Computer Science & Information Systems

Deep Neural Networks - Lab sheet 4

Linear Neural Network for Multi-class Classification

Prepared by Seetha Parameswaran

1 Objective

The objective is to

- Understand softmax regression using multiple output neurons.
- Implement mini-batch stochastic gradient descent from scratch.
- Classify iris flowers into three species using the Iris dataset.

2 Steps to be Performed

- **Tool:** Python3
- **Libraries required:** numpy, matplotlib, pandas, sklearn
- **Input:** Iris Dataset (from UCI repository)
- **Deep Learning Model:** Softmax Regression
- **ANN Architecture:** Single Neuron (no hidden layers) with Softmax activation function
- **Implementation:** L4-Softmax Regression.ipynb

2.1 Steps

- Import required Python libraries.
- Load the Iris dataset from sklearn.
- Understand the problem: Multi-class classification (3 species).
- Prepare the data: Extract features (X) and labels (y).
- Convert labels to one-hot encoding format.
- Partition the dataset into training (80%) and testing (20%) sets.
- Standardize features using StandardScaler (zero mean, unit variance).
- Create a SoftmaxRegression object with learning rate, epochs, and batch size.
- Train the model using mini-batch stochastic gradient descent.

- Predict class labels and probability distributions for testing set.
- Compute evaluation metrics: Overall accuracy, per-class precision, recall, F1-score.
- Generate and analyze confusion matrix (3×3 for 3 classes).
- Visualize training history (loss and accuracy curves).
- Plot prediction probabilities and decision boundaries.

2.2 Mathematical Formulation

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

$$\text{Softmax Activation:} \quad \hat{y}_k^{(i)} = \text{softmax}(\mathbf{z}^{(i)})_k = \frac{e^{z_k^{(i)}}}{\sum_{j=1}^{K} e^{z_j^{(i)}}}, \tag{2}$$

$$\text{One-Hot Encoding:} \quad \mathbf{y}^{(i)} = [0, \ldots, 1, \ldots, 0]^T \in \{0,1\}^K \tag{3}$$

$$\text{Loss Function (Categorical Cross-Entropy):} \quad J(\mathbf{W}, \mathbf{b}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_k^{(i)} \log(\hat{y}_k^{(i)}) \tag{4}$$

$$\text{Gradients:} \quad \frac{\partial J}{\partial \mathbf{W}} = \frac{1}{N}\mathbf{X}^T(\hat{\mathbf{Y}} - \mathbf{Y}) \tag{5}$$

$$\frac{\partial J}{\partial \mathbf{b}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)}) \tag{6}$$

$$\text{Parameter Update (Mini-batch SGD):} \quad \mathbf{W} := \mathbf{W} - \eta\frac{\partial J_{\mathcal{B}}}{\partial \mathbf{W}} \tag{7}$$

$$\mathbf{b} := \mathbf{b} - \eta\frac{\partial J_{\mathcal{B}}}{\partial \mathbf{b}} \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{d\times K}$ and $\mathbf{b} \in \mathbb{R}^K$ and $\mathcal{B}$ is a mini-batch of size B and $k = 1, 2, \ldots, K$.

3 Results

- Softmax regression model successfully trained using mini-batch SGD.
- Model achieved 96.67% accuracy on test set for iris classification.
- Categorical cross-entropy loss decreased smoothly over 1000 epochs.
- Confusion matrix shows the misclassifications across all three species.
- Softmax activation produces valid probability distributions ($\sum_k \hat{y}_k = 1$).
- Mini-batch training (batch size=16) balanced speed and stability effectively.
- Model provides confidence scores for all three classes simultaneously.
- Decision boundaries successfully separate three linearly separable classes.

4 Observation

- Multiple output neurons (K=3) with softmax enable multi-class classification.
- Softmax couples all class probabilities, unlike independent sigmoid outputs.
- One-hot encoding is essential for categorical cross-entropy loss computation.
- Weight matrix $\mathbf{W} \in \mathbb{R}^{4\times3}$ represents three linear classifiers.
- Mini-batch SGD (batch size 16) faster than batch GD, more stable than single-example SGD.
- Feature scaling critical for multi-class problems with gradient-based optimization.
- Model learned three coupled linear decision boundaries in 4D feature space.
- Iris dataset classes are well-separated, enabling perfect linear classification.
- Softmax naturally extends binary logistic regression to $K > 2$ classes.
- Probability outputs enable ranking alternatives (1st choice, 2nd choice, etc.).