

60k-image-classification

February 25, 2024

1 Image Classification using ANN

In this project, we explore the task of image classification using Artificial Neural Networks (ANNs). ANNs are a type of neural network where neurons in one layer are connected to all neurons in the next layer

```
[2]: import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
%matplotlib inline
```

2 The CIFAR-10 dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

```
[3]: (x_train , y_train), (x_test , y_test)=tf.keras.datasets.cifar10.load_data()
```

3 Scalling images

```
[6]: x_train_scaled=x_train/255
x_test_scaled=x_test/255
```

```
[7]: y_train[0:5]
```

```
[7]: array([[6],
          [9],
          [9],
          [4],
```

```
[1]], dtype=uint8)
```

4 one hot encoding

changing the categorical values into numerical values

```
[9]: y_train_categorical= keras.utils.to_categorical(
      y_train, num_classes=10,dtype='float32'
    )
      y_train_categorical[0:5]
```

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

```
[10]: y_test_categorical= keras.utils.to_categorical(
      y_test, num_classes=10,dtype='float32'
    )
      y_test_categorical[0:5]
```

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

```
[ ]: class_labels=['truck','ship','horse','frog','dog','deer','cat','bird','automobile','airplane']
```

Artificial Neural Network Model with 2 hidden layers

```
[ ]: model=keras.Sequential([
      keras.layers.Flatten(input_shape=(32,32,3)),
      keras.layers.Dense(3000 , activation='relu'),
      keras.layers.Dense(1500 , activation='relu'),
      keras.layers.Dense(10 , activation='sigmoid')
    ])

      model.compile(
        optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy']
      )
      model.fit(x_train_scaled , y_train_categorical , epochs=10
    )
```

```
[ ]: np.argmax(model.predict(x_test_scaled)[1])
```

```
[ ]: class_labels(y_test[1][0])
```

```
[ ]:
```