51796/51796 [==============================] - 80s 2ms/sample - loss: 2.3058 -
acc: 0.2328 - val_loss: 2.4627 - val_acc: 0.1803
Epoch 44/50
51796/51796 [==============================] - 81s 2ms/sample - loss: 2.3002 -
acc: 0.2355 - val_loss: 2.4613 - val_acc: 0.1799
```

```
Epoch 45/50
51796/51796 [==============================] - 81s 2ms/sample - loss: 2.2959 -
acc: 0.2363 - val_loss: 2.4675 - val_acc: 0.1802
Epoch 46/50
51796/51796 [==============================] - 80s 2ms/sample - loss: 2.2941 -
acc: 0.2383 - val_loss: 2.4630 - val_acc: 0.1869
Epoch 47/50
51796/51796 [==============================] - 82s 2ms/sample - loss: 2.2865 -
acc: 0.2408 - val_loss: 2.4649 - val_acc: 0.1839
Epoch 48/50
51796/51796 [==============================] - 82s 2ms/sample - loss: 2.2860 -
acc: 0.2414 - val_loss: 2.4645 - val_acc: 0.1820
Epoch 49/50
51796/51796 [==============================] - 81s 2ms/sample - loss: 2.2813 -
acc: 0.2424 - val_loss: 2.4618 - val_acc: 0.1839
Epoch 50/50
51796/51796 [==============================] - 81s 2ms/sample - loss: 2.2802 -
acc: 0.2420 - val_loss: 2.4615 - val_acc: 0.1898
10403/10403 [==============================] - 5s 444us/sample
```

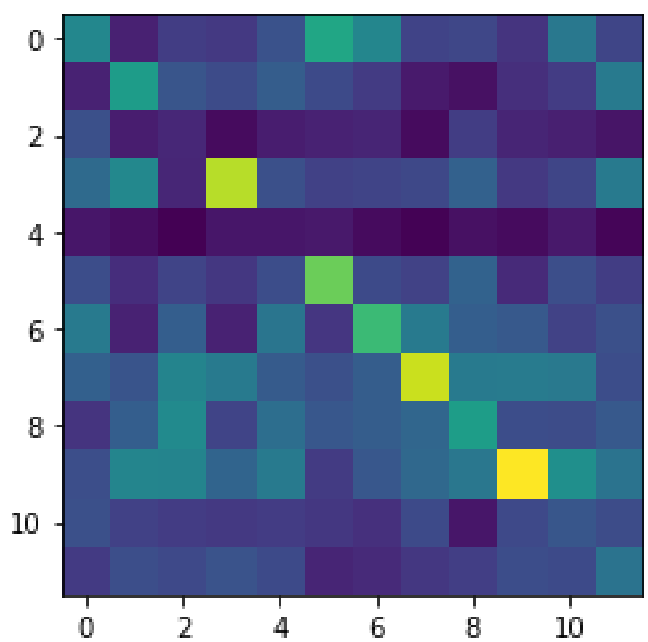



```
51796/51796 [==============================] - 22s 422us/sample
```

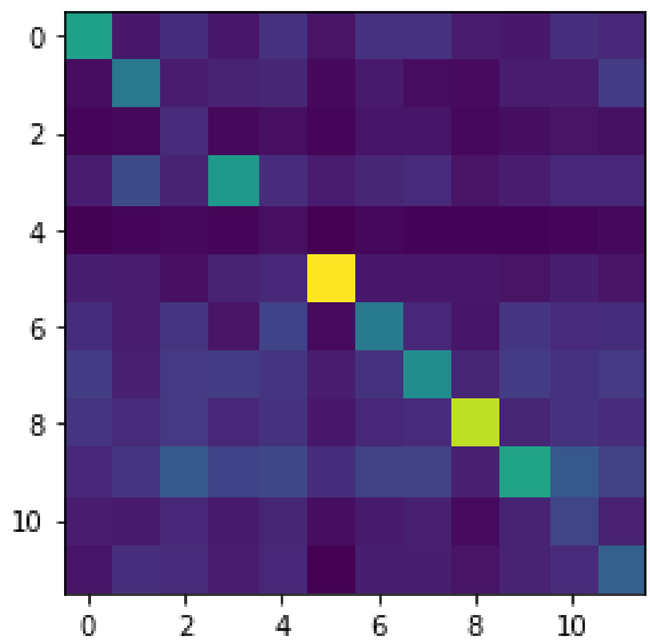



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4308. 4220. 4268. 4353. 4286. 4320. 4277. 4371. 4275. 4411. 4306. 4402.]
Train on 51797 samples, validate on 10438 samples
Epoch 51/60
51797/51797 [==============================] - 82s 2ms/sample - loss: 2.2902 -
acc: 0.2365 - val_loss: 2.4604 - val_acc: 0.1810
Epoch 52/60
51797/51797 [==============================] - 79s 2ms/sample - loss: 2.2834 -
acc: 0.2403 - val_loss: 2.4656 - val_acc: 0.1758
Epoch 53/60
51797/51797 [==============================] - 79s 2ms/sample - loss: 2.2763 -
acc: 0.2421 - val_loss: 2.4635 - val_acc: 0.1748
Epoch 54/60
51797/51797 [==============================] - 79s 2ms/sample - loss: 2.2722 -
acc: 0.2432 - val_loss: 2.4622 - val_acc: 0.1769
```

```
Epoch 55/60
51797/51797 [==============================] - 80s 2ms/sample - loss: 2.2697 -
acc: 0.2446 - val_loss: 2.4642 - val_acc: 0.1765
Epoch 56/60
51797/51797 [==============================] - 80s 2ms/sample - loss: 2.2679 -
acc: 0.2450 - val_loss: 2.4686 - val_acc: 0.1776
Epoch 57/60
51797/51797 [==============================] - 79s 2ms/sample - loss: 2.2658 -
acc: 0.2474 - val_loss: 2.4604 - val_acc: 0.1804
Epoch 58/60
51797/51797 [==============================] - 80s 2ms/sample - loss: 2.2600 -
acc: 0.2533 - val_loss: 2.4678 - val_acc: 0.1821
Epoch 59/60
51797/51797 [==============================] - 80s 2ms/sample - loss: 2.2609 -
acc: 0.2485 - val_loss: 2.4681 - val_acc: 0.1767
Epoch 60/60
51797/51797 [==============================] - 80s 2ms/sample - loss: 2.2597 -
acc: 0.2477 - val_loss: 2.4705 - val_acc: 0.1789
10438/10438 [==============================] - 4s 431us/sample
```

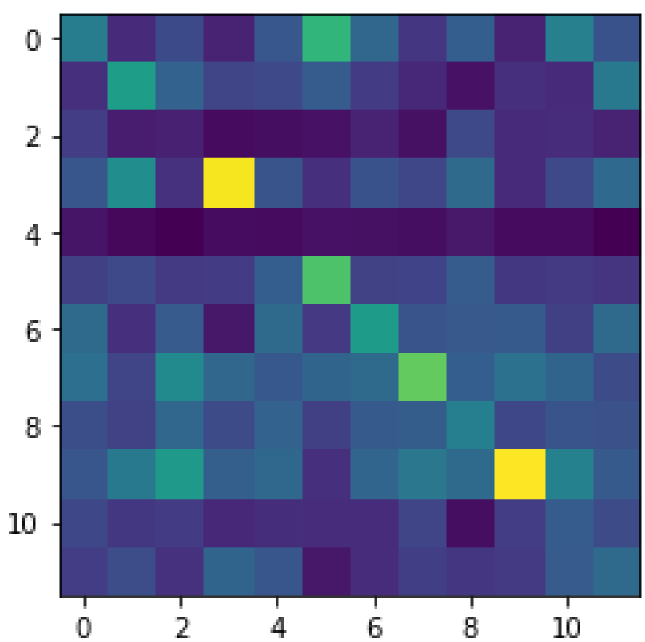



```
51797/51797 [==============================] - 21s 410us/sample
```

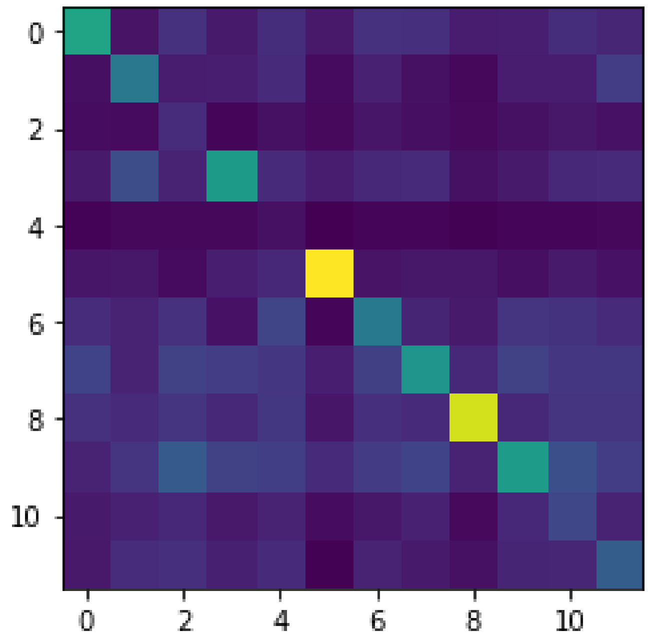



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4291. 4229. 4287. 4272. 4250. 4311. 4339. 4353. 4293. 4409. 4240. 4369.]
Train on 51643 samples, validate on 10348 samples
Epoch 61/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2680 -
acc: 0.2447 - val_loss: 2.4625 - val_acc: 0.1827
Epoch 62/70
51643/51643 [==============================] - 78s 2ms/sample - loss: 2.2620 -
acc: 0.2488 - val_loss: 2.4600 - val_acc: 0.1875
Epoch 63/70
51643/51643 [==============================] - 79s 2ms/sample - loss: 2.2594 -
acc: 0.2512 - val_loss: 2.4613 - val_acc: 0.1814
Epoch 64/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2594 -
acc: 0.2504 - val_loss: 2.4638 - val_acc: 0.1793
```

```
Epoch 65/70
51643/51643 [==============================] - 79s 2ms/sample - loss: 2.2554 -
acc: 0.2512 - val_loss: 2.4630 - val_acc: 0.1796
Epoch 66/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2485 -
acc: 0.2528 - val_loss: 2.4635 - val_acc: 0.1803
Epoch 67/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2457 -
acc: 0.2527 - val_loss: 2.4627 - val_acc: 0.1825
Epoch 68/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2427 -
acc: 0.2541 - val_loss: 2.4662 - val_acc: 0.1841
Epoch 69/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2406 -
acc: 0.2544 - val_loss: 2.4640 - val_acc: 0.1811
Epoch 70/70
51643/51643 [==============================] - 80s 2ms/sample - loss: 2.2450 -
acc: 0.2536 - val_loss: 2.4699 - val_acc: 0.1755
10348/10348 [==============================] - 5s 439us/sample
```

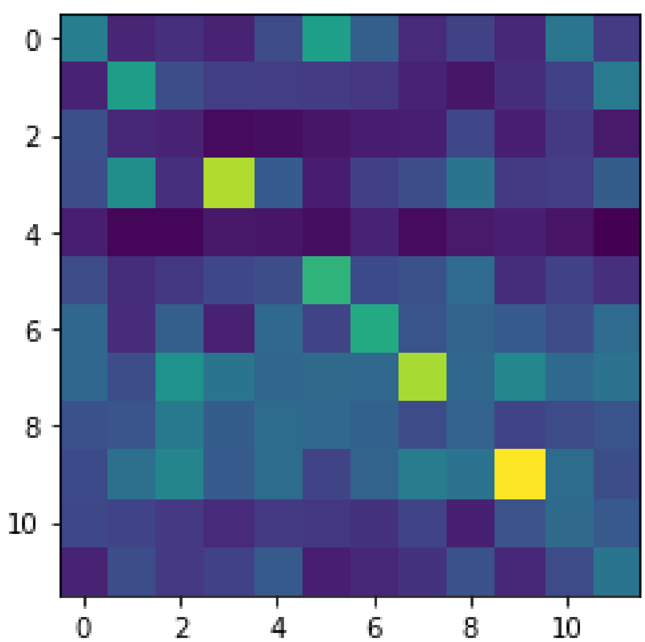



```
51643/51643 [==============================] - 22s 432us/sample
```

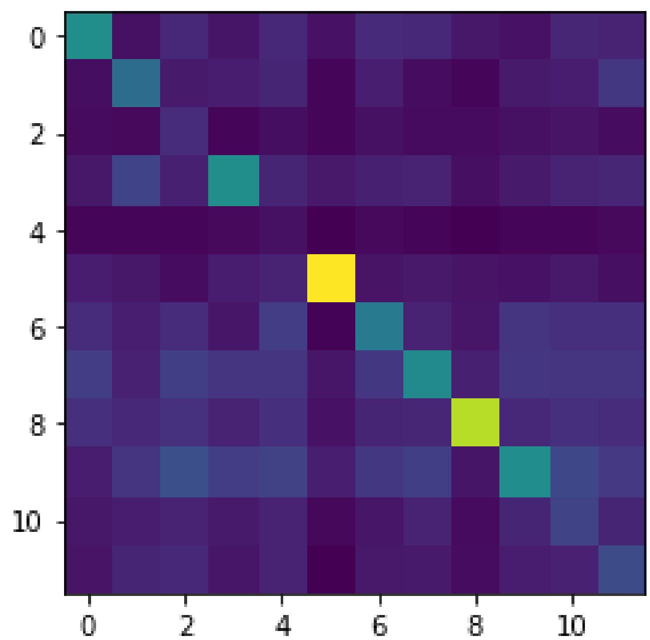



```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
getting CO
[4306. 4167. 4291. 4318. 4275. 4346. 4357. 4399. 4312. 4403. 4306. 4357.]
Train on 51837 samples, validate on 10379 samples
Epoch 71/80
51837/51837 [==============================] - 82s 2ms/sample - loss: 2.2470 -
acc: 0.2547 - val_loss: 2.4807 - val_acc: 0.1742
Epoch 72/80
51837/51837 [==============================] - 82s 2ms/sample - loss: 2.2443 -
acc: 0.2550 - val_loss: 2.4808 - val_acc: 0.1761
Epoch 73/80
51837/51837 [==============================] - 83s 2ms/sample - loss: 2.2375 -
acc: 0.2582 - val_loss: 2.4848 - val_acc: 0.1754
Epoch 74/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2368 -
acc: 0.2573 - val_loss: 2.4807 - val_acc: 0.1782
```

```
Epoch 75/80
51837/51837 [==============================] - 83s 2ms/sample - loss: 2.2362 -
acc: 0.2583 - val_loss: 2.4820 - val_acc: 0.1757
Epoch 76/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2378 -
acc: 0.2574 - val_loss: 2.4834 - val_acc: 0.1761
Epoch 77/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2300 -
acc: 0.2609 - val_loss: 2.4864 - val_acc: 0.1781
Epoch 78/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2310 -
acc: 0.2607 - val_loss: 2.4848 - val_acc: 0.1733
Epoch 79/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2303 -
acc: 0.2620 - val_loss: 2.4858 - val_acc: 0.1739
Epoch 80/80
51837/51837 [==============================] - 84s 2ms/sample - loss: 2.2299 -
acc: 0.2613 - val_loss: 2.4892 - val_acc: 0.1758
10379/10379 [==============================] - 5s 448us/sample
```




```
51837/51837 [==============================] - 22s 427us/sample
```




```
getting ZA
getting EG
getting TW
getting JP
getting DK
getting FI
getting US
getting CA
getting AU
getting NZ
getting BR
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: