AlexNet_

April 17, 2021

Name: Ventrapragada Sai Shravani
PRN:17070123120
Batch: G-5 (2017-21)

[1]:
```
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
# Dealing with 2D data
from keras.layers.normalization import BatchNormalization
```

[2]:
```
!pip install tflearn
```

```
Collecting tflearn
  Downloading https://files.pythonhosted.org/packages/e7/3c/0b156d08ef3d4e
2a8009ecab2af1ad2e304f6fb99562b6271c68a74a4397/tflearn-0.5.0.tar.gz (107kB)
     || 112kB 17.6MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-
packages (from tflearn) (1.19.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from tflearn) (1.15.0)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages
(from tflearn) (7.1.2)
Building wheels for collected packages: tflearn
  Building wheel for tflearn (setup.py) ... done
  Created wheel for tflearn: filename=tflearn-0.5.0-cp37-none-any.whl
size=127301
sha256=8e82407fc6d182708228a255d02d7d05a0a9b91e2d5c7277a06a66eccdec054f
  Stored in directory: /root/.cache/pip/wheels/31/d2/ed/fb9a0d301dd9586c11e95471
20278e624227f22fd5f4baf744
Successfully built tflearn
Installing collected packages: tflearn
Successfully installed tflearn-0.5.0
```

Load and preprocess data
One- hot: only true value is one

[3]:
```
import tflearn.datasets.oxflower17 as oxflower17
X, Y = oxflower17.load_data(one_hot=True)
```

```
# Large image dataset
```

```
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-
packages/tensorflow/python/compat/v2_compat.py:96: disable_resource_variables
(from tensorflow.python.ops.variable_scope) is deprecated and will be removed in
a future version.
Instructions for updating:
non-resource variables are not supported in the long term
Downloading Oxford 17 category Flower Dataset, Please wait...

100.0% 60276736 / 60270631

Succesfully downloaded 17flowers.tgz 60270631 bytes.
File Extracted
Starting to parse images...
Parsing Done!
```

[4]:

```
model = Sequential()

model.add(Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activation='relu',␣
 ↪input_shape=(224, 224, 3)))
#Stride: How many cells the image has to be moved
# 3 channel input layer

model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
# Making it zero centered - more learning
model.add(BatchNormalization())

model.add(Conv2D(256, kernel_size=(5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(4096, activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='tanh'))
model.add(Dropout(0.5))

model.add(Dense(17, activation='softmax'))
# Classification of 17 classes
```

[5]:

```python
model.summary()
#96*11*11*3 + 96 (bias)= 34944
```

```
Model: "sequential"
-----------------------------------------------------------------
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 54, 54, 96)        34944
-----------------------------------------------------------------
max_pooling2d (MaxPooling2D) (None, 26, 26, 96)        0
-----------------------------------------------------------------
batch_normalization (BatchNo (None, 26, 26, 96)        384
-----------------------------------------------------------------
conv2d_1 (Conv2D)            (None, 22, 22, 256)       614656
-----------------------------------------------------------------
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 256)       0
-----------------------------------------------------------------
batch_normalization_1 (Batch (None, 10, 10, 256)       1024
-----------------------------------------------------------------
conv2d_2 (Conv2D)            (None, 8, 8, 256)         590080
-----------------------------------------------------------------
conv2d_3 (Conv2D)            (None, 6, 6, 384)         885120
-----------------------------------------------------------------
conv2d_4 (Conv2D)            (None, 4, 4, 384)         1327488
-----------------------------------------------------------------
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 384)         0
-----------------------------------------------------------------
batch_normalization_2 (Batch (None, 1, 1, 384)         1536
-----------------------------------------------------------------
flatten (Flatten)            (None, 384)               0
-----------------------------------------------------------------
dense (Dense)                (None, 4096)              1576960
-----------------------------------------------------------------
dropout (Dropout)            (None, 4096)              0
-----------------------------------------------------------------
dense_1 (Dense)              (None, 4096)              16781312
-----------------------------------------------------------------
dropout_1 (Dropout)          (None, 4096)              0
-----------------------------------------------------------------
dense_2 (Dense)              (None, 17)                69649
=================================================================
Total params: 21,883,153
Trainable params: 21,881,681
Non-trainable params: 1,472

-----------------------------------------------------------------
```

[6]:

```
model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
```

[7]:

```
history= model.fit(X, Y, batch_size=64, epochs=10, verbose=1,␣
 ↪validation_split=0.1, shuffle=True)
# Shuffling will occur
```

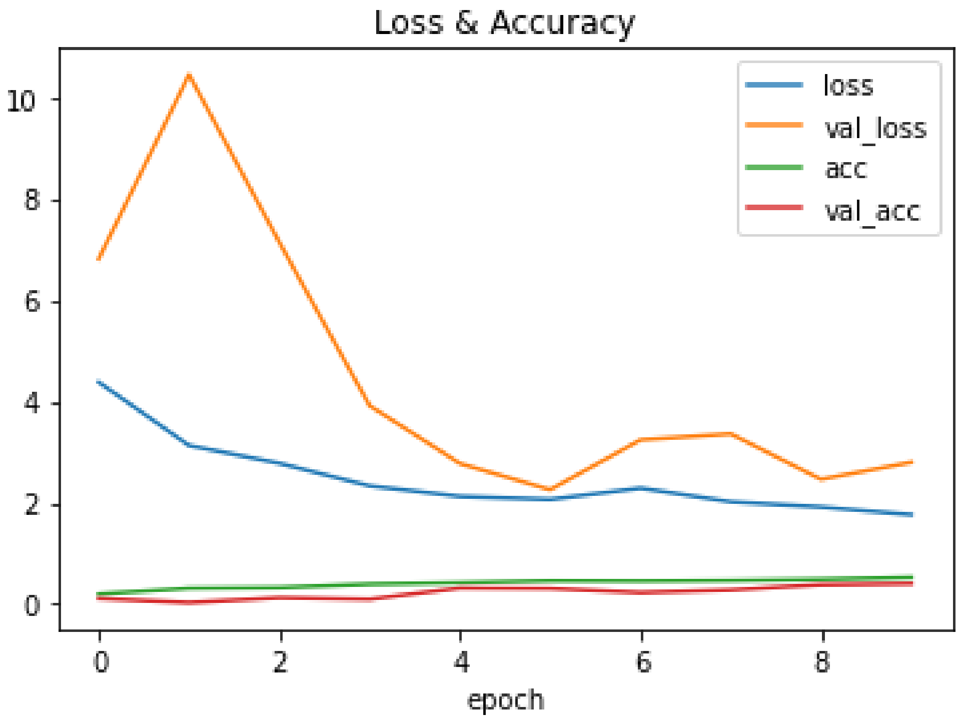
```
Train on 1224 samples, validate on 136 samples
Epoch 1/10
1224/1224 [==============================] - 34s 28ms/sample - loss: 4.3775 -
acc: 0.1985 - val_loss: 6.8056 - val_acc: 0.1103
Epoch 2/10
  64/1224 [>...] - ETA: 0s - loss: 2.4521 - acc:

/usr/local/lib/python3.7/dist-
packages/tensorflow/python/keras/engine/training.py:2325: UserWarning:
`Model.state_updates` will be removed in a future version. This property should
not be used in TensorFlow 2.0, as `updates` are applied automatically.
  warnings.warn('`Model.state_updates` will be removed in a future version. '

1224/1224 [==============================] - 1s 889us/sample - loss: 3.1263 -
acc: 0.3129 - val_loss: 10.4441 - val_acc: 0.0294
Epoch 3/10
1224/1224 [==============================] - 1s 882us/sample - loss: 2.7793 -
acc: 0.3243 - val_loss: 7.1387 - val_acc: 0.1250
Epoch 4/10
1224/1224 [==============================] - 1s 894us/sample - loss: 2.3327 -
acc: 0.3938 - val_loss: 3.9124 - val_acc: 0.0956
Epoch 5/10
1224/1224 [==============================] - 1s 895us/sample - loss: 2.1247 -
acc: 0.4224 - val_loss: 2.7741 - val_acc: 0.3162
Epoch 6/10
1224/1224 [==============================] - 1s 885us/sample - loss: 2.0648 -
acc: 0.4608 - val_loss: 2.2577 - val_acc: 0.3088
Epoch 7/10
1224/1224 [==============================] - 1s 901us/sample - loss: 2.2878 -
acc: 0.4559 - val_loss: 3.2386 - val_acc: 0.2279
Epoch 8/10
1224/1224 [==============================] - 1s 899us/sample - loss: 2.0187 -
acc: 0.4665 - val_loss: 3.3568 - val_acc: 0.2794
Epoch 9/10
1224/1224 [==============================] - 1s 891us/sample - loss: 1.9156 -
acc: 0.4877 - val_loss: 2.4631 - val_acc: 0.3824
Epoch 10/10
1224/1224 [==============================] - 1s 892us/sample - loss: 1.7646 -
acc: 0.5261 - val_loss: 2.7989 - val_acc: 0.4044
```

[10]:

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.legend(['loss', 'val_loss', 'acc', 'val_acc'])
plt.title('Loss & Accuracy')
plt.xlabel('epoch')
```

[10]:

```
Text(0.5, 0, 'epoch')
```



[14]:

```
score = model.evaluate(X, Y, verbose = 1)
print('Test Score', score[0])
print('Test Accuracy', score[1])
```

```
Test Score 2.3524821646073284
Test Accuracy 0.3992647
```

[ ]:

```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('AlexNet_.ipynb')
```

```
--2021-04-17 06:46:48--  https://raw.githubusercontent.com/brpy/colab-
pdf/master/colab_pdf.py
```

```
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: colab_pdf.py

colab_pdf.py        100%[===================>]   1.82K  --.-KB/s    in 0s

2021-04-17 06:46:49 (27.6 MB/s) - colab_pdf.py saved [1864/1864]

Mounted at /content/drive/

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```