

NNDL ASSIGNMENT - 4

VYSHNAVI NAGALLA - 700759215

Video Link:

https://drive.google.com/file/d/15kGMCtZRR02KeOss3c8PQIdS-e_Cxeas/view?usp=sharing

```
!pip install np_utils

collecting np_utils
Downloading np_utils-0.6.0.tar.gz (61 kB)
62.0/62.0 kB 823.3 kB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.0 in /usr/local/lib/python3.10/dist-packages (from np_utils) (1.25.2)
Building wheels for collected packages: np_utils
Building wheel for np_utils (setup.py) ... done
Created wheel for np_utils: filename=np_utils-0.6.0-py3-none-any.whl size=56441 sha256=9353583b53b4030bc2c95f08a32c31a5199592fd9d23240ec5fac009e9c846553
Stored in directory: /root/.cache/pip/wheels/b6/c7/50/2307607f44366dd021209f660045f8d51cb976514d30be7cc7
Successfully built np_utils
Installing collected packages: np_utils
Successfully installed np_utils-0.6.0

# Simple CNN model for CIFAR-10
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.utils import to_categorical
#from keras import backend as K
#K.set_image_dim_ordering('th')

# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()

# normalize inputs from 0-255 to 0.0-1.0
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train = X_train / 255.0
X_test = X_test / 255.0
# one hot encode outputs
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
num_classes = y_test.shape[1]

# Create the model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), padding='same', activation='relu'))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

Executing (5m 4s) <cell line: 9> > error_handler() > fit() > error_handler() > __call__() > _call() > call_function() > _call_flat() > call_pre flattened() > call_flat() > call_function() > quick_execute()

✓ RAM ▼ | ✦ Gemini

✓ RAM Disk Gemini

RAM Gemini

