Assignment 3

DL Assignment 3 [CNN] Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (1h:4m)

[4]:

```
import tensorflow as tf

import tensorflow_datasets as tfds
```

[5]:

```
from keras.layers.core import Dense, Activation, Dropout, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from tensorflow.keras.optimizers import SGD, RMSprop, Adam
from keras.utils import np_utils
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn import metrics
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import numpy as np
import os
import cv2
import random
from numpy import *
from PIL import Image
import theano
```

Data

Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

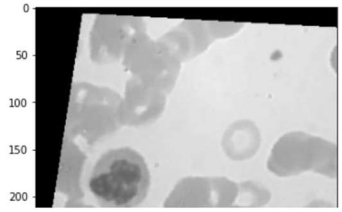
Settings

[6]:

```
path_test = "/kaggle/input/blood-cells/dataset2-master/dataset2-master/images/TRAIN"
```

[7]:

```
CATEGORIES = ["EOSINOPHIL", "LYMPHOCYTE", "MONOCYTE", "NEUTROPHIL"]
for category in CATEGORIES:
    path = os.path.join(path_test, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img), cv2.IMREAD_GRAYSCALE)
        plt.imshow(img_array, cmap="gray")
        plt.show()
        break
    break
```



Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

Settings

[8]:
```
print(img_array.shape)
IMG_SIZE =200
new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
print(new_array.shape)
```

```
(240, 320)
(200, 200)
```

[9]:
```
training = []
def createTrainingData():
    for category in CATEGORIES:
        path = os.path.join(path_test, category)
        class_num = CATEGORIES.index(category)
        for img in os.listdir(path):
          img_array = cv2.imread(os.path.join(path,img))
          new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
          training.append([new_array, class_num])
createTrainingData()
```

+ Code + Markdown

[10]:

[10]:
```
random.shuffle(training)
```

[11]:
```
X =[]
y =[]
for features, label in training:
    X.append(features)
    y.append(label)
X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3)
y = np.array(y)
```

+ Code + Markdown

[12]:
```
X = X.astype('float32')
X /= 255
from keras.utils import np_utils
Y = np_utils.to_categorical(y, 4)
print(Y[100])
print(shape(Y))
```

```
[0. 0. 1. 0.]
(9957, 4)
```

+ Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

Settings

+ Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

Settings

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_sta
```

[15]:

```
batch_size = 16
nb_classes =4
nb_epochs = 10
img_rows, img_columns = 200, 200
img_channel = 3
nb_filters = 32
nb_pool = 2
nb_conv = 3
```

+ Code + Markdown

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), padding='same', activation=tf.nn.relu,
                           input_shape=(200, 200, 3)),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Conv2D(32, (3,3), padding='same', activation=tf.nn.relu),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(4,  activation=tf.nn.softmax)
])
```

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accur
```

[18]:

```
model.fit(X_train, y_train, batch_size = batch_size, epochs = nb_epochs, verbose = 1,
```

```
2022-11-23 22:38:47.182055: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of th
e MLIR Optimization Passes are enabled (registered 2)
Epoch 1/10
498/498 [==============================] - 114s 228ms/step - loss: 1.4610 - accuracy: 0.2503 - val_los
s: 1.3867 - val_accuracy: 0.2515
Epoch 2/10
498/498 [==============================] - 113s 227ms/step - loss: 1.3861 - accuracy: 0.2562 - val_los
s: 1.3874 - val_accuracy: 0.2390
Epoch 3/10
498/498 [==============================] - 113s 226ms/step - loss: 1.3828 - accuracy: 0.2850 - val_los
s: 1.3879 - val_accuracy: 0.2415
Epoch 4/10
498/498 [==============================] - 115s 230ms/step - loss: 1.3626 - accuracy: 0.3141 - val_los
s: 1.3978 - val_accuracy: 0.2610
Epoch 5/10
498/498 [==============================] - 111s 223ms/step - loss: 1.3238 - accuracy: 0.3685 - val_los
s: 1.4298 - val_accuracy: 0.2570
Epoch 6/10
498/498 [==============================] - 111s 224ms/step - loss: 1.2317 - accuracy: 0.4350 - val_los
s: 1.5232 - val_accuracy: 0.2435
Epoch 7/10
498/498 [==============================] - 111s 223ms/step - loss: 1.0808 - accuracy: 0.5391 - val_los
s: 1.5800 - val_accuracy: 0.2525
Epoch 8/10
498/498 [==============================] - 111s 223ms/step - loss: 0.8535 - accuracy: 0.6525 - val_los
s: 1.7597 - val_accuracy: 0.2575
```

+ Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

Settings

+ Add Data

Input

- blood-cells
  - dataset-master
  - dataset2-master
    - dataset2-master
      - images
        - TEST
        - TEST_SIMPLE
        - TRAIN
          - EOSINOPHIL
          - LYMPHOCYTE
          - MONOCYTE
          - NEUTROPHIL
        - labels.csv

Output (60KB / 19.5GB)

- /kaggle/working

Settings

```
498/498 [==============================] - 113s 226ms/step - loss: 1.3828 - accuracy: 0.2850 - val_los
s: 1.3879 - val_accuracy: 0.2415
Epoch 4/10
498/498 [==============================] - 115s 230ms/step - loss: 1.3626 - accuracy: 0.3141 - val_los
s: 1.3978 - val_accuracy: 0.2610
Epoch 5/10
498/498 [==============================] - 111s 223ms/step - loss: 1.3238 - accuracy: 0.3685 - val_los
s: 1.4298 - val_accuracy: 0.2570
Epoch 6/10
498/498 [==============================] - 111s 224ms/step - loss: 1.2317 - accuracy: 0.4350 - val_los
s: 1.5232 - val_accuracy: 0.2435
Epoch 7/10
498/498 [==============================] - 111s 223ms/step - loss: 1.0808 - accuracy: 0.5391 - val_los
s: 1.5800 - val_accuracy: 0.2525
Epoch 8/10
498/498 [==============================] - 111s 223ms/step - loss: 0.8535 - accuracy: 0.6525 - val_los
s: 1.7597 - val_accuracy: 0.2575
Epoch 9/10
498/498 [==============================] - 110s 221ms/step - loss: 0.6308 - accuracy: 0.7498 - val_los
s: 2.0701 - val_accuracy: 0.2505
Epoch 10/10
498/498 [==============================] - 110s 221ms/step - loss: 0.4455 - accuracy: 0.8303 - val_los
s: 2.4517 - val_accuracy: 0.2671
```

[18]: `<keras.callbacks.History at 0x7f303ce23d10>`

+ Code + Markdown

[19]:

```
score = model.evaluate(X_test, y_test, verbose = 0 )
print("Test Score: ", score[0])
print("Test accuracy: ", score[1])
```

```
Test Score:  2.451676607131958
Test accuracy:  0.2670682668685913
```