



Search



AAYUSH SHAH · 1MO AGO · 7 VIEWS · PRIVATE



cifar-10 | convolutional layer

Python · CIFAR-10 - Object Recognition in Images



Notebook Data Logs Comments (0) Settings



Competition Notebook

CIFAR-10 - Object Recognition in Images

Run

5990.3s

Private Score

0.74990

Public Score

0.74990

Best Score

0.7499 V1

Version 1 of 1

Add Tags

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from tensorflow import keras
import tensorflow as tf
from keras import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Activation, BatchNormalization, Flatten
from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

/kaggle/input/cifar-10/trainLabels.csv
/kaggle/input/cifar-10/sampleSubmission.csv
/kaggle/input/cifar-10/test.7z
/kaggle/input/cifar-10/train.7z

In [2]:

```
!pip install py7zr
```

```
Collecting py7zr
  Downloading py7zr-0.18.3-py3-none-any.whl (76 kB)
[██████████| 76 kB 1.4 MB/s]
Collecting zipfile-deflate64>=0.2.0
  Downloading zipfile_deflate64-0.2.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (43 kB)
[██████████| 43 kB 1.1 MB/s]
Collecting pystd>=0.14.4
  Downloading pystd-0.15.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.4 MB)
[██████████| 2.4 MB 4.3 MB/s]
Requirement already satisfied: importlib-metadata in /opt/conda/lib/python3.7/site-packages (from py7zr) (4.11.3)
Collecting brotli>=1.0.9
  Downloading Brotli-1.0.9-cp37-cp37m-manylinux1_x86_64.whl (357 kB)
[██████████| 357 kB 49.3 MB/s]
```

```
Collecting multivolumefile>=0.2.3
  Downloading multivolumefile-0.2.3-py3-none-any.whl (17 kB)
Requirement already satisfied: texttable in /opt/conda/lib/python3.7/site-packages (from py7zr) (1.6.4)
Collecting pyppmd<0.19.0,>=0.18.1
  Downloading pyppmd-0.18.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (131 kB)
    ██████████| 131 kB 51.8 MB/s
Collecting pycryptodomex>=3.6.6
  Downloading pycryptodomex-3.14.1-cp35-abi3-manylinux2010_x86_64.whl (2.0 MB)
    ██████████| 2.0 MB 49.5 MB/s
Collecting pybcj>=0.5.0
  Downloading pybcj-0.5.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (57 kB)
    ██████████| 57 kB 3.2 MB/s
Requirement already satisfied: typing-extensions>=3.6.4 in /opt/conda/lib/python3.7/site-packages (from importlib-metadata->py7zr) (4.1.1)
Requirement already satisfied: zip>=0.5 in /opt/conda/lib/python3.7/site-packages (from importlib-metadata->py7zr) (3.6.0)
Installing collected packages: zipfile-deflate64, pyzstd, pyppmd, pycryptodomex, pybcj, multivolumefile, brotli, py7zr
Successfully installed brotli-1.0.9 multivolumefile-0.2.3 py7zr-0.18.3 pybcj-0.5.1 pycryptodomex-3.14.1 pyppmd-0.18.1 pyzstd-0.15.2 zipfile-deflate64-0.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

```
In [3]:  
from py7zr import unpack_7zarchive  
import shutil  
  
shutil.register_unpack_format('7zip',[ '.7z'],unpack_7zarchive)
```

```
In [4]:  
shutil.unpack_archive('../input/cifar-10/train.7z', '/kaggle/temp/')
```

```
In [5]:  
train_labels = pd.read_csv("../input/cifar-10/trainLabels.csv", header="infer")  
  
classes = train_labels['label'].unique()  
print(classes)  
  
['frog' 'truck' 'deer' 'automobile' 'bird' 'horse' 'ship' 'cat' 'dog'  
'airplane']
```

```
In [6]:  
if not os.path.exists("/kaggle/temp/valid"):  
    os.mkdir("/kaggle/temp/valid")  
  
parent_path_train = "/kaggle/temp/train"  
parent_path_valid = "/kaggle/temp/valid"  
parent_path_test = "/kaggle/temp/test"  
  
for class1 in classes:  
    path_train = os.path.join(parent_path_train, class1)  
    if not os.path.exists(path_train):  
        os.mkdir(path_train)  
    path_valid = os.path.join(parent_path_valid, class1)  
    if not os.path.exists(path_valid):  
        os.mkdir(path_valid)  
  
    for (int_ind, row) in train_labels.iterrows():  
        id = str(row["id"])+".png"  
        source_path = os.path.join(parent_path_train, id)  
  
        p=np.random.random()  
        if p<=0.8:  
            target_path = os.path.join(parent_path_train, row["label"], id)  
            os.replace(source_path, target_path)  
        else:  
            target_path = os.path.join(parent_path_valid, row["label"], id)  
            os.replace(source_path, target_path)
```

```
In [7]:  
!ls /kaggle/temp/valid  
!ls /kaggle/temp/train
```

```
airplane automobile bird cat deer dog frog horse ship truck
airplane automobile bird cat deer dog frog horse ship truck
```

```
In [8]: model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False, input_shape=(32,32, 3)))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Conv2D(filters=48, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Conv2D(filters=64, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Conv2D(filters=80, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Conv2D(filters=96, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Conv2D(filters=128, kernel_size=(3,3), strides=(1,1), padding='valid', activation=None, use_bias=False))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Flatten())
model.add(Dense(units=10))
model.add(BatchNormalization())
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
2022-03-28 10:59:10.843937: I tensorflow/core/common_runtime/process_util.cc:146] Creating new thread pool with default inter op setting: 2. Tune using inter_op_parallelism_threads for best performance.
```

```
In [9]: train_datagen = ImageDataGenerator(featurewise_center=False,
                                         samplewise_center=False,
                                         featurewise_std_normalization=False,
                                         samplewise_std_normalization=False,
                                         zca_whitening=False,
                                         rotation_range=10,
                                         zoom_range=0.1,
                                         width_shift_range=0.1,
                                         height_shift_range=0.1,
                                         horizontal_flip=False,
                                         vertical_flip=False,
                                         rescale=1./255)
valid_datagen = ImageDataGenerator()

train_generator = train_datagen.flow_from_directory(directory='/kaggle/temp/train/', shuffle=True, target_size=(32,32), batch_size=128)
valid_generator = valid_datagen.flow_from_directory(directory='/kaggle/temp/valid/', shuffle=True, target_size=(32,32), batch_size=128)
```

```
Found 40060 images belonging to 10 classes.
Found 9940 images belonging to 10 classes.
```

```
In [10]: model.fit(train_generator, epochs=10, validation_data=valid_generator, steps_per_epoch=train_generator.n//train_generator.batch_size,
               validation_steps=valid_generator.n//valid_generator.batch_size, workers=8, use_multiprocessing=True)
```

```
2022-03-28 10:59:13 915783: T tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:1851 None of the MLIR Optimization P
```

```
2022-08-20 10:57:51.073252: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that can speed up your training: AVX2 FMA
```

```
asses are enabled (registered 2)
```

```
Epoch 1/10
312/312 [=====] - 379s 1s/step - loss: 1.4938 - accuracy: 0.4866 - val_loss: 102.2503 - val_accuracy: 0.1355
Epoch 2/10
312/312 [=====] - 375s 1s/step - loss: 1.1637 - accuracy: 0.6184 - val_loss: 70.6234 - val_accuracy: 0.2143
Epoch 3/10
312/312 [=====] - 376s 1s/step - loss: 0.9989 - accuracy: 0.6757 - val_loss: 290.4420 - val_accuracy: 0.1089
Epoch 4/10
312/312 [=====] - 374s 1s/step - loss: 0.9032 - accuracy: 0.7070 - val_loss: 137.5347 - val_accuracy: 0.1854
Epoch 5/10
312/312 [=====] - 375s 1s/step - loss: 0.8236 - accuracy: 0.7303 - val_loss: 283.7267 - val_accuracy: 0.1894
Epoch 6/10
312/312 [=====] - 384s 1s/step - loss: 0.7654 - accuracy: 0.7476 - val_loss: 188.3864 - val_accuracy: 0.1764
Epoch 7/10
312/312 [=====] - 380s 1s/step - loss: 0.7179 - accuracy: 0.7622 - val_loss: 346.6114 - val_accuracy: 0.1857
Epoch 8/10
312/312 [=====] - 379s 1s/step - loss: 0.6768 - accuracy: 0.7792 - val_loss: 419.1365 - val_accuracy: 0.1233
Epoch 9/10
312/312 [=====] - 382s 1s/step - loss: 0.6442 - accuracy: 0.7873 - val_loss: 295.7296 - val_accuracy: 0.1655
Epoch 10/10
312/312 [=====] - 381s 1s/step - loss: 0.6043 - accuracy: 0.8013 - val_loss: 357.6776 - val_accuracy: 0.1085
```

```
Out[10]: <keras.callbacks.History at 0x7f9de81c5290>
```

```
In [11]: shutil.unpack_archive('/kaggle/input/cifar-10/test.7z','/kaggle/temp/test')
```

```
In [12]: shutil.unregister_unpack_format('7zip')
```

```
In [13]: test_datagen = ImageDataGenerator(rescale=1./255)

test_gen = test_datagen.flow_from_directory(directory='/kaggle/temp/test',target_size=(32,32),batch_size=64,class_mode=None,shuffle=False)
```

```
Found 300000 images belonging to 1 classes.
```

```
In [14]: test_gen.reset()
predictions_vecs = model.predict(test_gen)

predictions_final = np.argmax(predictions_vecs, axis=1)
```

```
In [15]: print(type(train_generator.class_indices))
print(train_generator.class_indices)

classes = {value:key for (key,value) in train_generator.class_indices.items()}
print(classes)

predicted_classes=np.empty(shape=300000,dtype=np.dtype('U20'))

ind=0
for i in predictions_final.tolist():
    predicted_classes[ind]=classes[i]
    ind=ind+1
```

```

filenames_wo_ext = []
for fname in test_gen.filenames:
    filenames_wo_ext.append(int(fname.split(sep="/")[1].split(sep=".")[0])-1)

predicted_classes_final = np.empty(shape=300000,dtype=np.dtype('U20'))
predicted_classes_final[filenames_wo_ext]=predicted_classes

<class 'dict'>
{'airplane': 0, 'automobile': 1, 'bird': 2, 'cat': 3, 'deer': 4, 'dog': 5, 'frog': 6, 'horse': 7, 'ship': 8, 'truck': 9}
{0: 'airplane', 1: 'automobile', 2: 'bird', 3: 'cat', 4: 'deer', 5: 'dog', 6: 'frog', 7: 'horse', 8: 'ship', 9: 'truck'}

```

In [16]:

```

sub = pd.read_csv('../input/cifar-10/sampleSubmission.csv',header='infer')
sub.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300000 entries, 0 to 299999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
 ---  --     --           --    
 0   id      300000 non-null  int64  
 1   label   300000 non-null  object 
dtypes: int64(1), object(1)
memory usage: 4.6+ MB

```

In [17]:

```

sub['label'] = predicted_classes_final
sub.to_csv('submission.csv',index=False)

```

In []:

Continue exploring



Data
1 input and 1 output



Logs
5990.3 second run - successful



Comments
0 comments

