CS 203: Software Tools & Techniques for AI

IIT Gandhinagar

Sem-II - 2024-25

LAB 06

Lead TA: Section 1- Eshwar Dhande; Section 2- Himanshu Beniwal

Total marks: 100
Submission deadline: Tuesday, 25/02/2025 11:59 PM
Submission guidelines:

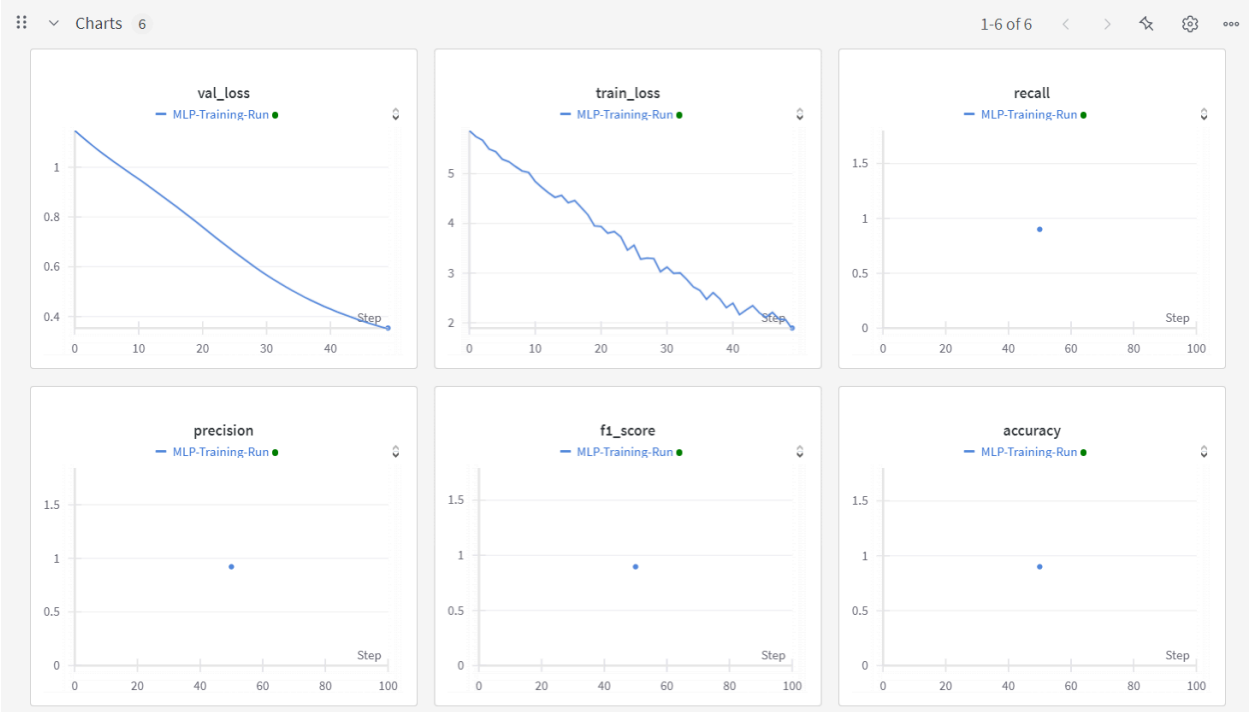
1. Code should be added to a GitHub repository, and the **repository details should be shared in the pdf.**
2. **Submit the PDF showing screenshots** of all steps involved in the following code.

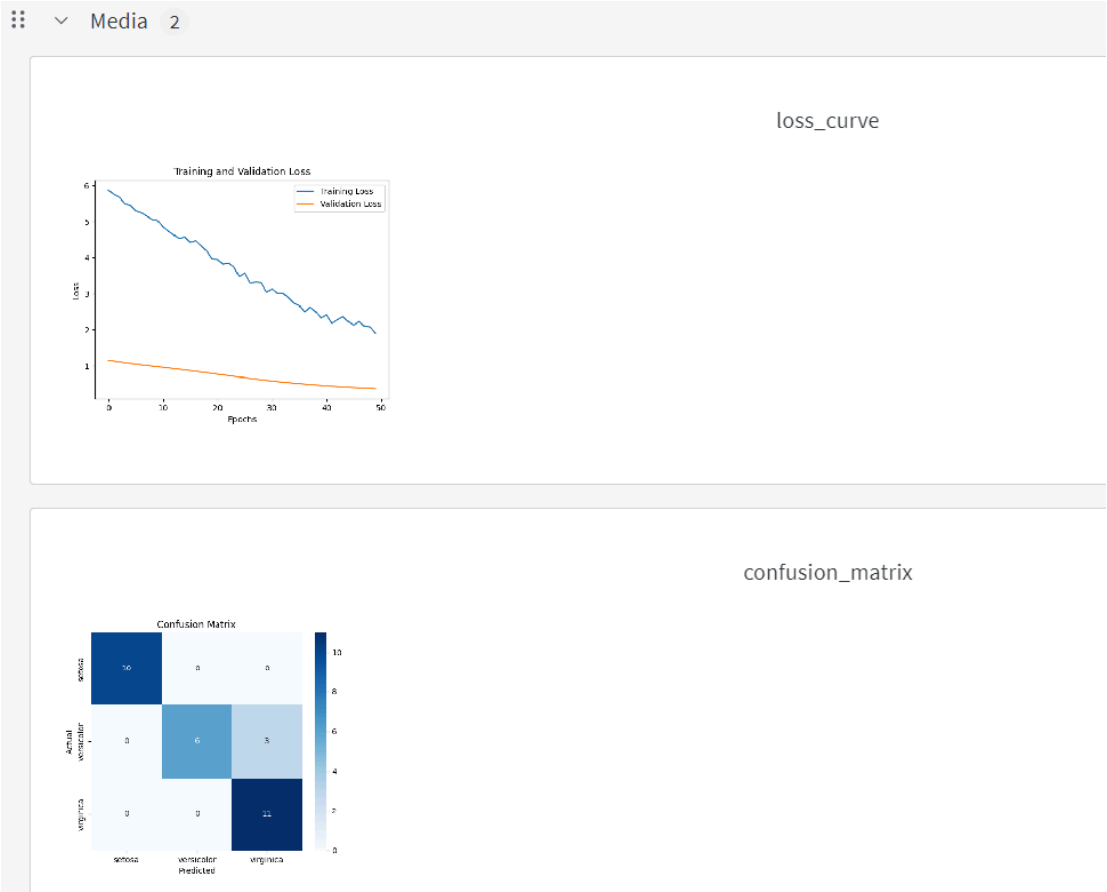
Note: By submitting this assignment solution you confirm to follow the IITGN's honor code. We shall strictly penalize the submissions containing plagiarized text/code.

Roll Number: (Aryan Solanki) 23110049 and (Parthiv Patel) 23110237

LINK TO GITHUB

Graph in Weight and Biases





■ Model architecture and hyperparameters.

The Multi-Layer Perceptron (MLP) model used for classification consists of the following layers:

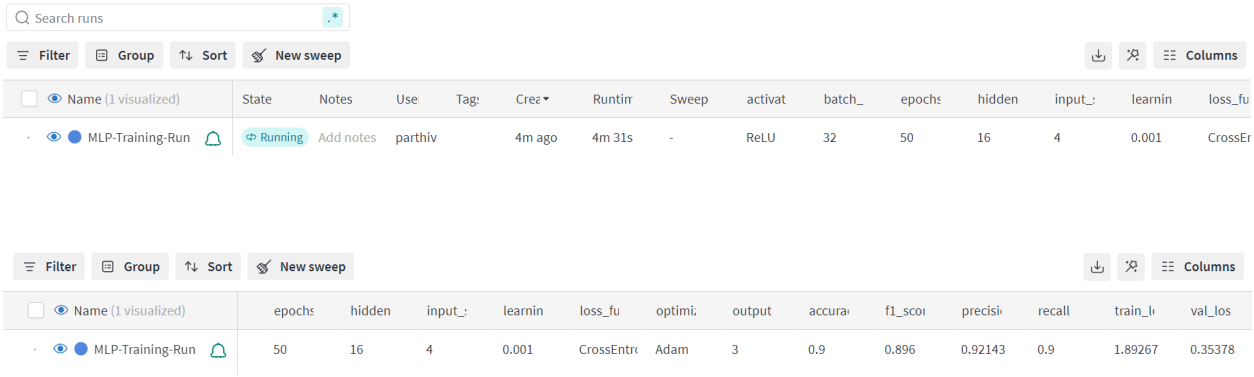
- **Input Layer**: 4 neurons (corresponding to the 4 input features)
- **Hidden Layer**: 16 neurons with **ReLU activation**
- **Output Layer**: 3 neurons with **softmax activation** for multi-class classification

The model was trained using the **Adam optimizer** with a learning rate of **0.001**. Other hyperparameters include:

- **Loss Function**: Categorical Cross-Entropy
- **Batch Size**: 32
- **Epochs**: 50

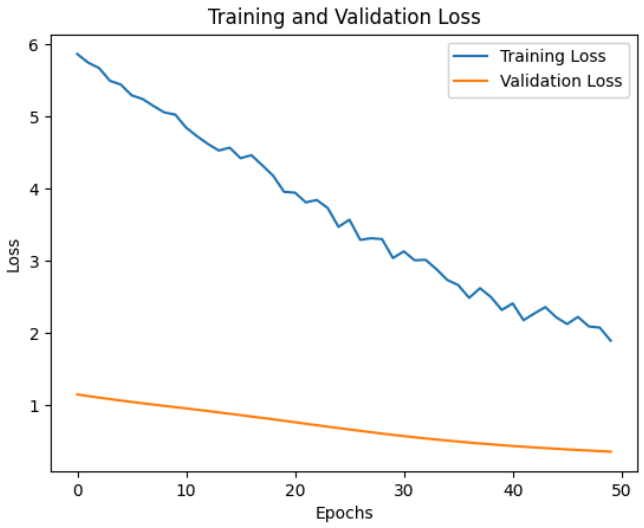
All these hyperparameters were logged in **Weights & Biases (W&B)** to ensure reproducibility and tracking of training progress.

It can also be seen in the W&B screenshot given below

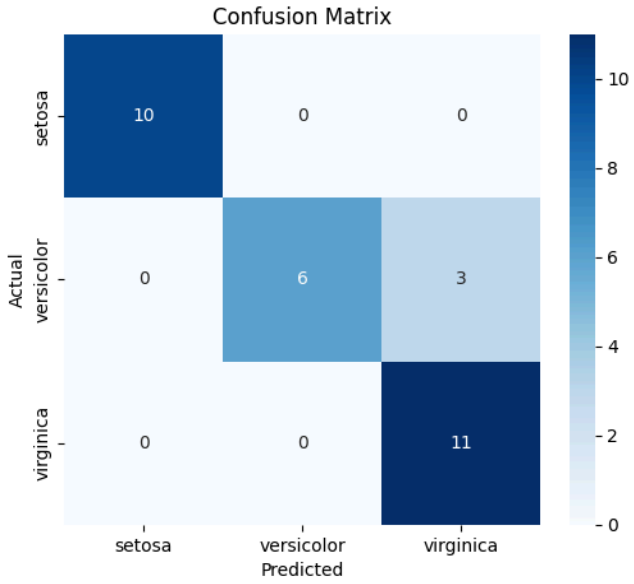


The following graph shows the loss curves over 50 epochs

Training & Validation loss Graph



Confusion Matrix



Final evaluation results.

```
print(accuracy, precision, recall, f1,)
✓ 0.0s

0.9 0.9214285714285714 0.9 0.896
```

Metric	Value
Accuracy	90%
Precision	92.1%
Recall	90%
F1	89.6%

Total loss in batch in each epoch

```
Epoch 1: train_loss=5.8671, val_loss=1.1470
Epoch 2: train_loss=5.7476, val_loss=1.1246
Epoch 3: train_loss=5.6729, val_loss=1.1030
Epoch 4: train_loss=5.4961, val_loss=1.0823
Epoch 5: train_loss=5.4439, val_loss=1.0622
Epoch 6: train_loss=5.2940, val_loss=1.0430
Epoch 7: train_loss=5.2436, val_loss=1.0243
Epoch 8: train_loss=5.1471, val_loss=1.0062
Epoch 9: train_loss=5.0572, val_loss=0.9883
Epoch 10: train_loss=5.0275, val_loss=0.9705
Epoch 11: train_loss=4.8466, val_loss=0.9532
Epoch 12: train_loss=4.7280, val_loss=0.9351
Epoch 13: train_loss=4.6204, val_loss=0.9163
Epoch 14: train_loss=4.5289, val_loss=0.8977
Epoch 15: train_loss=4.5689, val_loss=0.8789
Epoch 16: train_loss=4.4225, val_loss=0.8599
Epoch 17: train_loss=4.4645, val_loss=0.8407
Epoch 18: train_loss=4.3239, val_loss=0.8215
Epoch 19: train_loss=4.1788, val_loss=0.8017
Epoch 20: train_loss=3.9562, val_loss=0.7817
Epoch 21: train_loss=3.9455, val_loss=0.7612
Epoch 22: train_loss=3.8099, val_loss=0.7407
Epoch 23: train_loss=3.8434, val_loss=0.7203
Epoch 24: train_loss=3.7337, val_loss=0.7000
Epoch 25: train_loss=3.4714, val_loss=0.6806
...
Epoch 47: train_loss=2.2222, val_loss=0.3767
Epoch 48: train_loss=2.0895, val_loss=0.3687
Epoch 49: train_loss=2.0737, val_loss=0.3612
Epoch 50: train_loss=1.8927, val_loss=0.3538
```

SECTION 2

Compare manual tuning vs. automated search

- Which approach is better, and why?

ANS: In general, automated search is advantageous because it methodically examines a wider range of hyperparameters, minimising human bias and effort while frequently attaining better results. However, when resources are scarce, manual adjustment may be helpful for short experiments.

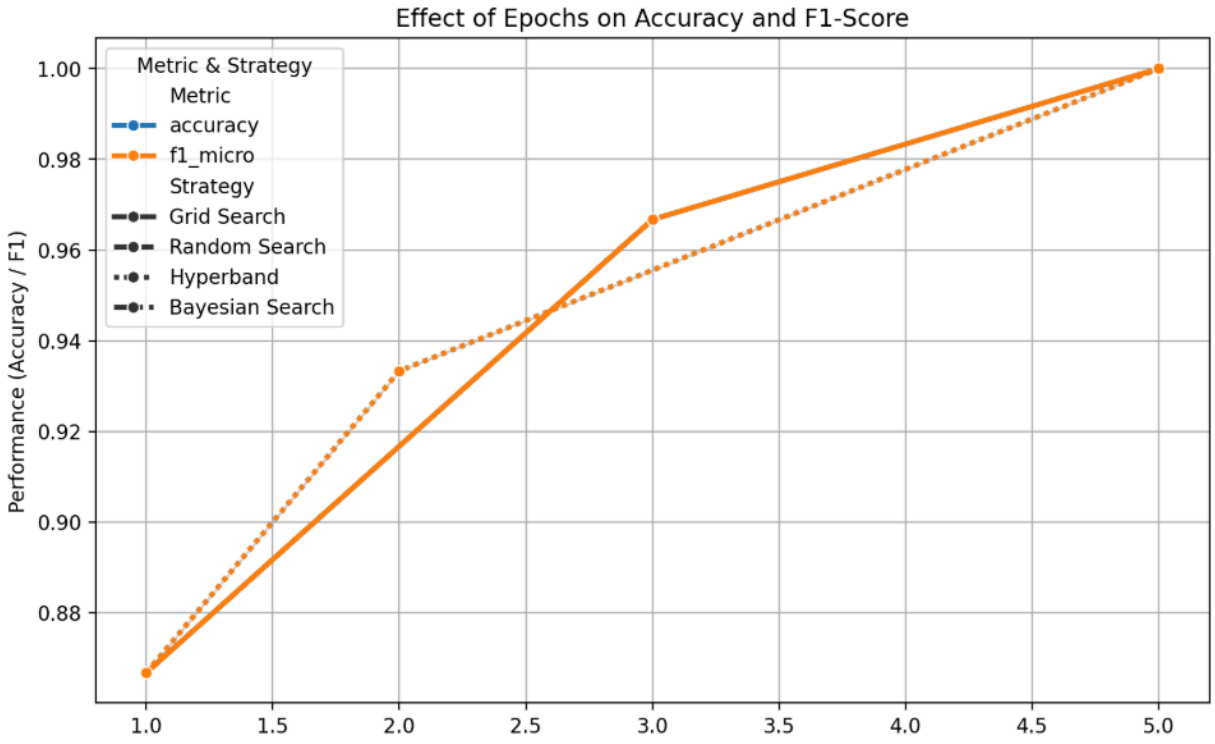
—----------

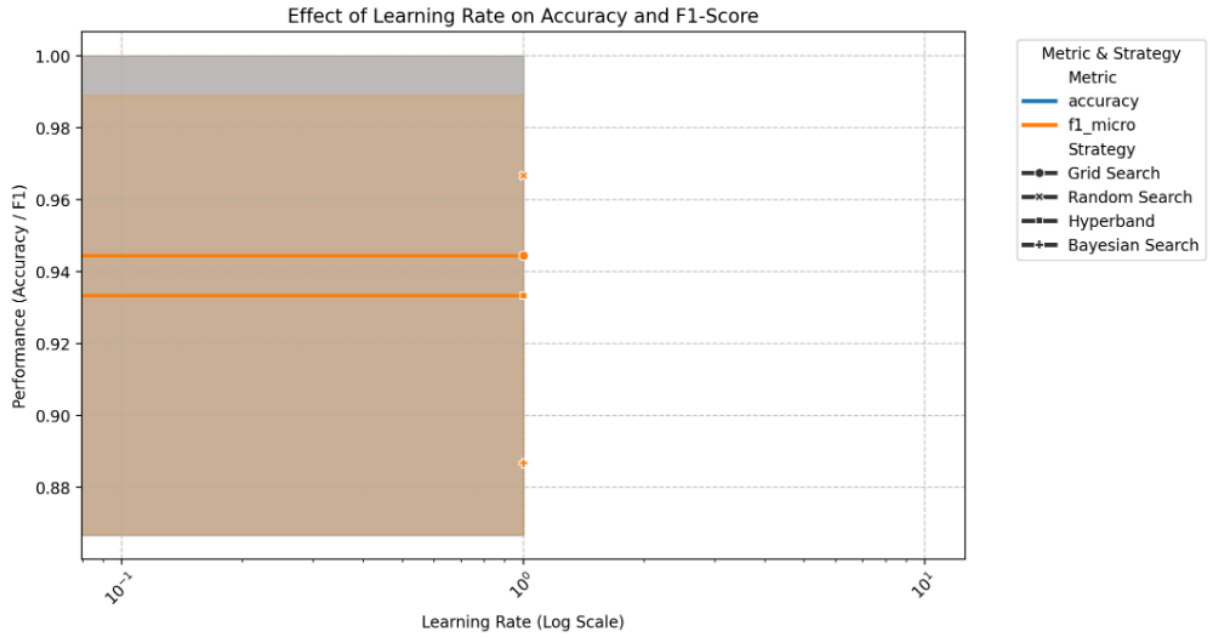
Epochs vs. Performance: accuracy and F1-score increase with more epochs. At one epoch, performance is around 86.67%, while at five epochs, both metrics reach 100%, confirming a direct relationship.

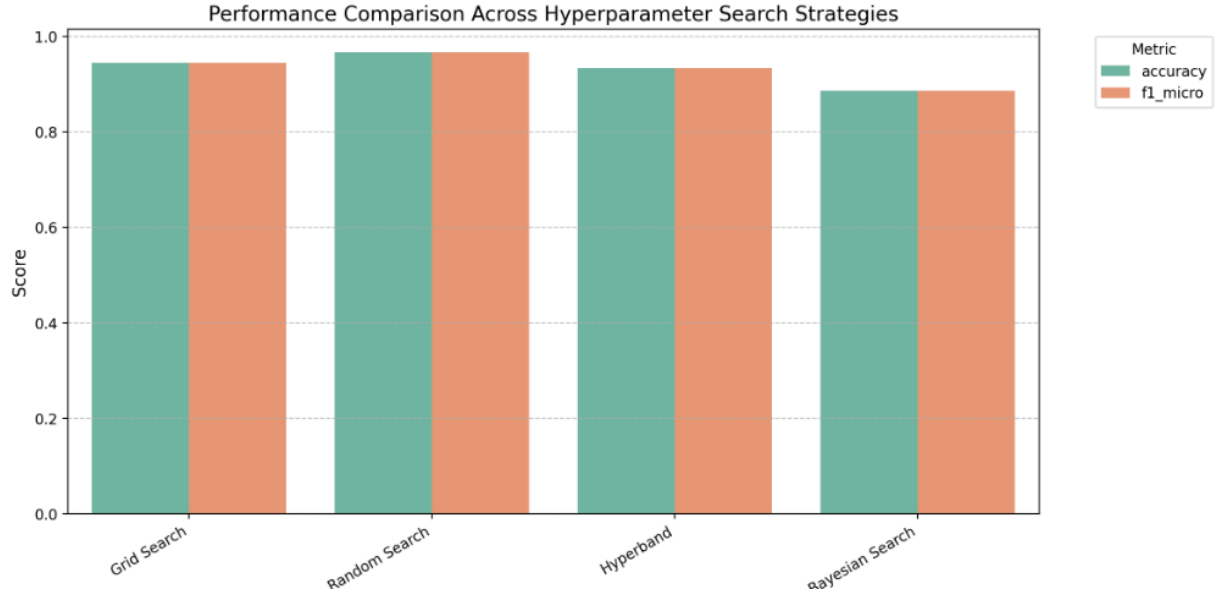
Learning Rate vs. Performance: A higher learning rate (0.001) allows faster convergence, achieving near-optimal performance in fewer epochs. A lower learning rate (0.00001) still achieves good results but requires more epochs to reach similar accuracy.

Batch Size vs. Performance (Hypothetical): Smaller batch sizes (e.g., 2 or 4) may provide more frequent updates, improving convergence and final performance. However, a batch size that is too small may introduce noise, affecting stability. Larger batch sizes could result in smoother but slower updates, potentially reducing generalization ability.

Plots for the training vs. validation loss for each hyperparameter configuration.

- 


Effect of Learning Rate on Accuracy and F1-Score
Performance (Accuracy / F1)
1.00
0.98
0.96
0.94
0.92
0.90
0.88
10^{-1}
10^{0}
10^{1}
Learning Rate (Log Scale)
Metric & Strategy
Metric
accuracy
f1_micro
Strategy
Grid Search
Random Search
Hyperband
Bayesian Search


Performance Comparison Across Hyperparameter Search Strategies
Score
1.0
0.8
0.6
0.4
0.2
0.0
Grid Search
Random Search
Hyperband
Bayesian Search
Metric
accuracy
f1_micro