Computer Science & Information Systems

Deep Neural Networks - Lab sheet 3

Linear Neural Network for Binary Classification

Prepared by Seetha Parameswaran

1 Objective

The objective is to

- Understand logistic regression using a single neuron with sigmoid activation.
- Implement stochastic gradient descent (SGD) from scratch.
- Classify tumors as malignant or benign using Breast Cancer Wisconsin dataset.

2 Steps to be Performed

- **Tool:** Python3
- **Libraries required:** numpy, matplotlib, pandas, sklearn
- **Input:** Breast Cancer Wisconsin (Diagnostic) Dataset (from UCI repository)
- **Deep Learning Model:** Logistic Regression
- **ANN Architecture:** Single Neuron (no hidden layers) with Sigmoid activation function
- **Implementation:** L3-Logistic Regression.ipynb

2.1 Steps

- Import required Python libraries.
- Load the Breast Cancer Wisconsin dataset from sklearn.
- Understand the problem: Binary classification (malignant vs benign).
- Prepare the data: Extract features (X) and labels (y).
- Partition the dataset into training (80%) and testing (20%) sets.
- Standardize features using StandardScaler (zero mean, unit variance).
- Create a LogisticRegressionNeuron object with learning rate and epochs.
- Train the model using stochastic gradient descent (SGD).
- Predict class labels and probabilities for the testing set.

- Compute evaluation metrics: Accuracy, Precision, Recall, F1-score.
- Generate and analyze confusion matrix.
- Visualize training history (loss and accuracy curves).
- Plot predictions with confidence scores.

2.2 Mathematical Formulation

$$\text{Model (Linear Combination):} \quad z^{(i)} = \mathbf{w}^T \mathbf{x}^{(i)} + b \tag{1}$$

$$\text{Sigmoid Activation:} \quad \hat{y}^{(i)} = \sigma(z^{(i)}) = \frac{1}{1 + e^{-z^{(i)}}} \tag{2}$$

$$\text{Loss Function (Binary Cross-Entropy):} \quad J(\mathbf{w}, b) = -\frac{1}{N} \sum_{i=1}^{N} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right] \tag{3}$$

$$\text{Gradients:} \quad \frac{\partial J}{\partial \mathbf{w}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)}) \mathbf{x}^{(i)} \tag{4}$$

$$\frac{\partial J}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)}) \tag{5}$$

$$\text{Parameter Update (SGD):} \quad \mathbf{w} := \mathbf{w} - \eta \frac{\partial \ell}{\partial \mathbf{w}} \tag{6}$$

$$b := b - \eta \frac{\partial \ell}{\partial b} \tag{7}$$

where ℓ is the loss for a single example in SGD.

3 Results

- Logistic regression model successfully trained using SGD.
- Model achieved $\approx 97\%$ accuracy on test set for cancer classification.
- Binary cross-entropy loss decreased smoothly over 1000 epochs.
- High precision (≈ 0.98) and recall (≈ 0.97) indicate balanced performance.
- F1-score of ≈ 0.97 demonstrates excellent classification capability.
- Confusion matrix shows minimal false negatives (critical for medical diagnosis).
- Sigmoid activation successfully maps logits to probability range [0, 1].
- Model provides confidence scores for each prediction, enabling threshold adjustment.

4 Observation

- Single neuron with sigmoid activation effectively performs binary classification.
- Sigmoid function outputs probabilities, unlike perceptron's binary step function.
- Cross-entropy loss is appropriate for classification, penalizing confident wrong predictions.
- SGD (one example per update) converges faster than batch gradient descent.
- Feature scaling remains critical for stable and fast convergence.
- Model learned linear decision boundary separating malignant and benign tumors.
- Probabilistic output enables medical practitioners to assess prediction confidence.
- Low false negative rate is crucial for cancer detection applications.