

Experiment 9 - Image Classification - Alexnet on CIFAR-10 Dataset

Problem Statement:

To train a CNN model to classify images from the CIFAR-10 database.

GitHub & Google Colab Links:

GitHub Link: <https://github.com/piyush-gambhir/ncu-lab-manual-and-end-semester-projects/blob/main/NCU-CSL312%20-%20DL%20-%20Lab%20Manual/Experiment%209/Experiment%209.ipynb>

Google Colab Link:



Installing Dependencies:

```
In [ ]: ! pip install tabulate numpy pandas matplotlib seaborn
```

```
Requirement already satisfied: tabulate in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (0.9.0)
Requirement already satisfied: numpy in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (1.26.4)
Requirement already satisfied: pandas in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (2.2.2)
Requirement already satisfied: matplotlib in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (3.8.4)
Requirement already satisfied: seaborn in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

Code

```
In [ ]: import keras
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D
from keras.layers import BatchNormalization

from keras.regularizers import l2
```

```
In [ ]: # Constants
NUM_CLASSES = 10
BATCH_SIZE = 32
EPOCHS = 1
L2_REG_RATE = 0.01

def load_and_preprocess_data():
```

```

# Loads the CIFAR10 dataset
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# One hot encode outputs
y_train = keras.utils.to_categorical(y_train, NUM_CLASSES)
y_test = keras.utils.to_categorical(y_test, NUM_CLASSES)

print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

return x_train, y_train, x_test, y_test

def build_alexnet(input_shape):
    # Initialize model
    model = Sequential()

    # 1st Conv Layer
    model.add(Conv2D(96, (11, 11), input_shape=input_shape,
                    padding='same', kernel_regularizer=l2(L2_REG_RATE)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # 2nd through 5th Conv Layers
    layer_configs = [(256, 5, 2), (512, 3, 2), (1024, 3, 0), (1024, 3, 2)]
    for filters, kernel_size, padding in layer_configs:
        if padding:
            model.add(ZeroPadding2D((1, 1)))
            model.add(Conv2D(filters, (kernel_size, kernel_size), padding='same'))
            model.add(BatchNormalization())
            model.add(Activation('relu'))
        if padding:
            model.add(MaxPooling2D(pool_size=(2, 2)))

    # Fully Connected Layers
    model.add(Flatten())
    model.add(Dense(3072))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.5))

    model.add(Dense(4096))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.5))

    model.add(Dense(NUM_CLASSES))
    model.add(BatchNormalization())
    model.add(Activation('softmax'))

    return model

def main():
    x_train, y_train, x_test, y_test = load_and_preprocess_data()
    model = build_alexnet(x_train.shape[1:])
    model.compile(loss='categorical_crossentropy',
                  optimizer=keras.optimizers.Adadelta(), metrics=['accuracy'])

    # Train the model
    model.fit(x_train, y_train, batch_size=BATCH_SIZE, epochs=EPOCHS,
              validation_data=(x_test, y_test), shuffle=True)

    # Save the model
    model.save("Trained_Models/CIFAR10_AlexNet_1_Epoch.h5")

    # Evaluate the model
    scores = model.evaluate(x_test, y_test, verbose=1)
    print('Test loss:', scores[0])
    print('Test accuracy:', scores[1])

if __name__ == '__main__':
    main()

```

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>

170498071/170498071 ————— 18s 0us/step

x_train shape: (50000, 32, 32, 3)

50000 train samples

10000 test samples

```
c:\Users\mainp\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\layers\convolutional\base_con  
v.py:99: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models  
, prefer using an `Input(shape)` object as the first layer in the model instead.  
    super().__init__(
```

103/1563 — **1:01:58** 3s/step - accuracy: 0.1258 - loss: 2.7051

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js