Computer Science & Information Systems

Deep Neural Networks - Lab sheet 2

Linear Neural Network for Regression

Prepared by Seetha Parameswaran

1 Objective

The objective is to

- Understand linear regression using a single neuron.
- Implement gradient descent optimization from scratch.
- Predict continuous values using the Auto MPG dataset.

2 Steps to be Performed

- **Tool:** Python3
- **Libraries required:** numpy, matplotlib, pandas, sklearn
- **Input:** auto-mpg.data (from UCI repository)
- **Deep Learning Model:** Linear Regression
- **ANN Architecture:** Single Neuron (no hidden layers) with identity activation function
- **Implementation:** L2-Linear Regression.ipynb

2.1 Steps

- Import required Python libraries.
- Load the Auto MPG dataset and convert to dataframe.
- Handle missing values and remove non-numeric columns.
- Understand the problem: Predict MPG from vehicle features.
- Prepare the data: Extract features (X) and target (y).
- Partition the dataset into training (80%) and testing (20%) sets.
- Standardize features using StandardScaler (zero mean, unit variance).
- Create a SingleNeuronRegressor object with learning rate and iterations.
- Train the model using batch gradient descent.

- Predict values for the testing set using the trained model.
- Compute evaluation metrics: MSE, RMSE, MAE, R^2 score.
- Visualize training loss, predictions vs actual, and residuals.
- Analyze learned feature weights for interpretation.

2.2 Mathematical Formulation

$$\text{Model:} \quad \hat{y} = \mathbf{w}^T\mathbf{x} + b \tag{1}$$

$$\text{Loss Function (MSE):} \quad J(\mathbf{w}, b) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2}$$

$$\text{Gradients:} \quad \frac{\partial J}{\partial \mathbf{w}} = \frac{1}{N}\mathbf{X}^T(\hat{\mathbf{y}} - \mathbf{y}) \tag{3}$$

$$\frac{\partial J}{\partial b} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)}) \tag{4}$$

$$\text{Parameter Update:} \quad \mathbf{w} := \mathbf{w} - \eta\frac{\partial J}{\partial \mathbf{w}} \tag{5}$$

$$b := b - \eta\frac{\partial J}{\partial b} \tag{6}$$

3 Results

- Linear regression model successfully trained using gradient descent.
- Model achieved $R^2 \approx 0.77$ on test set, explaining 77% of variance.
- Training loss decreased smoothly over 1000 iterations.
- RMSE indicates average prediction error.
- Feature weights learned: negative for weight/horsepower (heavier cars use more fuel), positive for model year (newer cars more efficient).
- Residual plots show randomly scattered errors, validating linear model assumptions.

4 Observation

- Single neuron with identity activation effectively models linear relationships.
- Feature scaling (standardization) is critical for gradient descent convergence.
- Batch gradient descent provides stable, deterministic optimization.
- Learned weights align with physical intuition about vehicle fuel efficiency.
- Model generalizes well with similar performance on training and test sets.
- Linear regression provides interpretable baseline for regression problems.