



NAME: RADHIKA GUPTA

REG. NO: 22MCA1119

ITA-6016 Machine Learning

Digital Assignment –Lab-5

SUBMITTED TO: Dr_Dominic Savio M

CNN:

CODE OF THE PROGRAM AND OUTPUT:

```
In [2]: import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten
from keras.utils import np_utils
from sklearn.datasets import load_iris

# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target

# Encode the target labels
encoder = LabelEncoder()
encoder.fit(y)
encoded_y = encoder.transform(y)
dummy_y = np_utils.to_categorical(encoded_y)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, dummy_y, test_size=0.2, random_state=42)

# Reshape the input data to match the CNN input shape
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1, 1).astype('float32')
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1, 1).astype('float32')

# Create the CNN model
model = Sequential()
model.add(Conv2D(32, (1, 1), input_shape=(X_train.shape[1], 1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(1, 1)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(3, activation='softmax'))
```

```

# Create the CNN model
model = Sequential()
model.add(Conv2D(32, (1, 1), input_shape=(X_train.shape[1], 1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(1, 1)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(3, activation='softmax'))

# Compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=5)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Accuracy:", accuracy)

```

```

Epoch 1/10
24/24 [=====] - 1s 13ms/step - loss: 0.9523 - accuracy: 0.4917 - val_loss: 0.8213 - val_accuracy: 0.7000
Epoch 2/10
24/24 [=====] - 0s 5ms/step - loss: 0.7666 - accuracy: 0.7500 - val_loss: 0.7039 - val_accuracy: 0.8333
Epoch 3/10
24/24 [=====] - 0s 5ms/step - loss: 0.6376 - accuracy: 0.7667 - val_loss: 0.5679 - val_accuracy: 0.7000
Epoch 4/10
24/24 [=====] - 0s 4ms/step - loss: 0.5538 - accuracy: 0.7917 - val_loss: 0.4856 - val_accuracy: 0.7667
Epoch 5/10
24/24 [=====] - 0s 4ms/step - loss: 0.4700 - accuracy: 0.8750 - val_loss: 0.4360 - val_accuracy: 0.9667

```

```

Epoch 4/10
24/24 [=====] - 0s 4ms/step - loss: 0.5538 - accuracy: 0.7917 - val_loss: 0.4856 - val_accuracy: 0.7667
Epoch 5/10
24/24 [=====] - 0s 4ms/step - loss: 0.4700 - accuracy: 0.8750 - val_loss: 0.4360 - val_accuracy: 0.9667
Epoch 6/10
24/24 [=====] - 0s 4ms/step - loss: 0.4218 - accuracy: 0.9167 - val_loss: 0.3767 - val_accuracy: 0.9667
Epoch 7/10
24/24 [=====] - 0s 4ms/step - loss: 0.3686 - accuracy: 0.9583 - val_loss: 0.3549 - val_accuracy: 0.8000
Epoch 8/10
24/24 [=====] - 0s 5ms/step - loss: 0.3386 - accuracy: 0.8667 - val_loss: 0.3371 - val_accuracy: 0.9000
Epoch 9/10
24/24 [=====] - 0s 4ms/step - loss: 0.3232 - accuracy: 0.9083 - val_loss: 0.3035 - val_accuracy: 0.9000
Epoch 10/10
24/24 [=====] - 0s 5ms/step - loss: 0.3123 - accuracy: 0.8833 - val_loss: 0.2767 - val_accuracy: 0.9000
1/1 [=====] - 0s 255ms/step - loss: 0.2767 - accuracy: 0.9000
Accuracy: 0.8999999761581421

```

]:

