# Option 1 - Linear Regression Using TensorFlow

In this assignment, I explored the fundamentals of using TensorFlow for linear regression. The objective is to generate a synthetic dataset with linear trends, add some noise to mimic real-world data variability, and then build a TensorFlow model that learns to fit a line to this data. This exercise emphasizes key machine learning concepts, such as defining placeholders for inputs, initializing trainable parameters, and minimizing a cost function using an optimizer.

Started by visualizing the noisy dataset to understand the underlying linear relationship. Next, we build the linear regression model by defining TensorFlow variables for the slope and intercept (bias), which will be updated during training. I used gradient descent to minimize the mean squared error (MSE) between the model’s predictions and the actual data points. By running the training loop for 1000 epochs, aim is to find the optimal weight and bias that minimize this error.

Finally, I visualized the learned linear relationship by plotting the fitted line on top of the noisy data. This exercise not only reinforces the principles of linear regression but also provides hands-on experience with TensorFlow’s computational graph, variable management, and session-based execution. The final deliverables include the Python code, screenshots of the plots, and this introduction summarizing the approach and goals.

# Execution

Setting up the baseline data with random shuffling using np.random

A screenshot of a computer

AI-generated content may be incorrect.

Visualize the set up data

A graph with blue dots

AI-generated content may be incorrect.

Creating tensor flow linear regressin and visualizing the output

A screenshot of a computer

AI-generated content may be incorrect.