University of Tripoli
Department of Electrical and Electronic Engineering
FALL 2025

EE569

Assignment 1
part c

نسيبة عمر بن زاهية

2200209233

Instructor: Dr. Nuri BenBarka

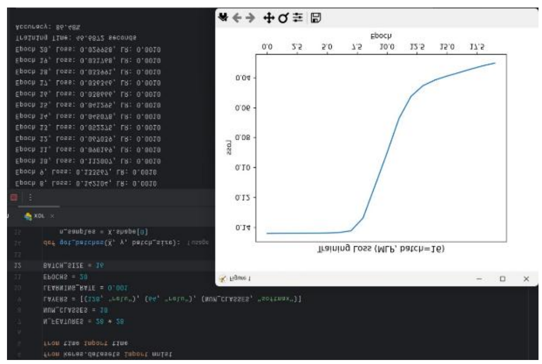
Task 1 – MLP on MNIST

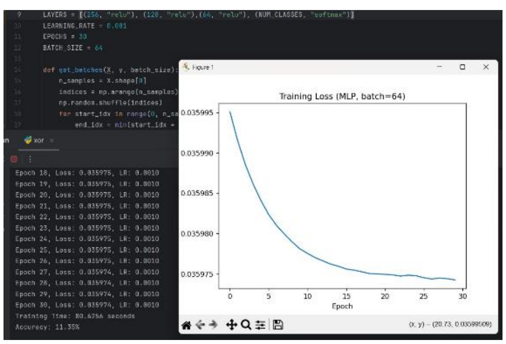
Several MLP configurations were trained on the full MNIST dataset. Different batch sizes, learning rates, activations, and depths were tested.

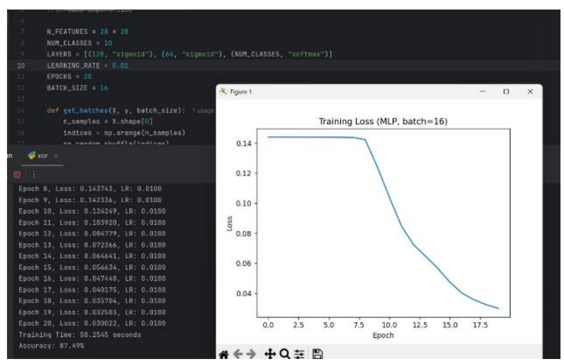
Key findings:

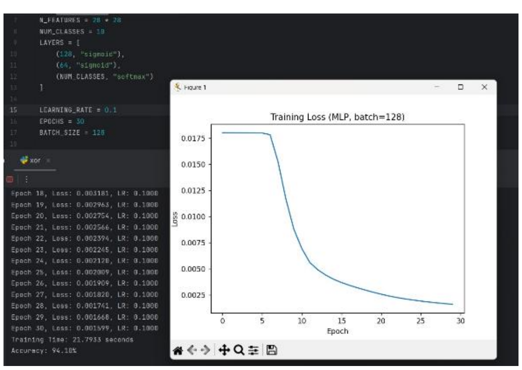
- **ReLU models converged faster and reached ~94–97% accuracy.**
- **Sigmoid models were slower but could reach ~85–95% with higher LR.**
- **Smaller batches improved convergence but increased training time.**

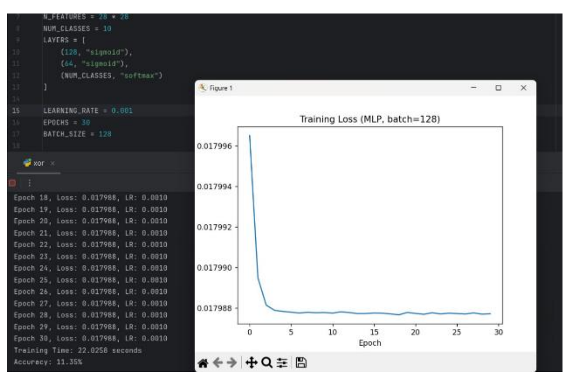
Figures:











Task 2 – Convolutions & Pooling

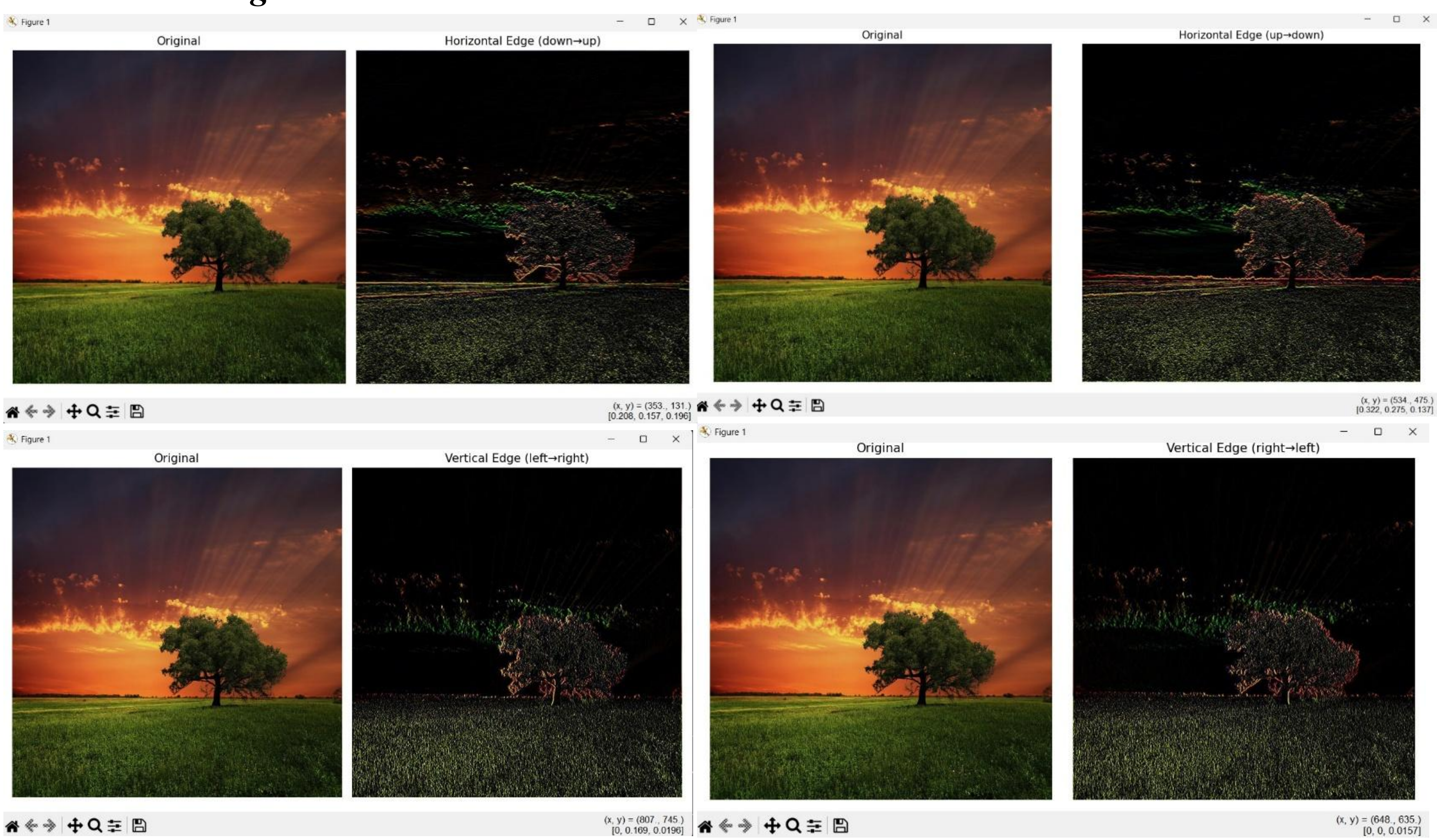
Custom convolution and max-pooling layers were implemented and validated using an RGB image.

Operations tested:

- **Horizontal & vertical edge detection**
- **Blur filters (3×3, 11×11)**
- **MaxPool (2×2, 8×8)**

Figures:

- ***Edge Detection Results***



- ***Blur Results***

- ***Pooling Results***

Task 3 – CNN + Gradient Check

A full CNN (Conv → Pool → Dense) was implemented.
Gradient checking was performed to validate backward pass.

Result:

- **Numerical difference was small → backprop implementation correct.**

Figure:

- ***Gradient Check Output***

```
D:\PythonProject5\.venv\Scripts\python.exe D:\PythonProject5\xor.py
Loss: 2.050595863523136
Performing gradient check...
Difference: 0.0026593150215638434
```

Task 4 – CNN Training on MNIST

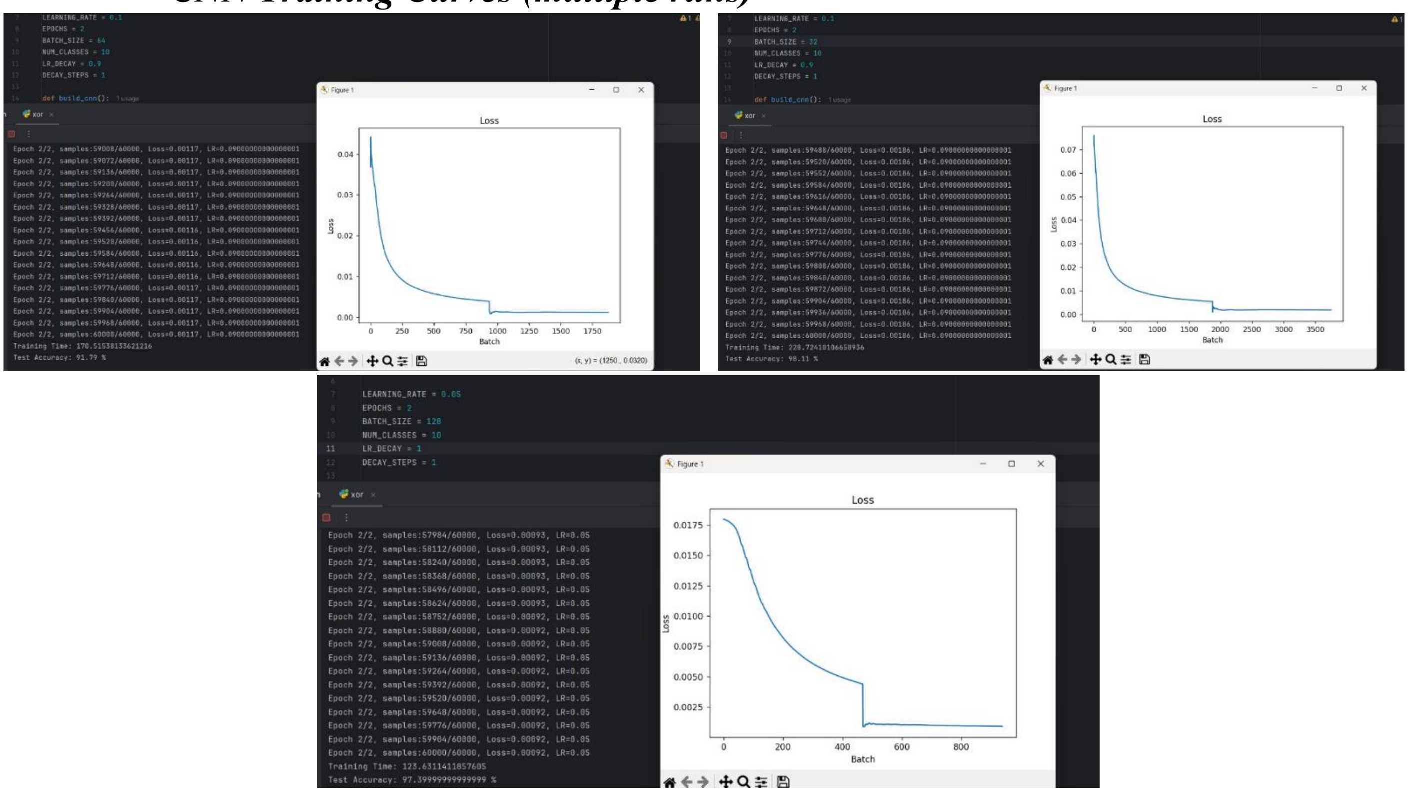
The custom CNN was trained end-to-end on MNIST using LR decay.

Key findings:

- **Accuracy ranged from 91% to 98% depending on batch size.**
- **LR decay and small batches improved performance.**

Figures:

- ***CNN Training Curves (multiple runs)***



Conclusion

- **MLP works well but is slower and less accurate.**
- **Convolution filters & pooling correctly modify the image.**
- **CNN implementation passed gradient check.**
- **CNN achieves the best performance (up to 98–99%).**
