Course: Applications of Artificial Intelligence and Machine Learning
Code: PROG74000
Professor/Instructor(s): Akrem El-ghazal

Assignment/Lab 5: Winter 2025

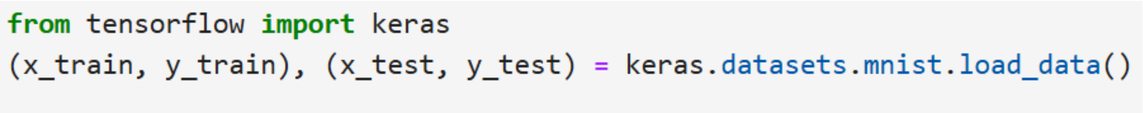
Convolutional Neural Networks (CNNs)

Objectives

- Understand Convolutional Neural Networks (CNNs) and demonstrate how they can be used for image classification.
- Implement CNNs using TensorFlow and Keras.
- Explore the importance of convolutional layers, pooling layers, and fully connected layers in building effective image classifiers.
- Compare the performance of Fully Connected Neural Networks (FNN) and Convolutional Neural Networks (CNN) for handwritten digit classification.
- Evaluate the performance of FNN and CNN models on noisy images.
- Gain hands-on experience with TensorFlow and Keras.

Dataset Description

- **MNIST dataset**: Contains 70,000 grayscale images of handwritten digits (0-9), each of size 28x28 pixels. 60,000 images are used for training, and 10,000 for testing. Below is a simple code to load the dataset:



```
from tensorflow import keras
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()
```

- **Add impulsive noise** (randomly occurring white and black pixels) to some of the test samples, and use these noisy images to test the performance of both the FNN and CNN models. You can write your own function to add impulsive noise to the images. Alternatively, a function has already been provided for you in the course shell.

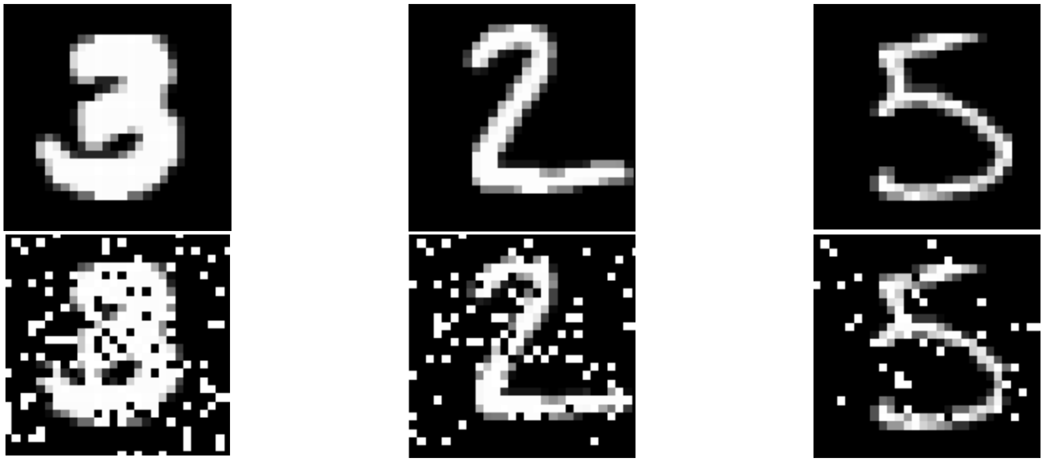


Figure 1. Some test images and their corresponding noisy versions

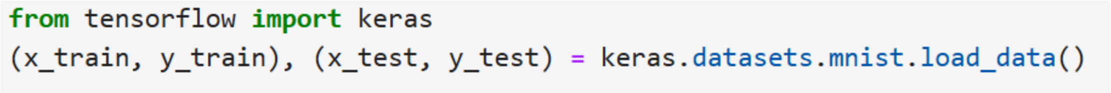
Instructions

Fully Connected Neural Networks (FNN)

Step 1: Dataset Exploration and Preprocessing (1 pts.)

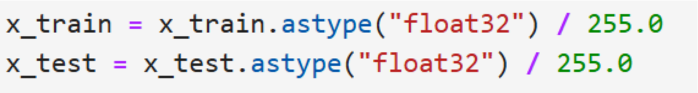
In this step, you will explore and preprocess the dataset using your machine learning and visualization skills. The tasks include, but are not limited to:

- Load the MNIST dataset using TensorFlow:



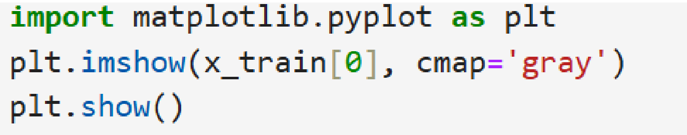
```
from tensorflow import keras
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()
```

- Normalize the pixel values of the images to a range of [0, 1]:



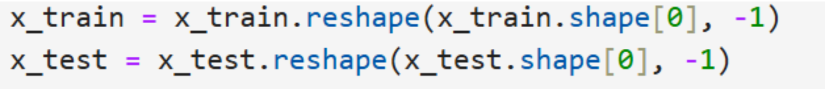
```
x_train = x_train.astype("float32") / 255.0
x_test = x_test.astype("float32") / 255.0
```

- Visualize a few sample images from the dataset using Matplotlib.



```
import matplotlib.pyplot as plt
plt.imshow(x_train[0], cmap='gray')
plt.show()
```

- Reshape the input images for the FNN. For the FNN, reshape the images to a 1D vector:



```
x_train = x_train.reshape(x_train.shape[0], -1)
x_test = x_test.reshape(x_test.shape[0], -1)
```

Step 2: Fully Connected Neural Network (FNN) Model (1 pts.)

- Implement an FNN with 2 hidden layers, using ReLU activations, followed by a softmax output layer.
- You are free to choose other hyperparameters, such as the number of units in each hidden layer and any other relevant parameters.

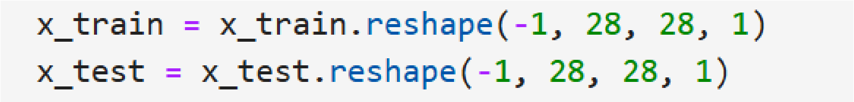
Step 3: Model Evaluation (1.5 pts.)

- Evaluate the FNN model on the test data and record the performance metrics.
- Add impulsive noise to a selected number of test images (choose a number and specify it in your report), then evaluate the FNN model on the noisy test data and record the performance metrics.

Convolutional Neural Network (CNN)

Step 4: Dataset Exploration and Preprocessing (1 pts.)

- Follow the same steps as for the FNN, but reshape the images differently for the CNN as follows:
-



```
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)
```

Step 5: Convolutional Neural Network (CNN) Model (1 pts.)

Implement a CNN with two convolutional layers, each followed by pooling layers. After that, flatten the output and add dense layers. You are free to select other hyperparameters, such as the size and number of filters, as well as any other relevant parameters.

Step 6: Model Evaluation (1.5 pts.)

- Evaluate the CNN model on the test data and record the performance metrics
- Use the same noisy images created in Step 3 to evaluate the CNN model on the noisy test data and record the performance metrics.

Step 7: Model Comparison (2 pts.)

- Based on your results, compare the FNN and CNN models, and write your final conclusion about the performance of both models.

The overall organization and Clarity of your solution (1 pts.)

- Ensure your solution is well-organized, clearly commented, and easy to follow
- In this lab the only limit is your imagination. Be creative \o/ (+2 pts.)

What to hand in?

Read the Assignment/Lab instructions section on the course shell.