AI: Machine Learning Foundations

Final Project

Your final project is designed to demonstrate your understanding of AI models and the training process.

Fashion MNIST Dataset:

The Fashion MNIST dataset consists of grayscale images of fashion items with a size of 28x28 pixels. You can download the dataset directly through PyTorch.

Model Baseline:

In this project, you will build and train a convolutional neural network (CNN) for classifying fashion items. You will then explore the impact of hyperparameter tuning on model performance.

Timeline:

Part 1 will be due by the start of the second class. Part 2 will be due by the end of the second class. Part 3 will be due by the end of the fourth class.

Part 1 Loading the dataset and model - Due on 8/20/2024:

For this part, you will have loaded the Fashion MNIST dataset and a CNN model using Python and the PyTorch library. This is the first checkpoint to make sure the training sessions will be performed properly. For the convolutional layers, kernel size should be set to 5, the number of filters should be set to 8 initially, and padding and stride should be set to their default values.

Part 2 First Training Session - Due on 8/22/2024:

For this part, you must create the training loop and run your first training session with your initial hyperparameters (number of filters: 8, batch size: 64, epochs: 3). Your loss function should be a cross entropy loss function and your optimizer should be the Adam optimizer with a learning rate of 0.01.

1. Make sure the initial hyperparameters are set.

2. Implement code to save your trained model after the training session. This ensures you can reload the model if needed, especially when working on Google Colab.
3. Evaluate the performance of your trained model on the Fashion MNIST test set. Calculate and record the accuracy metric.

You will submit this by uploading the model (.pt file) to Teams or linking to it in the correct spot in Teams.

Part 3 Requirements:

Requirements for Python Code:

1. Exploratory Analysis
    a. Select at least two of the initial hyperparameters (number of filters, batch size, epochs) for exploration. Experiment with different values for your chosen hyperparameters and train separate models for each configuration.
    b. Evaluate the performance (accuracy) of each trained model on the Fashion MNIST test set. Compare the results and analyze how the chosen hyperparameter variations impacted model performance.
    c. Save all the trained models (initial and tuned) along with their corresponding evaluation scores on the Fashion MNIST dataset. You will need this information for the project write-up.
2. Conclusion Drawing
    a. Create at least two graphs to determine/demonstrate/show any meaningful correlations in the change of a hyperparameter and the model evaluation score.
    b. Customize the graphs to be presentable (change color, scale, labels, or whatever else is needed to make them accurate and meaningful).

Requirements for Write-Up (1-2 pages length, not including the code).

a. Your write-up must include the following parts: Summary or Abstract, Objective, Process, Results, Conclusion, and Further Steps.
b. Your write-up must include a consistent and professional format.
c. You must cite any outside information that you used in your project.
d. You must insert items from your code into your write-up. You must have at least the following in your write-up: 2 figures, and 1 code snippet.
e. It is recommended that you use a style guide to remain consistent with your formatting, although not required.
f. You must have a write-up free of grammatical and spelling errors to the best of your ability. Feel free to use tools like Grammarly to help with this.

g. Make sure that your writing remains professional. Make sure to monitor your language choice and precision.

Submission - Due on 9/1/2024:

You must submit your final project by [Fill in Date]. When you submit your project, it must include the following items: Folder of trained models (.pt), Code File (.py or .ipynb), and Writeup. You can submit your files by uploading them to the Teams assignment or link the Teams assignment to a public GitHub repository with the files in there.

Review Rubric

Category	0 Points	1 Point	2 Points	3 Points	Score
CNN Architecture	Incomplete or poorly designed architecture.	Basic architecture might lack essential CNN components.	Functional architecture with core CNN components.	Well-designed architecture with appropriate layers for CNN tasks.	
Model Training	Poor or missing documentation of the training process.	Limited documentation of the training process.	Training process is documented.	Training loop is well-defined with appropriate optimizer, loss function, and metrics.	
Hyperparameter Tuning	No hyperparameter tuning performed.	Limited exploration of hyperparameters.	Exploration of hyperparameters is attempted.	Exploration of hyperparameters with clear rationale for chosen configurations.	
Model Evaluation	Minimal or missing analysis of hyperparameter tuning.	Limited analysis of hyperparameter tuning results.	Analysis of hyperparameter tuning is present.	In-depth analysis of the impact of hyperparameter tuning on model performance.	
Project Report	Poorly organized	Report is organized but	Clear report structure with	Well-organized report with	

	report or missing key sections.	might lack details or clarity.	explanations and some figures.	clear explanations, figures, and references.	
Code Quality	Code is not functional.	Code is functional but missing important details.	Code is functional.	Well-structured and documented code with comments and clear variable names.	
Total Points					