Crnn

June 11, 2020

0.0.1 -- *coding: utf-8* --

MusicTaggerCRNN model for Keras. ### Reference: - Music-auto_tagging-keras

[1]:
```
from keras import backend as K
from keras.layers import Input, Dense
from keras.models import Model
from keras.layers import Dense, Dropout, Reshape, Permute
from keras.layers.convolutional import Convolution2D
from keras.layers.convolutional import MaxPooling2D, ZeroPadding2D
from keras.layers.normalization import BatchNormalization
from keras.layers.advanced_activations import ELU
from keras.layers.recurrent import GRU
from keras.utils.data_utils import get_file
```

```
Using Theano backend.
WARNING (theano.tensor.blas): Using NumPy C-API based implementation for BLAS
functions.
```

[2]:
```
TH_WEIGHTS_PATH = 'https://github.com/keunwoochoi/music-auto_tagging-keras/blob/
 ↪master/data/music_tagger_crnn_weights_theano.h5'
TF_WEIGHTS_PATH = 'https://github.com/keunwoochoi/music-auto_tagging-keras/blob/
 ↪master/data/music_tagger_crnn_weights_tensorflow.h5'
```

1 MusicTaggerCRNN

Instantiate the MusicTaggerCRNN architecture, optionally loading weights pre-trained on Million Song Dataset. Note that when using TensorFlow, for best performance you should set image_dim_ordering="tf" in your Keras config at ~/.keras/keras.json.

The model and the weights are compatible with both TensorFlow and Theano. The dimension ordering convention used by the model is the one specified in your Keras config file.

For preparing mel-spectrogram input, see [audio_conv_utils.py] in applications. You will need to install Librosa to use it.

1.1 Arguments

weights: one of `None` (random initialization) or "msd" (pre-training on ImageNet). input_tensor: optional Keras tensor (i.e. output of `layers.Input()`) to use as image input for the model. in-

clude_top: whether to include the 1 fully-connected layer (output layer) at the top of the network. If False, the network outputs 32-dim features.

1.2 Returns

A Keras model instance.

[ ]:

[3]:
```
def MusicTaggerCRNN(weights='msd', input_tensor=None,
                    include_top=True):
    '''Instantiate the MusicTaggerCRNN architecture,
    optionally loading weights pre-trained
    on Million Song Dataset. Note that when using TensorFlow,
    for best performance you should set
    `image_dim_ordering="tf"` in your Keras config
    at ~/.keras/keras.json.
    The model and the weights are compatible with both
    TensorFlow and Theano. The dimension ordering
    convention used by the model is the one
    specified in your Keras config file.
    For preparing mel-spectrogram input, see
    `audio_conv_utils.py` in [applications](https://github.com/fchollet/keras/
↪tree/master/keras/applications).
    You will need to install [Librosa](http://librosa.github.io/librosa/)
    to use it.
    # Arguments
        weights: one of `None` (random initialization)
            or "msd" (pre-training on ImageNet).
        input_tensor: optional Keras tensor (i.e. output of `layers.Input()`)
            to use as image input for the model.
        include_top: whether to include the 1 fully-connected
            layer (output layer) at the top of the network.
            If False, the network outputs 32-dim features.
    # Returns
        A Keras model instance.
    '''
    if weights not in {'msd', None}:
        raise ValueError('The `weights` argument should be either '
                         '`None` (random initialization) or `msd` '
                         '(pre-training on Million Song Dataset).')

    # Determine proper input shape
    if keras.backend.image_data_format() == 'channels_first':
        input_shape = (1, 96, 1366)
    else:
        input_shape = (96, 1366, 1)
```

```
    if input_tensor is None:
        melgram_input = Input(shape=input_shape)
    else:
        if not K.is_keras_tensor(input_tensor):
            melgram_input = Input(tensor=input_tensor, shape=input_shape)
        else:
            melgram_input = input_tensor

    # Determine input axis
    if keras.backend.image_data_format() == 'channels_first':
        channel_axis = 1
        freq_axis = 2
        time_axis = 3
    else:
        channel_axis = 3
        freq_axis = 1
        time_axis = 2

    # Input block
    x = ZeroPadding2D(padding=(0, 37))(melgram_input)
    x = BatchNormalization(axis=freq_axis, name='bn_0_freq')(x)

    # Conv block 1
    x = Convolution2D(64, 3, 3, border_mode='same', name='conv1')(x)
    x = BatchNormalization(axis=channel_axis, mode=0, name='bn1')(x)
    x = ELU()(x)
    x = MaxPooling2D(pool_size=(2, 2), strides=(2, 2), name='pool1')(x)
    x = Dropout(0.1, name='dropout1')(x)

    # Conv block 2
    x = Convolution2D(128, 3, 3, border_mode='same', name='conv2')(x)
    x = BatchNormalization(axis=channel_axis, mode=0, name='bn2')(x)
    x = ELU()(x)
    x = MaxPooling2D(pool_size=(3, 3), strides=(3, 3), name='pool2')(x)
    x = Dropout(0.1, name='dropout2')(x)

    # Conv block 3
    x = Convolution2D(128, 3, 3, border_mode='same', name='conv3')(x)
    x = BatchNormalization(axis=channel_axis, mode=0, name='bn3')(x)
    x = ELU()(x)
    x = MaxPooling2D(pool_size=(4, 4), strides=(4, 4), name='pool3')(x)
    x = Dropout(0.1, name='dropout3')(x)

    # Conv block 4
    x = Convolution2D(128, 3, 3, border_mode='same', name='conv4')(x)
    x = BatchNormalization(axis=channel_axis, mode=0, name='bn4')(x)
    x = ELU()(x)
```

```
    x = MaxPooling2D(pool_size=(4, 4), strides=(4, 4), name='pool4')(x)
    x = Dropout(0.1, name='dropout4')(x)

    # reshaping
    if keras.backend.image_data_format() == 'channels_first':
        x = Permute((3, 1, 2))(x)
    x = Reshape((15, 128))(x)

    # GRU block 1, 2, output
    x = GRU(32, return_sequences=True, name='gru1')(x)
    x = GRU(32, return_sequences=False, name='gru2')(x)
    x = Dropout(0.3)(x)
    if include_top:
        x = Dense(10, activation='sigmoid', name='output')(x)

    # Create model
    model = Model(melgram_input, x)
    if weights is None:
        return model
    else:
        # Load input
        if keras.backend.image_data_format() == 'channels_last':
            raise RuntimeError("Please set keras.backend.image_data_format() ==␣
↪'channels_first'."
                               "You can set it at ~/.keras/keras.json")

        model.load_weights('data/music_tagger_crnn_weights_%s.h5' % K._BACKEND,
                           by_name=True)
        return model
```

[4]:
```
from IPython.display import Image
Image(filename='tf_th_keras_v2.png')
```

[4]:

Keras is ignoring the image_dim_ordering setting, because with **Keras v2** (v2.0.4 being the latest at the time of this writing) the name of the parameter has been changed.

Now you set it using the **image_data_format** parameter.

image_data_format can be set to **"channels_last"** or **"channels_first"**, which corresponds to the TensorFlow or Theano dimension orders respectively.

i.e. When using TensorFlow, you now set your **keras.json** like this,

```
{
  "image_data_format": "channels_last",
  "epsilon": 1e-07,
  "floatx": "float32",
  "backend": "tensorflow"
}
```

For Theano, you need to set it like this,

```
{
  "image_data_format": "channels_first",
  "epsilon": 1e-07,
  "floatx": "float32",
  "backend": "theano"
}
```

[5]:
```
import keras

if keras.backend.image_data_format() == 'channels_last':
    print("here backend is tensorflow" , keras.backend.image_data_format())
elif keras.backend.image_data_format() == 'channels_first':
    print("here backend is theano" , keras.backend.image_data_format())

#print(K.image_dim_ordering())
```

```
here backend is theano channels_first
```

TRY TO MAKE A MODEL USER-FRIENDLY IN NEAR FUTURE.

[6]:
```
model = MusicTaggerCRNN(weights=None)
```

```
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:62:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(64, (3, 3),
name="conv1", padding="same")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:63:
UserWarning: Update your `BatchNormalization` call to the Keras 2 API:
`BatchNormalization(axis=1, name="bn1")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:69:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
```

```
name="conv2", padding="same")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:70:
UserWarning: Update your `BatchNormalization` call to the Keras 2 API:
`BatchNormalization(axis=1, name="bn2")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:76:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
name="conv3", padding="same")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:77:
UserWarning: Update your `BatchNormalization` call to the Keras 2 API:
`BatchNormalization(axis=1, name="bn3")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:83:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
name="conv4", padding="same")`
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:84:
UserWarning: Update your `BatchNormalization` call to the Keras 2 API:
`BatchNormalization(axis=1, name="bn4")`
```

[7]:
```
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 1, 96, 1366)       0
_________________________________________________________________
zero_padding2d_1 (ZeroPaddin (None, 1, 96, 1440)       0
_________________________________________________________________
bn_0_freq (BatchNormalizatio (None, 1, 96, 1440)       384
_________________________________________________________________
conv1 (Conv2D)               (None, 64, 96, 1440)      640
_________________________________________________________________
bn1 (BatchNormalization)     (None, 64, 96, 1440)      256
_________________________________________________________________
elu_1 (ELU)                  (None, 64, 96, 1440)      0
_________________________________________________________________
pool1 (MaxPooling2D)         (None, 64, 48, 720)       0
_________________________________________________________________
dropout1 (Dropout)           (None, 64, 48, 720)       0
_________________________________________________________________
conv2 (Conv2D)               (None, 128, 48, 720)      73856
_________________________________________________________________
bn2 (BatchNormalization)     (None, 128, 48, 720)      512
_________________________________________________________________
elu_2 (ELU)                  (None, 128, 48, 720)      0
_________________________________________________________________
pool2 (MaxPooling2D)         (None, 128, 16, 240)      0
_________________________________________________________________
dropout2 (Dropout)           (None, 128, 16, 240)      0
_________________________________________________________________
```

```
conv3 (Conv2D)               (None, 128, 16, 240)      147584
_________________________________________________________________
bn3 (BatchNormalization)     (None, 128, 16, 240)      512
_________________________________________________________________
elu_3 (ELU)                  (None, 128, 16, 240)      0
_________________________________________________________________
pool3 (MaxPooling2D)         (None, 128, 4, 60)        0
_________________________________________________________________
dropout3 (Dropout)           (None, 128, 4, 60)        0
_________________________________________________________________
conv4 (Conv2D)               (None, 128, 4, 60)        147584
_________________________________________________________________
bn4 (BatchNormalization)     (None, 128, 4, 60)        512
_________________________________________________________________
elu_4 (ELU)                  (None, 128, 4, 60)        0
_________________________________________________________________
pool4 (MaxPooling2D)         (None, 128, 1, 15)        0
_________________________________________________________________
dropout4 (Dropout)           (None, 128, 1, 15)        0
_________________________________________________________________
permute_1 (Permute)          (None, 15, 128, 1)        0
_________________________________________________________________
reshape_1 (Reshape)          (None, 15, 128)           0
_________________________________________________________________
gru1 (GRU)                   (None, 15, 32)            15456
_________________________________________________________________
gru2 (GRU)                   (None, 32)                6240
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
output (Dense)               (None, 10)                330
=================================================================
Total params: 393,866
Trainable params: 392,778
Non-trainable params: 1,088
_________________________________________________________________
```

1.2.1 Compiling the model

[8]:
```
#compile model using accuracy to measure model performance
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

1.2.2 Load data

[9]:
```
import numpy as np
```

[10]:
```
concat_x = np.load('concat_x.npy')
concat_y = np.load('concat_y.npy')
```

[11]:
```
print(len(concat_x),'   ',(len(concat_y)) )
```

```
1000     1000
```

[12]:
```
train_x = concat_x[0:800]
train_y = concat_y[0:800]
```

[13]:
```
test_x = concat_x[800:1000]
test_y = concat_y[800:1000]
```

1.2.3 Training the model

[14]:
```
#train the model
model.fit(train_x, train_y, validation_data=(test_x, test_y), epochs=12 )
```

```
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
```

```
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"
/home/user/anaconda3/lib/python3.7/site-
packages/theano/scan_module/scan_perform_ext.py:76: UserWarning: The file
scan_perform.c is not available. This donot happen normally. You are probably in
a strangesetup. This mean Theano can not use the cython code for scan. If
youwant to remove this warning, use the Theano flag'cxx=' (set to an empty
string) to disable all ccode generation.
  "The file scan_perform.c is not available. This do"

Train on 800 samples, validate on 200 samples
Epoch 1/12
800/800 [==============================] - 1768s 2s/step - loss: 2.1622 - acc:
0.1988 - val_loss: 2.3895 - val_acc: 0.0850
Epoch 2/12
800/800 [==============================] - 1633s 2s/step - loss: 2.0017 - acc:
0.3050 - val_loss: 2.4312 - val_acc: 0.1500
Epoch 3/12
800/800 [==============================] - 1644s 2s/step - loss: 1.8741 - acc:
0.3837 - val_loss: 2.2725 - val_acc: 0.2200
Epoch 4/12
800/800 [==============================] - 1671s 2s/step - loss: 1.8218 - acc:
0.4300 - val_loss: 2.2619 - val_acc: 0.1400
Epoch 5/12
800/800 [==============================] - 1681s 2s/step - loss: 1.7332 - acc:
0.4738 - val_loss: 2.3157 - val_acc: 0.1600
Epoch 6/12
800/800 [==============================] - 1656s 2s/step - loss: 1.6637 - acc:
0.5038 - val_loss: 2.2028 - val_acc: 0.2100
Epoch 7/12
800/800 [==============================] - 1656s 2s/step - loss: 1.6061 - acc:
0.4900 - val_loss: 2.1739 - val_acc: 0.2100
Epoch 8/12
800/800 [==============================] - 1682s 2s/step - loss: 1.5785 - acc:
0.4863 - val_loss: 2.0594 - val_acc: 0.1950
Epoch 9/12
800/800 [==============================] - 1623s 2s/step - loss: 1.5061 - acc:
0.5087 - val_loss: 2.2164 - val_acc: 0.1550
Epoch 10/12
800/800 [==============================] - 1675s 2s/step - loss: 1.4251 - acc:
0.5613 - val_loss: 1.9138 - val_acc: 0.1500
```

```
Epoch 11/12
800/800 [==============================] - 1677s 2s/step - loss: 1.4085 - acc:
0.5437 - val_loss: 2.1111 - val_acc: 0.1050
Epoch 12/12
800/800 [==============================] - 1627s 2s/step - loss: 1.3380 - acc:
0.5962 - val_loss: 1.9574 - val_acc: 0.2600
```

[14]:
```
<keras.callbacks.History at 0x7f1954887f50>
```

[15]:
```
from keras.models import load_model

model.save('Crnn.h5')  # creates a HDF5 file 'my_model.h5'
del model  # deletes the existing model

# returns a compiled model
# identical to the previous one
#model = load_model('k2c2.h5')
```

[ ]:
```

```