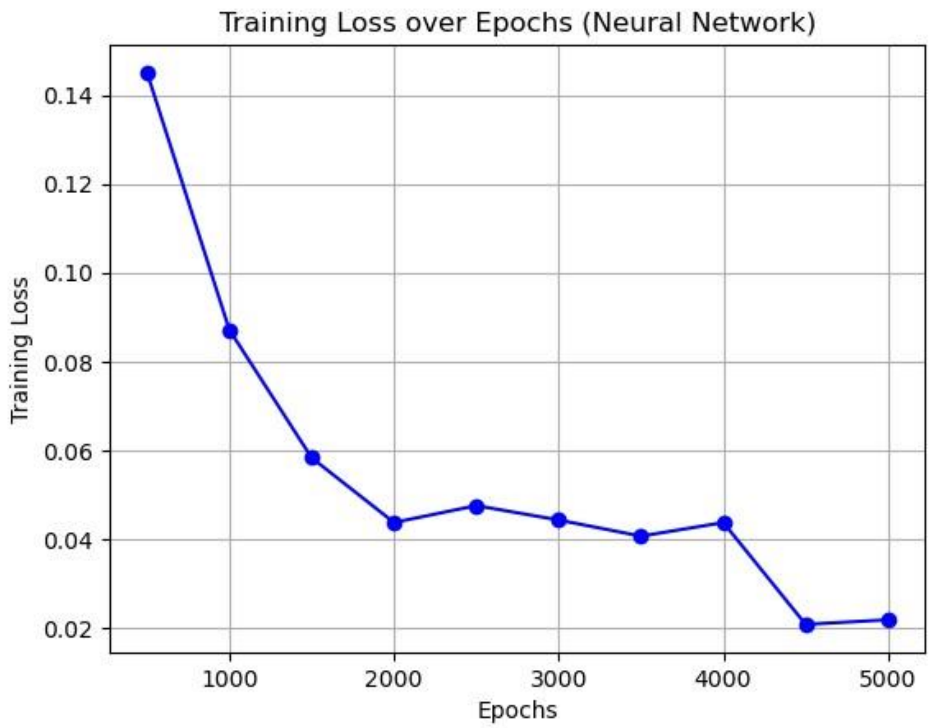
Neural Network vs. Logistic Regression on Diabetes Dataset

This project aims to compare the performance of a neural network model with multiple hidden layers to that of a simple logistic regression model using the Diabetes dataset. The work includes the implementation, training, and evaluation of both models, as well as a detailed analysis of the results. The project is implemented in TensorFlow 1.x and involves loading and preprocessing the Dataset (from part1), splitting the data to train and test, neural network model, logistic regression model and comparison.

The full notebook is available as "DeepLearningPart2" files (.ipynb, .md) in the zipped file, as well as the plots (.png files). To review the detailed hyperparameters refer to the attached list at the end of the document.

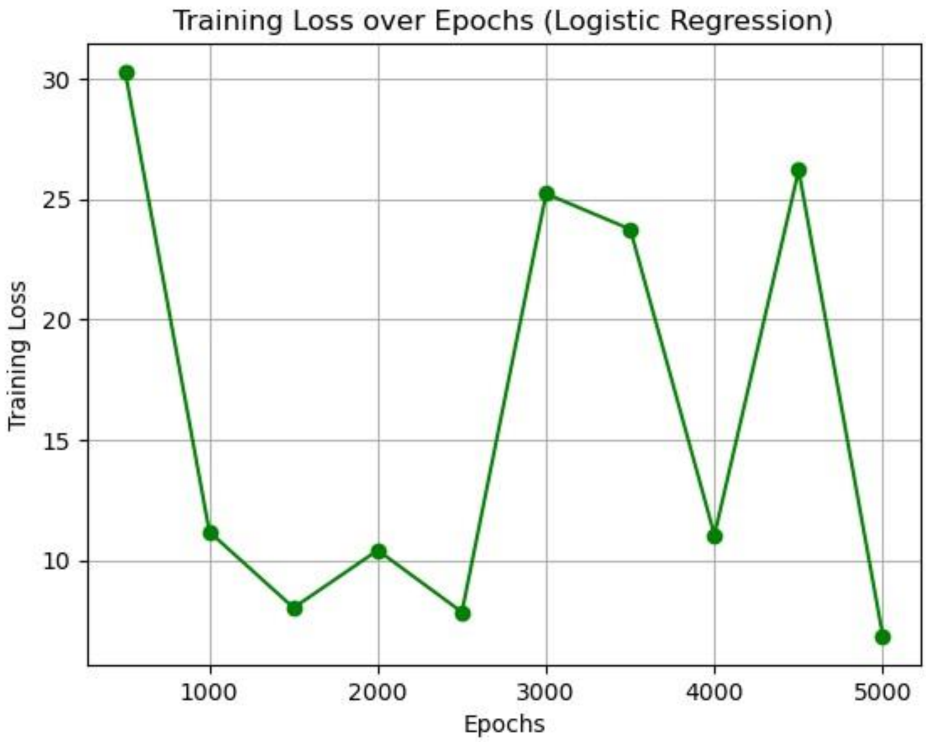
Neural Network Model

The neural network model is designed with L=3 hidden layers (each with 32 neurons) to capture intricate patterns and relationships within the Diabetes dataset. The architecture includes an input layer, several hidden layers with ReLU activation functions, and an output layer. The model is trained using the mean squared error loss function and the Adam optimizer. During training, adjustments are made to the number of hidden layers, units, and learning rates to find the optimal configuration. Regularization techniques and dropout layers are employed to prevent overfitting.



Logistic Regression Model

The logistic regression model serves as a baseline comparison to the more complex neural network. It consists of a single layer with a sigmoid activation function, making it suitable for binary classification. The model uses binary cross-entropy loss and is trained with gradient descent. Feature scaling is applied to improve convergence, and efforts are made to address underfitting by adjusting learning rates and batch sizes.

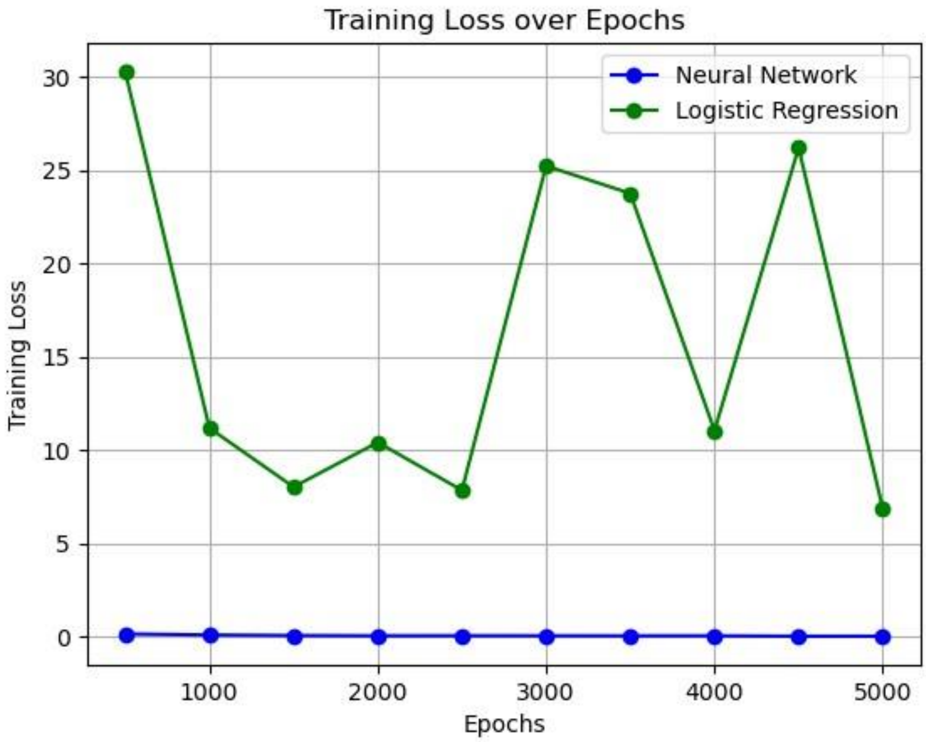


Comparison and Scores

The comparison between the neural network and logistic regression models is visualized through training loss plots over epochs. The neural network consistently outperforms the logistic regression model, showcasing its superior ability to learn complex patterns. The training losses for the neural network demonstrate a significant reduction over time, indicating effective learning.

Scores:

- Neural Network: Achieves a final training loss of approximately 0.02 after 5000 epochs.
- Logistic Regression: Attains a final training loss of around 6.83 after 5000 epochs.



The lower training loss of the neural network indicates better model fit and superior performance in capturing the underlying patterns of the Diabetes dataset. The visualized plots clearly depict the divergence in learning capabilities between the two models, emphasizing the effectiveness of the neural network in comparison to the logistic regression model for this specific classification task.

By

Alon Meshulam

Ori Ekshtein

Ariel Dvori

Neural Network (NN) Model:

1. Number of Hidden Layers (L):
    - L is set to 3.
2. Number of Neurons per Hidden Layer:
    - hidden_size is set to 32.
3. Activation Function:
    - ReLU (Rectified Linear Unit) activation is used in each hidden layer.
4. Learning Rate:
    - learning_rate is set to 0.001 for the Adam optimizer.
5. Epochs:
    - Training is performed for 5000 epochs.
6. Batch Size:
    - batch_size is set to 32.
7. Optimizer:
    - Adam optimizer is used.
8. Loss Function:
    - Mean Squared Error (MSE) loss is used.

Logistic Regression (LogR) Model:

1. Learning Rate:
    - learning_rate is set to 0.01 for the Gradient Descent optimizer.
2. Epochs:
    - Training is performed for 5000 epochs.
3. Batch Size:
    - batch_size is set to 32.
4. Loss Function:
    - Binary Cross-Entropy loss is used for logistic regression.