**REPORT**

**1. Objective**

The goal is to analyze traffic data and predict or reconstruct traffic flow using an **Autoencoder Neural Network**, with a focus on how different optimization strategies and batch normalization affect performance.

**2. Dataset Overview**

* **Initial Cleaning:**
  + Dropped rows with too many missing values (≥30%).
  + Imputed missing categorical values like Road\_Condition with "Unknown".

**3. Preprocessing**

* StandardScaler was used from sklearn.preprocessing to normalize input features.
* Dataset was split using train\_test\_split() into training and testing sets.
* Tensor datasets and DataLoaders were created to support different GDM modes:
  + Full Batch
  + Mini-Batch
  + Stochastic (1-sample)

**4. Model Architecture**

* **Autoencoder** built using torch.nn.Module
* Consists of:
  + Encoder: Compresses input to latent space
  + Decoder: Reconstructs input from compressed space
* Includes experiments with and without **Batch Normalization**

**5. Training Strategy**

Also experimented with 3 gradient descent variants:

|  |  |  |  |
| --- | --- | --- | --- |
| **Variant** | **Batch Size** | **DataLoader Setup** | **Behavior** |
| Full Batch | All data | 1 batch | Slow but stable |
| Mini-Batch | 32–64 | Multiple batches | Balance between speed & accuracy |
| Stochastic | 1 sample | Single sample | Fast updates but noisy |

**6. Results & Comparisons**

Key plots (detected in later cells):

* **Training loss vs Epochs**
* **Comparison of loss curves for the 3 optimization modes**
* Visual reconstruction of input vs predicted flow

**Observations:**

* **Mini-Batch Gradient Descent** showed faster convergence and stable performance.
* **Batch Normalization** improved:
  + Convergence speed
  + Model stability
  + Slight increase in accuracy
* **Stochastic Gradient Descent** showed noisy curves but potentially faster initial learning.

**7. Conclusion**

Using an Autoencoder with Batch Normalization and Mini-Batch training led to the most effective traffic flow prediction. The reconstruction errors were minimized better, and training was stable.

.