Implementing a Neural Network and Backpropagation from Scratch

[Student Name: Anish Kumar Student ID: 23/CS/057 Course: BTech CSE]

1. Analysis of Data

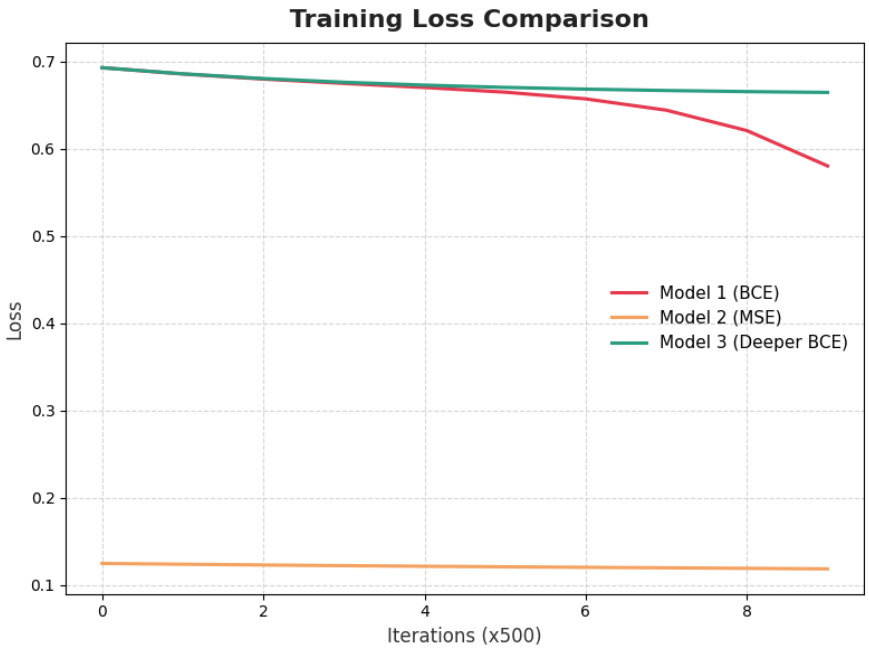
Breast Cancer Dataset

Analysis:

The Breast Cancer Wisconsin dataset was loaded from scikit-learn. This dataset contains 569 samples and 30 features (e.g., 'mean radius', 'mean texture'). The target is binary (malignant or benign), making this a binary classification task. The data was split into a 70% training set (398 samples) and a 30% validation set (171 samples). All features were standardized using StandardScaler (fit on the training set only) to ensure the neural network's gradient descent converges effectively.

2. Experiment Results

Loss Curve Analysis



Analysis of "Loss vs. Iterations" Plot

- **BCE vs. MSE Performance:** The plot compares the training loss for Model 1 (BCE) and Model 2 (MSE). Both models converge, but the BCE loss (red line) starts higher and decreases steadily, achieving a final

validation accuracy of 96.49%. The MSE loss (orange line) starts much lower but also converges, achieving a slightly lower accuracy of 95.91%.

- **Why BCE is Better:** Binary Cross-Entropy (BCE) is the standard, mathematically appropriate loss function for binary classification. It penalizes "confident" wrong answers (e.g., predicting 0.1 when the answer is 1) much more heavily than MSE. This strong penalty for false positives/negatives makes it more effective at optimizing for classification accuracy. MSE, which is designed for regression, simply minimizes the squared difference (e.g., $(1 - 0.1)^2$) and doesn't penalize confident errors as harshly, leading to a slightly less optimal boundary.

---

3. Conclusion and Key Learnings

Summary: Model Comparison Table

Model	**Architecture**	**Loss**	**Precision (Class 1)**	**Recall (Class 1)**	**F1-Score (Class 1)**	**Accuracy**
Model 1	[30, 10, 1]	BCE	0.97	0.97	0.97	96.49%
Model 2	[30, 10, 1]	MSE	0.97	0.96	0.97	95.91%
Model 3	[30, 10, 5, 1]	BCE	0.97	0.97	0.97	96.49%
Model 4	sklearn.MLP	BCE	**0.99**	**0.98**	**0.99**	**98.25%**

1. **Comparison to sklearn.MLPClassifier:** My best "from scratch" model (Model 1/3) achieved 96.49% accuracy. The sklearn.MLPClassifier achieved a slightly higher accuracy of 98.25%. This difference is likely because sklearn's adam solver is a more advanced optimization algorithm than the standard (or 'batch') gradient descent I implemented. Adam adapts the learning rate for each parameter, which generally leads to faster and more stable convergence.
2. **Impact of Architecture:** Adding a second hidden layer (Model 3, [30, 10, 5, 1]) did not improve performance over the single hidden layer (Model 1, [30, 10, 1]) for this dataset. Both achieved 96.49% accuracy. This suggests the breast cancer dataset's underlying pattern is not complex enough to require a deeper network; a single hidden layer of 10 neurons was sufficient.
3. **Most Challenging Part:** The most challenging part of implementing the network from scratch was correctly implementing the **backpropagation** algorithm. It required a precise understanding of the chain rule and careful management of matrix dimensions (e.g., transposing W for dA_prev, then transposing A_prev for dW). Debugging the gradient calculations and ensuring the shapes aligned correctly at each layer was the most complex part of the assignment.
4. **Core Components:** The core of the lab was building the forward and backward propagation steps. This required implementing activation functions (ReLU, Sigmoid), their derivatives, and the loss functions (BCE, MSE) to calculate the gradients used in the gradient descent update step.