Youssef Ahmed Ibrahim
223101109

Assignemt 1 : human face emotions

Overview

This analysis focuses on the **Multi-Layer Perceptron (MLP)** implementation for multi-class human face emotion classification based on the provided PyTorch code (Human Face Emotions.ipynb).

The task is **Multi-class classification** for 5 distinct emotions. The framework used is **PyTorch** with **torchvision transforms** for image preprocessing and augmentation.

Dataset & Preprocessing

The dataset for face emotion recognition contains **59,099** total samples.

The images are preprocessed to **48 \times 48 grayscale** (1 channel) and normalized (mean=0.5, std=0.5). Training data utilizes several augmentations, including random horizontal flip, random rotation , and random affine transformations, to improve generalization.

The task involves classifying images into **5 classes**: 'Angry', 'Fear', 'Happy', 'Sad', and 'Suprise'.

Data Split

The total samples are 59,099.

- **Training:** 80% or **47,279** samples.
- **Test:** 20% or **11,820** samples.

Model Architecture (MLPClassifier)

The MLP model used is a deep, fully-connected network with a large number of parameters, designed to process the flattened image data.

Input Dimension: The flattened times 48 grayscale image results in an input dimension of **2,304** features.

Architecture Detail: The network is sequential, beginning with a nn.Flatten() layer. It then consists of four hidden layers and one output layer, all using **ReLU** activation:

Linear (2304 to 1024) + ReLU

Linear (1024 to 512) + ReLU

Linear (512 to 256) + ReLU

Linear (256 to 128) + ReLU

Linear (128 to 5) for the final output logits.

Total Parameters: The model is highly complex for an MLP, having **3,049,989** total trainable parameters.

Regularization: No explicit regularization techniques like Batch Normalization or Dropout were included *within the neural network layers*. However, **data augmentation** was heavily used in the preprocessing steps to act as a form of implicit regularization.

Training Configuration

Parameter	**Value**
Optimizer	**Adam**
Learning Rate (lr)	**0.001**
Loss Function	**CrossEntropyLoss**
Batch Size	**128**
Epochs	**5**
Device	**'cpu'** (A CUDA-enabled GPU was not found/used in the

Parameter	Value
	execution)

Results

The model was trained for **5 epochs**. The progression is summarized below:

EPOCH	TRAIN LOSS	TRAIN ACC	TEST LOSS	TEST ACC
1	1.4783	36.63%	1.4369	40.17%
2	1.4220	40.23%	1.3917	41.95%
3	1.3842	42.39%	1.3856	42.04%
4	1.3655	43.37%	1.3659	43.27%
5	**1.3423**	**44.74%**	**1.3363**	**44.37%**

Final Performance (Epoch 5)

METRIC	TRAINING	TESTING
ACCURACY	44.74%	**44.37%**
LOSS	1.3423	1.3363
OVERFITTING GAP	**0.37%** (Train Acc $>$ Test Acc)	**Minimal**

Key Observations & Conclusion

The MLP achieved a final test accuracy of **44.37%** for the 5-class face emotion recognition task.

This result stands in **stark contrast** to the **90.88%** test accuracy achieved on the structured diabetes dataset. The substantial difference highlights a fundamental weakness of the standard Multi-Layer Perceptron architecture when applied to complex image data like face recognition.

While the model converged quickly, increasing the training and testing accuracy from Epoch 1 to 5, the overall performance remains low, suggesting that even with extensive data augmentation, a simple MLP is structurally incapable of efficiently extracting the necessary local, hierarchical features required for accurate image classification. The minimal overfitting gap of 0.37% suggests that the model is **underfitting** the complex problem space rather than suffering from poor generalization. The network is simply too shallow and the architecture (fully connected) is too unsophisticated for this task.