PROBLEM STATEMENT DOCUMENT

SPACE TRAFFIC DENSITY PREDICTION

Team Members

1.Shravani R S

2.Himaja S

3.Tarunaa A C

4.Karanbir Singh

5.Anushka Dutta

Date: 19th December 2024

Prepared By: Shravani R S

Table of Contents

1. **Development Phase**
    - 1.1 Hyperparameter Tuning
    - 1.2 Data Loading and Preprocessing
    - 1.3 Model Definition and Hyperparameter Tuning
    - 1.4 Model Evaluation
    - 1.5 Visualization
2. **PyTorch Model**
    - 2.1 Set Up PyTorch Model
    - 2.2 Define Hyperparameters and Model Training
    - 2.3 Hyperparameter Tuning with GridSearch
    - 2.4 Model Training
    - 2.5 Evaluation and Metrics
    - 2.6 Model Saving and Loading
    - 2.7 Model Optimization
    - 2.8 Transfer Learning
    - 2.9 Model Pruning
    - 2.10 Distributed Training
    - 2.11 TensorBoard for Visualization
    - 2.12 Deployment and Inference
    - 2.13 Quantization
    - 2.14 Custom Loss Functions and Optimizers
3. **Deployment Phase**
    - 3.1 Model Exporting
    - 3.2 Creating an API for Model Serving
4. **Grid Search in PyTorch**
    - 4.1 Imports and Model Definition
    - 4.2 Data Generation and Preprocessing
    - 4.3 Hyperparameter Grid Definition
    - 4.4 Grid Search and Model Training
    - 4.5 Model Evaluation on Validation Set
    - 4.6 Tracking Best Hyperparameters

Problem Statement

Modern machine learning (ML) workflows involve numerous steps, from data preparation and model definition to hyperparameter tuning, evaluation, and deployment. Each phase comes with its own set of challenges that can significantly impact model performance, scalability, and usability in real-world applications. This document outlines the key problems addressed in the ML pipeline, with a focus on PyTorch as the core framework for implementation.

Challenges and Problems Addressed

Development Phase

1.1 Hyperparameter Tuning

- **Problem**: Identifying the best combination of hyperparameters (e.g., learning rate, batch size) is time-consuming and computationally expensive.
- **Impact**: Poorly chosen hyperparameters can lead to suboptimal model performance.

1.2 Data Loading and Preprocessing

- **Problem**: Handling raw data often involves dealing with missing values, inconsistencies, and large volumes, which can slow down the pipeline.
- **Impact**: Poorly preprocessed data leads to unreliable models and lower accuracy.

1.3 Model Definition and Hyperparameter Tuning

- **Problem**: Designing an effective architecture and tuning it simultaneously is complex and prone to errors.
- **Impact**: A non-optimal architecture may result in overfitting or underfitting.

1.4 Model Evaluation

- **Problem**: Evaluating models requires consistent metrics and robust validation processes.
- **Impact**: Inconsistent evaluation leads to misleading performance insights.

1.5 Visualization

- **Problem**: Lack of real-time visualization tools during training and evaluation.
- **Impact**: Difficulty in monitoring model progress and debugging issues effectively.

PyTorch Model

2.1 Set Up PyTorch Model

- **Problem**: Defining complex models in PyTorch can be error-prone without proper guidance.
- **Impact**: Errors in model design can hinder training and evaluation.

2.2 Define Hyperparameters and Model Training

- **Problem**: Setting inappropriate hyperparameters can negatively impact convergence and training speed.
- **Impact**: Inefficient training can lead to wasted resources and time.

2.3 Hyperparameter Tuning with GridSearch

- **Problem**: Exhaustive hyperparameter searches require high computational resources.
- **Impact**: Limited exploration of parameter space may leave the model under-optimized.

2.4 Model Training

- **Problem**: Ensuring proper gradient updates and convergence during training is challenging.
- **Impact**: Poor training results in models that fail to generalize.

2.5 Evaluation and Metrics

- **Problem**: Selecting appropriate evaluation metrics for different tasks is critical.
- **Impact**: Incorrect metrics can misrepresent model performance.

2.6 Model Saving and Loading

- **Problem**: Saving and reloading models without errors or inconsistencies.
- **Impact**: Deployment or further fine-tuning becomes difficult.

2.7 Model Optimization

- **Problem**: Optimizing large models for speed and memory efficiency.
- **Impact**: High computational costs and long inference times.

2.8 Transfer Learning

- **Problem**: Adapting pre-trained models to new tasks while retaining performance.
- **Impact**: Ineffective fine-tuning leads to subpar results.

2.9 Model Pruning

- **Problem**: Reducing model size without losing accuracy.
- **Impact**: Large models are unsuitable for edge devices and real-time applications.

2.10 Distributed Training

- **Problem**: Efficiently scaling training across multiple GPUs or systems.
- **Impact**: Poorly configured setups waste computational resources.

2.11 TensorBoard for Visualization

- **Problem**: Lack of integrated tools for tracking metrics during training.
- **Impact**: Debugging and monitoring training progress becomes cumbersome.

2.12 Deployment and Inference

- **Problem**: Translating trained models into production-ready systems.
- **Impact**: Deployment errors disrupt real-time predictions.

2.13 Quantization
- **Problem**: Reducing model size while maintaining accuracy.
- **Impact**: Inefficient models hinder deployment on resource-constrained devices.

2.14 Custom Loss Functions and Optimizers
- **Problem**: Standard loss functions and optimizers may not meet all application requirements.
- **Impact**: Limited task-specific performance.

Deployment Phase

3.1 Model Exporting
- **Problem**: Ensuring compatibility and portability across platforms.
- **Impact**: Lack of standardized formats can block cross-platform deployment.

3.2 Creating an API for Model Serving
- **Problem**: Building scalable and efficient APIs for real-time predictions.
- **Impact**: Poorly implemented APIs can introduce latency and errors in predictions.

Grid Search in PyTorch

4.1 Imports and Model Definition
- **Problem**: Properly defining the model and importing required libraries.
- **Impact**: Errors in setup delay the development process.

4.2 Data Generation and Preprocessing
- **Problem**: Generating suitable datasets for training and validation.
- **Impact**: Poor data quality affects Grid Search results.

4.3 Hyperparameter Grid Definition
- **Problem**: Defining hyperparameter ranges for effective exploration.
- **Impact**: Incomplete grids leave optimal configurations unexplored.

4.4 Grid Search and Model Training
- **Problem**: Managing computational resources for multiple training runs.
- **Impact**: Resource constraints limit the parameter search space.

4.5 Model Evaluation on Validation Set
- **Problem**: Identifying the best configuration using validation metrics.
- **Impact**: Inconsistent evaluation can lead to incorrect conclusions.

4.6 Tracking Best Hyperparameters

- **Problem**: Logging and managing results effectively.
- **Impact**: Lack of proper tracking leads to reproducibility issues.