

Observations on Training a Neural Network with Numbers and Their Squares using PyTorch

Objective: Train a neural network model in PyTorch to predict the square of a given number.

Dataset:

- Input: A list of numbers, e.g., [1, 2, 3, 4, ...].
- Target: Corresponding squares of these numbers, e.g., [1, 4, 9, 16, ...].

Training Procedure:

Initialized a basic feedforward neural network in PyTorch.
Used the Mean Squared Error (MSE) loss function since it's a regression task.
Employed the Adam optimizer with a learning rate of $1e-4$.
Conducted training for 50 epochs with a batch size of 2.

Observations:

Convergence: The model demonstrated unsteady convergence across epochs, with loss declining slowly.
Performance: By the final epoch, the training loss was not minimal, indicating a not right fit to the data
Large Numbers: The model had some difficulty predicting the square of numbers accurately.

Recommendations:

To improve generalization, introduce dropout layers or regularization techniques.
Consider normalizing the input data to aid the training process, especially if the range of numbers is large.

Conclusion: Using PyTorch, the neural network trained on the dataset of numbers and their squares showed machine learning models show weaker performance with mathematical functions.