```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import tensorflow as tf
from keras import layers
from keras import models
```

```
from google.colab import drive
drive.mount('/gdrive')
# drive.mount("/gdrive", force_remount=True)
%cd /gdrive/My\ Drive/Colab\ Notebooks
```

```
import pickle
f_link = "features.pkl"
pickle_in = open(f_link,"rb")
features = pickle.load(pickle_in)
features
```

```
flist = features.values.tolist()
```

Double-click (or enter) to edit

```
g_link = "genre_id_map.pkl"
pickle_in = open(g_link,"rb")
genre_map = pickle.load(pickle_in)
genre_map
```

	track_id	track_genres
0	2	[{'genre_id': '21', 'genre_title': 'Hip-Hop', ...
1	3	[{'genre_id': '21', 'genre_title': 'Hip-Hop', ...
2	5	[{'genre_id': '21', 'genre_title': 'Hip-Hop', ...
3	10	[{'genre_id': '10', 'genre_title': 'Pop', 'gen...
4	20	[{'genre_id': '76', 'genre_title': 'Experiment...
...	...	...
109722	155316	[{'genre_id': '25', 'genre_title': 'Punk', 'ge...
109723	155317	[{'genre_id': '25', 'genre_title': 'Punk', 'ge...
109724	155318	[{'genre_id': '25', 'genre_title': 'Punk', 'ge...
109725	155319	[{'genre_id': '25', 'genre_title': 'Punk', 'ge...
109726	155320	[{'genre_id': '10', 'genre_title': 'Pop', 'gen...

109727 rows × 2 columns

```
genre_ids = []
bad_count = 0
for i in range(0, len(genre_map)):
  try:
    curr_genre = eval(genre_map[0:]["track_genres"][i]) #list of genre
    genre_ids.append(curr_genre[0]['genre_id']) #first genre in list
  except:
    bad_count += 1
    # print(i, "problematic, listing as -1")
    genre_ids.append(-1)
```

```
print(bad_count, '/', len(genre_map), "bad", str(bad_count/len(genre_ma
# genre_ids
```

```
 2609 / 109727 bad 2.3777192486808167%
```

```
with open('genre_labels.pkl', 'wb') as f:
    pickle.dump(genre_ids, f)
with open('flist.pkl', 'wb') as f:
    pickle.dump(flist, f)
```

```
1
```

```
import sklearn.model_selection as model_selection

data = flist[:300]
labels = genre_ids[:300]
data_train, data_test, label_train, label_test = model_selection.train_
```

```
print(len(data_train), len(data_test), len(label_train), len(label_test
```

```
 225 75 225 75
```

```
from keras.utils.np_utils import to_categorical

one_hot_train_labels = to_categorical(label_train)
one_hot_test_labels = to_categorical(label_test)
one_hot_test_labels
```

```
```

```
array([[0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.]], dtype=float32)
```

```
from collections import Counter

len(Counter(genre_ids).keys()) # equals to list(set(words))
```

```
150
```

```
model = models.Sequential()
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='softmax')) # all genres
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l
```

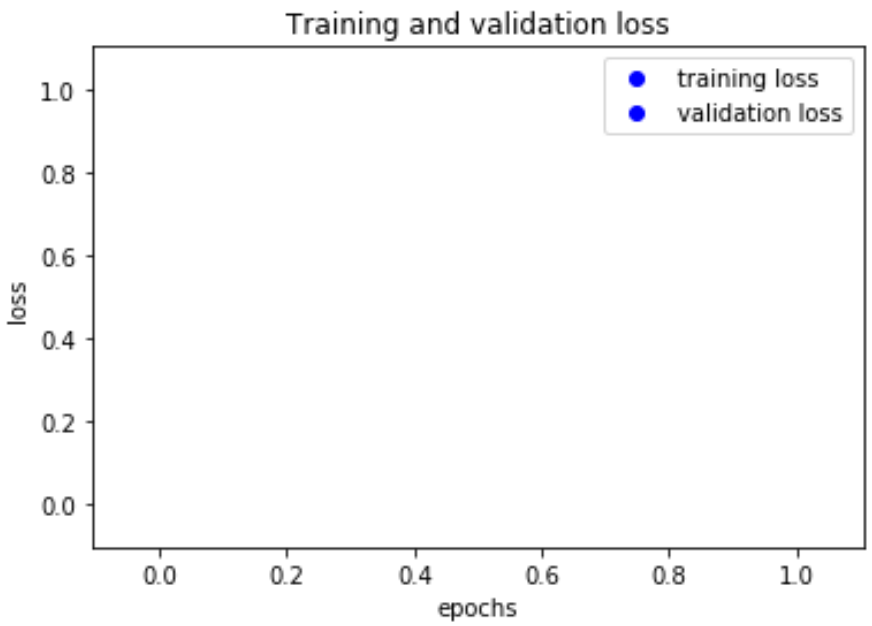
```
model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentrop
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/c
```

```
# x_train, x_test = data_train, data_test
x_train = np.array([np.array(i) for i in data_train])
x_test = np.array([np.array(i) for i in data_test])
# y_train, y_test = label_train, label_test
y_train = np.asarray(label_train, dtype=np.float32)
y_test = np.asarray(label_test, dtype=np.float32)

history = model.fit(x_train, y_train, epochs=10, batch_size=512, valida
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensor
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

Train on 150 samples, validate on 75 samples
Epoch 1/10
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/l

150/150 [==============================] - 5s 30ms/step - loss: nan -
Epoch 2/10
150/150 [==============================] - 0s 343us/step - loss: nan
Epoch 3/10
150/150 [==============================] - 0s 337us/step - loss: nan
Epoch 4/10
150/150 [==============================] - 0s 342us/step - loss: nan
Epoch 5/10
150/150 [==============================] - 0s 362us/step - loss: nan
Epoch 6/10
150/150 [==============================] - 0s 345us/step - loss: nan
Epoch 7/10
150/150 [==============================] - 0s 368us/step - loss: nan
Epoch 8/10
150/150 [==============================] - 0s 360us/step - loss: nan
Epoch 9/10
150/150 [==============================] - 0s 370us/step - loss: nan
Epoch 10/10
150/150 [==============================] - 0s 344us/step - loss: nan
```

```
import matplotlib.pyplot as plt

loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)

plt.plot(epochs, loss, 'bo', label="training loss")
plt.plot(epochs, val_loss, 'bo', label="validation loss")
plt.title("Training and validation loss")
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```




```
1
```