Praneesh R V
CB.SC.U4CYS23036

AI & NN Lab - 4 CNN

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D,
Dropout, BatchNormalization
from tensorflow.keras.datasets import mnist
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt

# Load and preprocess the data
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Data Augmentation
datagen = ImageDataGenerator(
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=False,
    fill_mode='nearest'
)
datagen.fit(x_train)

# Build a more complex model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    BatchNormalization(),
```

```
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

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

# Train the model with augmented data
epochs = 10 # Increased epochs for better training
history = model.fit(datagen.flow(x_train, y_train, batch_size=64),
                    epochs=epochs,
                    validation_data=(x_test, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Loss = {loss}")
print(f"Accuracy = {accuracy}")

# Plot accuracy and loss
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.title("Training vs Validation Accuracy")

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.title("Training vs Validation Loss")

plt.show()
```

Loss = 0.06209929659962654
Accuracy = 0.9805999994277954

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Epoch 1/10
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class
  self._warn_if_super_not_called()
938/938 ━━━━━━━━━━━━━━━━━━━━ 90s 94ms/step - accuracy: 0.7867 - loss: 0.6718 - val_accuracy: 0.9667 - val_loss: 0.1077
Epoch 2/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 87s 92ms/step - accuracy: 0.9470 - loss: 0.1777 - val_accuracy: 0.9713 - val_loss: 0.0887
Epoch 3/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 87s 92ms/step - accuracy: 0.9589 - loss: 0.1392 - val_accuracy: 0.9787 - val_loss: 0.0639
Epoch 4/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 87s 92ms/step - accuracy: 0.9636 - loss: 0.1206 - val_accuracy: 0.9794 - val_loss: 0.0660
Epoch 5/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9659 - loss: 0.1140 - val_accuracy: 0.9775 - val_loss: 0.0701
Epoch 6/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9701 - loss: 0.1012 - val_accuracy: 0.9832 - val_loss: 0.0519
Epoch 7/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9701 - loss: 0.0993 - val_accuracy: 0.9848 - val_loss: 0.0466
Epoch 8/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9743 - loss: 0.0866 - val_accuracy: 0.9845 - val_loss: 0.0493
Epoch 9/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9735 - loss: 0.0906 - val_accuracy: 0.9876 - val_loss: 0.0400
Epoch 10/10
938/938 ━━━━━━━━━━━━━━━━━━━━ 86s 92ms/step - accuracy: 0.9751 - loss: 0.0830 - val_accuracy: 0.9806 - val_loss: 0.0621
313/313 ━━━━━━━━━━━━━━━━━━━━ 3s 8ms/step - accuracy: 0.9769 - loss: 0.0714
Loss = 0.06209929659962654
Accuracy = 0.9805999994277954
```