In [1]:
```
# Plot ad hoc CIFAR10 instances
from tensorflow.keras.datasets import cifar10
from matplotlib import pyplot
```

In [2]:
```
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
 (https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz)
170498071/170498071 [==============================] - 70s 0us/step
```

In [3]:
```
# Simple CNN model for the CIFAR-10 Dataset
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.constraints import maxnorm
from keras.optimizers import SGD
from keras.layers.convolutional import Convolution2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
#K.set_image_dim_ordering( 'th' )
```

In [4]:
```
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
```

In [5]:
```
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

In [7]:
```
# 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
```

In [8]:
```
# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]
```

In [9]:
```
# Create the model
model = Sequential()
model.add(Convolution2D(32,(3, 3), input_shape=(32, 32, 3), activation= 'relu'))
model.add(Dropout(0.2))
model.add(Convolution2D(32, 3, 3, activation= 'relu',))
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' ))
```

In [10]:
```
# Compile model
epochs = 25
lrate = 0.01
decay = lrate/epochs
sgd = SGD(lr=lrate, momentum=0.9, decay=decay, nesterov=False)
model.compile(loss= 'categorical_crossentropy' , optimizer=sgd, metrics=[ 'accura
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 dropout (Dropout)           (None, 30, 30, 32)        0

 conv2d_1 (Conv2D)           (None, 10, 10, 32)        9248

 max_pooling2d (MaxPooling2D  (None, 5, 5, 32)         0
 )

 flatten (Flatten)           (None, 800)               0

 dense (Dense)               (None, 512)               410112

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 425,386
Trainable params: 425,386
Non-trainable params: 0
_________________________________________________________________
None

D:\anaconda3\lib\site-packages\keras\optimizers\optimizer_v2\gradient_descent.p
y:111: UserWarning: The `lr` argument is deprecated, use `learning_rate` instea
d.
  super().__init__(name, **kwargs)
```

In [11]:
```
# Fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs,batch
          verbose=1)
```

```
Epoch 1/25
1563/1563 [==============================] - 73s 45ms/step - loss: 1.7763 - acc
uracy: 0.3566 - val_loss: 1.4066 - val_accuracy: 0.5038
Epoch 2/25
1563/1563 [==============================] - 69s 44ms/step - loss: 1.4216 - acc
uracy: 0.4882 - val_loss: 1.2681 - val_accuracy: 0.5529
Epoch 3/25
1563/1563 [==============================] - 70s 45ms/step - loss: 1.3068 - acc
uracy: 0.5302 - val_loss: 1.1992 - val_accuracy: 0.5732
Epoch 4/25
1563/1563 [==============================] - 70s 45ms/step - loss: 1.2297 - acc
uracy: 0.5622 - val_loss: 1.1907 - val_accuracy: 0.5831
Epoch 5/25
1563/1563 [==============================] - 70s 45ms/step - loss: 1.1719 - acc
uracy: 0.5830 - val_loss: 1.1069 - val_accuracy: 0.6130
Epoch 6/25
1563/1563 [==============================] - 70s 45ms/step - loss: 1.1261 - acc
uracy: 0.5997 - val_loss: 1.0979 - val_accuracy: 0.6091
Epoch 7/25
1563/1563 [==============================] - 69s 44ms/step - loss: 1.0894 - acc
uracy: 0.6151 - val_loss: 1.0565 - val_accuracy: 0.6286
Epoch 8/25
1563/1563 [==============================] - 69s 44ms/step - loss: 1.0559 - acc
uracy: 0.6228 - val_loss: 1.0409 - val_accuracy: 0.6299
Epoch 9/25
1563/1563 [==============================] - 70s 45ms/step - loss: 1.0313 - acc
uracy: 0.6340 - val_loss: 1.0329 - val_accuracy: 0.6365
Epoch 10/25
1563/1563 [==============================] - 71s 45ms/step - loss: 1.0041 - acc
uracy: 0.6441 - val_loss: 1.0177 - val_accuracy: 0.6391
Epoch 11/25
1563/1563 [==============================] - 71s 45ms/step - loss: 0.9793 - acc
uracy: 0.6532 - val_loss: 1.0007 - val_accuracy: 0.6476
Epoch 12/25
1563/1563 [==============================] - 71s 45ms/step - loss: 0.9591 - acc
uracy: 0.6608 - val_loss: 0.9891 - val_accuracy: 0.6517
Epoch 13/25
1563/1563 [==============================] - 71s 46ms/step - loss: 0.9362 - acc
uracy: 0.6684 - val_loss: 0.9896 - val_accuracy: 0.6528
Epoch 14/25
1563/1563 [==============================] - 71s 45ms/step - loss: 0.9219 - acc
uracy: 0.6727 - val_loss: 0.9775 - val_accuracy: 0.6575
Epoch 15/25
1563/1563 [==============================] - 71s 45ms/step - loss: 0.9049 - acc
uracy: 0.6783 - val_loss: 0.9692 - val_accuracy: 0.6612
Epoch 16/25
1563/1563 [==============================] - 79s 50ms/step - loss: 0.8927 - acc
uracy: 0.6826 - val_loss: 0.9627 - val_accuracy: 0.6623
Epoch 17/25
1563/1563 [==============================] - 69s 44ms/step - loss: 0.8749 - acc
uracy: 0.6892 - val_loss: 0.9549 - val_accuracy: 0.6640
Epoch 18/25
```

```
1563/1563 [==============================] - 75s 48ms/step - loss: 0.8621 - acc
uracy: 0.6953 - val_loss: 0.9561 - val_accuracy: 0.6650
Epoch 19/25
1563/1563 [==============================] - 75s 48ms/step - loss: 0.8519 - acc
uracy: 0.6986 - val_loss: 0.9448 - val_accuracy: 0.6665
Epoch 20/25
1563/1563 [==============================] - 71s 45ms/step - loss: 0.8386 - acc
uracy: 0.7039 - val_loss: 0.9443 - val_accuracy: 0.6723
Epoch 21/25
1563/1563 [==============================] - 70s 45ms/step - loss: 0.8284 - acc
uracy: 0.7063 - val_loss: 0.9432 - val_accuracy: 0.6680
Epoch 22/25
1563/1563 [==============================] - 70s 45ms/step - loss: 0.8160 - acc
uracy: 0.7113 - val_loss: 0.9334 - val_accuracy: 0.6745
Epoch 23/25
1563/1563 [==============================] - 70s 45ms/step - loss: 0.8060 - acc
uracy: 0.7151 - val_loss: 0.9375 - val_accuracy: 0.6745
Epoch 24/25
1563/1563 [==============================] - 69s 44ms/step - loss: 0.7947 - acc
uracy: 0.7169 - val_loss: 0.9354 - val_accuracy: 0.6742
Epoch 25/25
1563/1563 [==============================] - 68s 44ms/step - loss: 0.7886 - acc
uracy: 0.7193 - val_loss: 0.9308 - val_accuracy: 0.6752
```

Out[11]: <keras.callbacks.History at 0x10ea76ac9a0>

In [12]:
```
# Final evaluation of the model
scores = model.evaluate(X_test, y_test)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
313/313 [==============================] - 3s 8ms/step - loss: 0.9308 - accurac
y: 0.6752
Accuracy: 67.52%
```

In [ ]: