

Name: Sindhuja Kasagoni

```
jupyter ICP7 (unsaved changes) Python 3 (ipykernel) Logout
```

```
File Edit View Insert Cell Kernel Widgets Help Not Trusted
```

```
In [5]: import numpy as np
        from keras.datasets import cifar10
        from keras.models import Sequential
        from keras.layers import Dense, Dropout, Flatten
        from keras.constraints import maxnorm
        from keras.optimizers import SGD
        from keras.layers.convolutional import Conv2D, MaxPooling2D
        from keras.utils import np_utils

In [6]: np.random.seed(7)

In [7]: (X_train, y_train), (X_test, y_test) = cifar10.load_data()

In [8]: X_train = X_train.astype('float32') / 255.0
        X_test = X_test.astype('float32') / 255.0

In [9]: y_train = np_utils.to_categorical(y_train)
        y_test = np_utils.to_categorical(y_test)
        num_classes = y_test.shape[1]

In [10]: model = Sequential()
         model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), padding='same', activation='relu', kernel_constraint=maxnorm(3)))
         model.add(Dropout(0.2))
         model.add(Conv2D(32, (3, 3), activation='relu', padding='same', kernel_constraint=maxnorm(3)))
         model.add(MaxPooling2D(pool_size=(2, 2)))
         model.add(Conv2D(32, (3, 3), activation='relu', padding='same', kernel_constraint=maxnorm(3)))
         model.add(Dropout(0.2))
         model.add(Flatten())
         model.add(Dense(1000, activation='relu'))
         model.add(Dropout(0.5))
         model.add(Dense(num_classes, activation='softmax'))
         model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])

In [11]: sgd = SGD(learning_rate=0.01, momentum=0.9, decay=1e-6)
         model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
         print(model.summary())

Model: "sequential_1"
-----
Layer (type)                 Output Shape              Param #
-----
conv2d_2 (Conv2D)            (None, 32, 32, 32)        896
dropout_2 (Dropout)          (None, 32, 32, 32)         0
conv2d_3 (Conv2D)            (None, 32, 32, 32)       9248
max_pooling2d_1 (MaxPooling (None, 16, 16, 32)         0
2D)
flatten_1 (Flatten)          (None, 8192)               0
dense_2 (Dense)              (None, 512)              4194816
dropout_3 (Dropout)          (None, 512)               0
dense_3 (Dense)              (None, 10)                5130
-----
Total params: 4,210,090
Trainable params: 4,210,090

In [13]: scores = model.evaluate(X_test, y_test, verbose=0)
         print("Accuracy: %.2f%%" % (scores[1]*100))

Accuracy: 65.50%

In [14]: import numpy as np
         from keras.datasets import cifar10
         from keras.models import Sequential
         from keras.layers import Dense, Dropout, Flatten
         from keras.layers.convolutional import Conv2D, MaxPooling2D
         from keras.constraints import maxnorm
         from keras.utils import np_utils
         from keras.optimizers import SGD

         # Fix random seed for reproducibility
         np.random.seed(7)

         # 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') / 255.0
         X_test = X_test.astype('float32') / 255.0

         # One hot encode outputs
         y_train = np_utils.to_categorical(y_train)
         y_test = np_utils.to_categorical(y_test)
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
=====		
conv2d_4 (Conv2D)	(None, 32, 32, 32)	896
dropout_4 (Dropout)	(None, 32, 32, 32)	0
conv2d_5 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_2 (MaxPooling 2D)	(None, 16, 16, 32)	0
conv2d_6 (Conv2D)	(None, 16, 16, 64)	18496
dropout_5 (Dropout)	(None, 16, 16, 64)	0
conv2d_7 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_3 (MaxPooling 2D)	(None, 8, 8, 64)	0
conv2d_8 (Conv2D)	(None, 8, 8, 128)	73856
dropout_6 (Dropout)	(None, 8, 8, 128)	0
conv2d_9 (Conv2D)	(None, 8, 8, 128)	147584

