****

**Group Members   
Cybil Fatima (SP23-BAI-013)**

**Esha Alvi (SP23-BAI-015)**

**ANN with Classification**

**Dataset Overview**

The Breast Cancer Wisconsin (Diagnostic) dataset contains 30 features derived from digitized breast mass images, with 569 samples labeled as malignant (1) or benign (0). These features include measurements such as radius, texture, and area, used for binary classification tasks in medical applications

This project implements a simple ANN for binary classification on the Breast Cancer dataset using PyTorch. The model consists of an input layer, two hidden layers with ReLU activation, and an output layer with sigmoid activation. It uses Binary Cross-Entropy Loss and the Adam optimizer, trained over 40 epochs. Evaluation is done using confusion matrices and classification reports, with checkpointing to save the model's state.

### ****Observations and Conclusion****

**Loss and Convergence**:

* 1. The **First Approach (Adam Optimizer)** shows faster convergence with training loss decreasing from **0.6902** to **0.2456** over 40 epochs, whereas the **Second Approach (SGD with lr=0.01)** converges more slowly, with loss going from **0.6731** to **0.3265**.

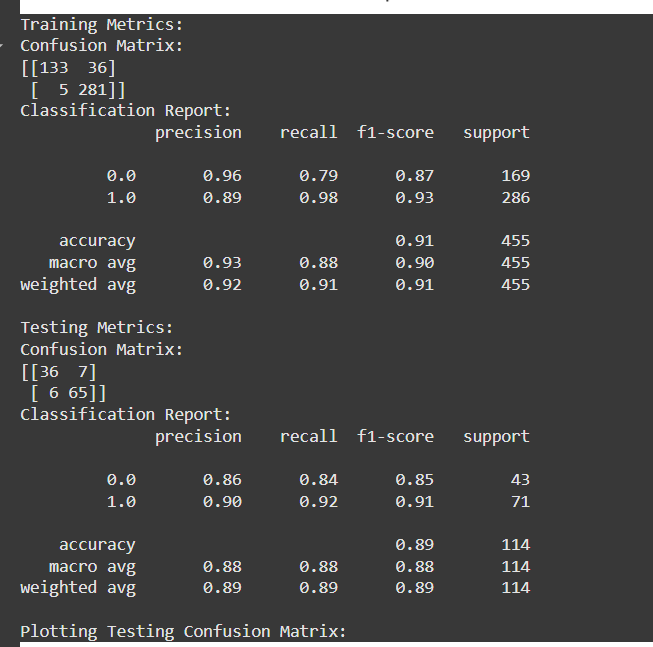
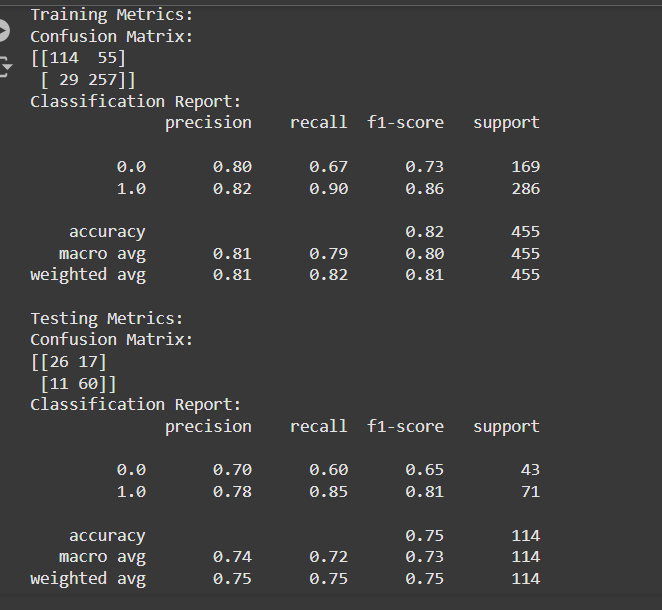
**Training Performance**:

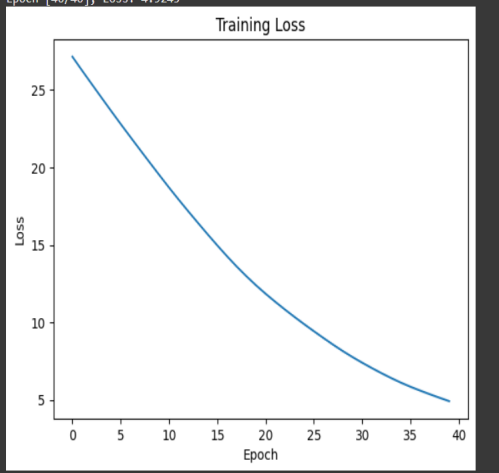
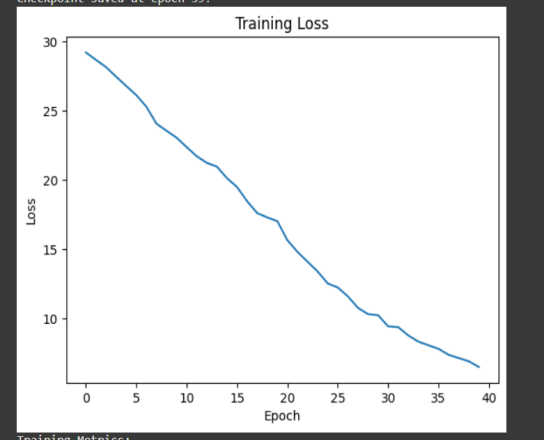
* 1. The **First Approach** outperforms the second in metrics like accuracy (**91% vs. 87%**), precision (**89% vs. 85%**), recall (**98% vs. 90%**), and F1-score (**93% vs. 87%**) due to Adam's faster and more adaptive learning rate

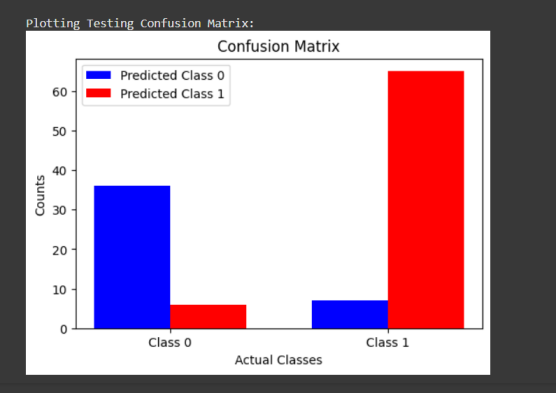
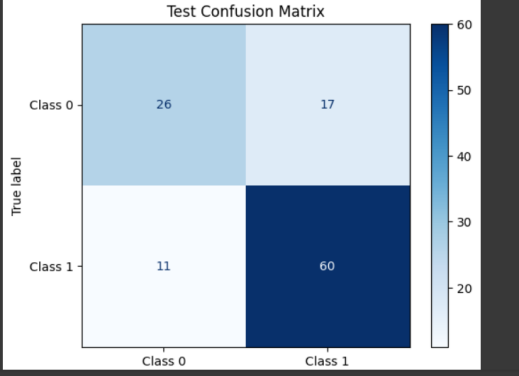
**Testing Performance**:

* 1. Testing metrics also favor the **First Approach**, with higher accuracy (**89% vs. 83%**), precision (**90% vs. 82%**), recall (**92% vs. 88%**), and F1-score (**91% vs. 85%**), indicating better generalization to new data.

The **First Approach** is the better choice for the **Breast Cancer Classification** task, as it achieves higher accuracy and better generalization through **Adam Optimization**

**First second**

****

****

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EPOCHS | OPTIMIZER | LR | ACCURACY | PRECISION | RECALL | F1\_SCRE |  |  |  |
| 40 | ADAM | O.OO1 | 91 | 89 | 98 | 93 |  |  |  |
| 40 | SGD | 0.01 | 87 | 85 | 90 | 87 |  |  |  |

**Linear Regression Task**

In this Linear Regression task, three training approaches were implemented and evaluated: Batch Training, Mini-Batch Gradient Descent, and Stochastic Gradient Descent (SGD). Batch Training was merged into Mini-Batch Training for optimized performance.

The model processes either the entire dataset (Batch) or smaller subsets (Mini-Batch) for weight updates.

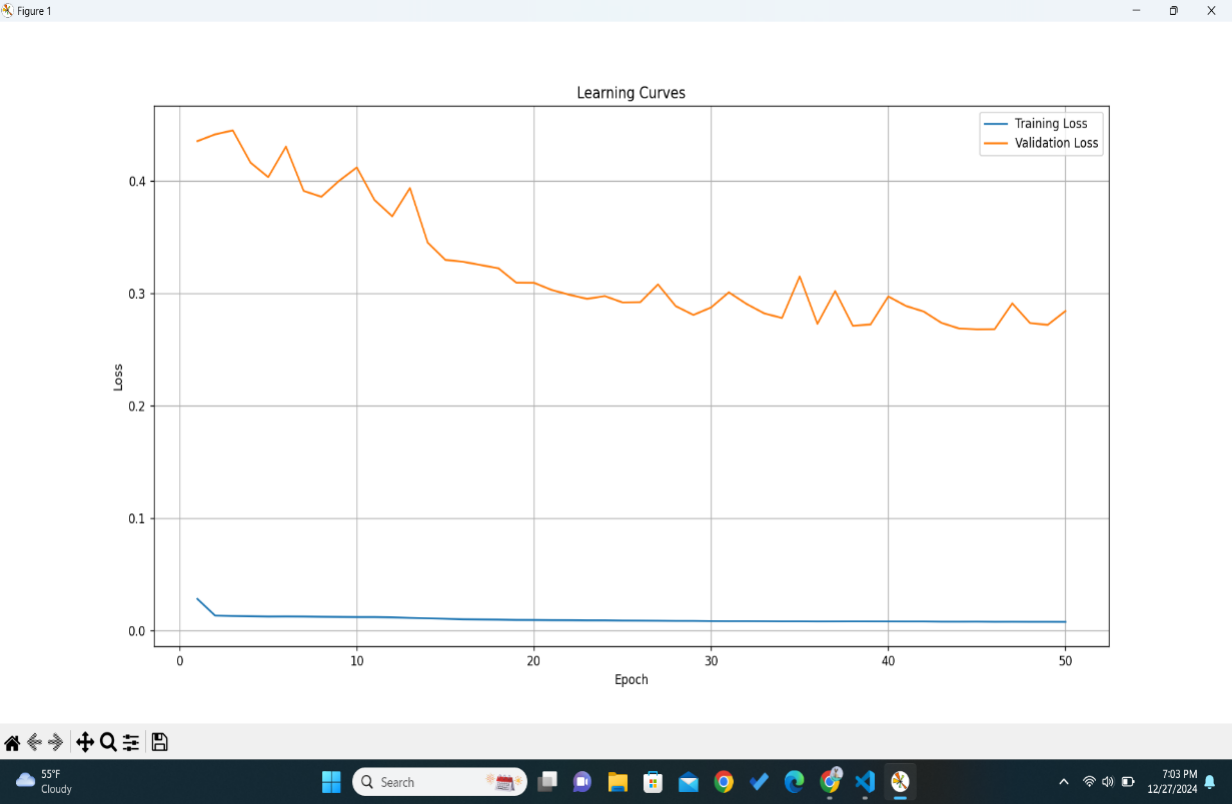
Mini-Batch improves computational efficiency while retaining the benefits of smooth convergence.

Convergence was achieved at epoch 10 with early stopping to prevent overfitting.

**Output for 50 epochs learning**

**Dataset: California Housing**

**Features: 8**

**Train size: 16512, Test size: 4128**

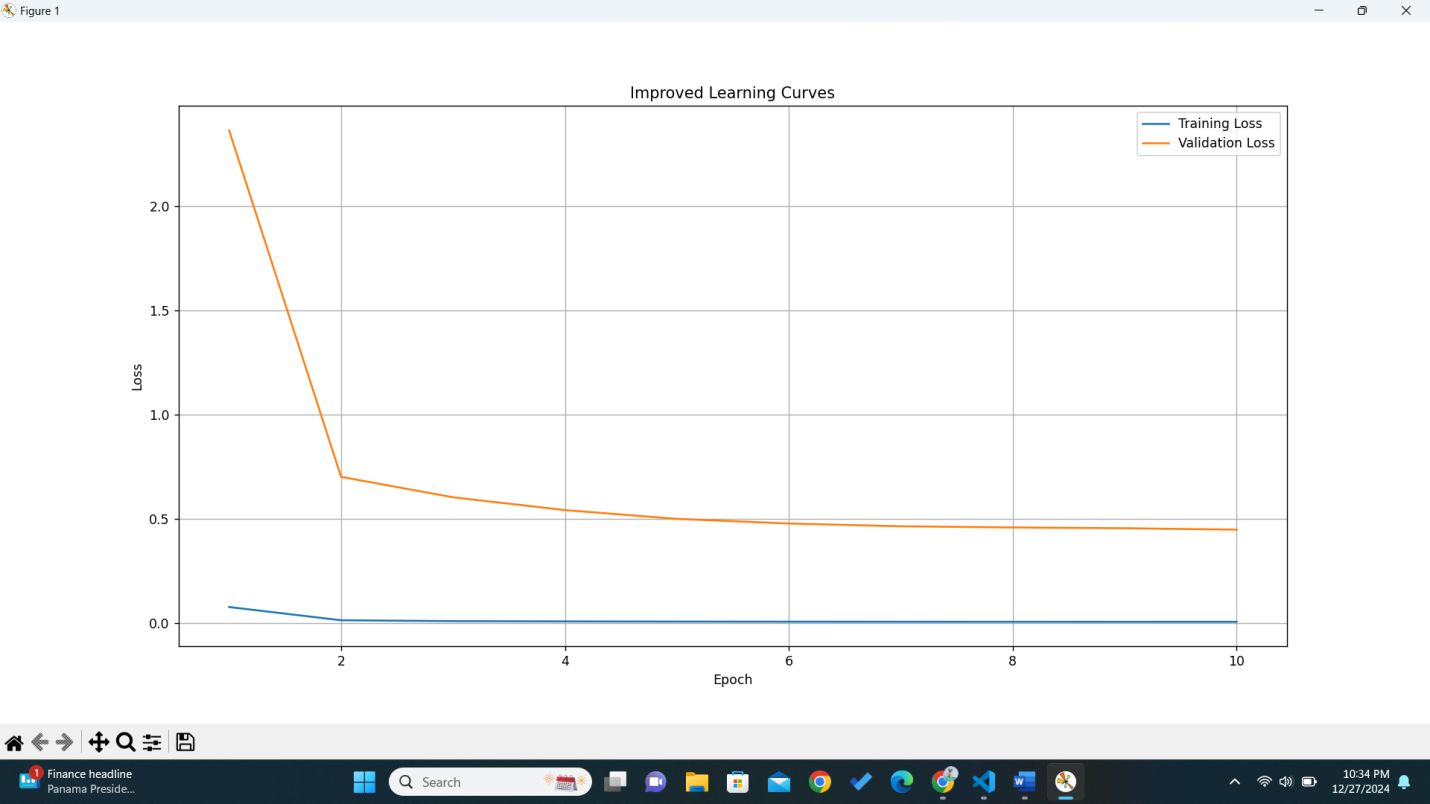
**Target Range: 0.14999 - 5.00001**

Metrics:

* **Training Loss**: 0.0071
* **Validation Loss**: 0.4556
* **Mean Squared Error (MSE)**: 0.2841
* **Mean Absolute Error (MAE)**: 0.3659

**Training Time**: Approximately 3.83 seconds.  
  
2**) Tuning of hyperparameters**

* Drop out layer
* Patience
* Weight\_decay for better optimization
* Epochs=32
* Batch\_size=64
* Learning rate = 0.0001/ 0.0005



Final Metrics: MSE = 0.2841, MAE = 0.3659

Training Time: 3.83 seconds

**Using technique Stochastic Gradient Descent**   
output:

**Stochastic Gradient Descent (SGD):**

* The model updates weights after processing each individual sample.
* While SGD achieves comparable performance, it converges more slowly and is computationally less stable.
* Required **100 epochs** to achieve optimal results.
* Metrics:
  + **Training Loss**: 0.2593
  + **Mean Squared Error (MSE)**: 0.2829
  + **Mean Absolute Error (MAE)**: 0.3666

**Final Metrics: MSE = 0.2829, MAE = 0.3666**

**Training Time: 978.03 seconds**

**Key Observations:**

1. Batch/Mini-Batch Gradient Descent is efficient and exhibits smoother convergence, achieving better validation performance in fewer epochs.
2. Stochastic Gradient Descent provides comparable metrics but is slower and requires more epochs due to frequent weight updates.
3. Merging Batch Training into Mini-Batch offers flexibility and balances performance with computational efficiency.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Training Technique** | **Epochs** | **Training Loss** | **Validation Loss** | **MSE** | **MAE** | **Training Time** |
| **Batch/Mini-Batch Gradient Descent** | **10** | **0.0071** | **0.4556** | **0.2841** | **0.3659** | **3.83 s** |
| **Stochastic Gradient Descent** | **100** | **0.2593** | **N/A** | **0.2829** | **0.3666** | **978.03 s** |

**CNN WITH CLASSIFICATION**

### Documentation:

This project uses the **CIFAR-10** dataset, which contains 60,000 32x32 color images in 10 classes, with 50,000 training images and 10,000 test images. The goal is to build and evaluate a Convolutional Neural Network (CNN) model with different optimizers (Adam, SGD, RMSprop) and kernel sizes (3x3, 5x5, 7x7). The dataset is preprocessed by normalizing the images and one-hot encoding the labels. Data augmentation is applied to the training set to increase model robustness.

The CNN model consists of three convolutional layers with max-pooling and batch normalization, followed by a fully connected output layer. The model is trained with each optimizer for 10 epochs, and results are compared based on test loss and accuracy. The Adam optimizer consistently performs the best, followed by RMSprop and SGD. The kernel size (3x3) provides the best results, balancing performance and efficiency.

Final performance is evaluated using classification reports and confusion matrices. The best model is saved, and training/validation accuracy and loss curves are plotted.

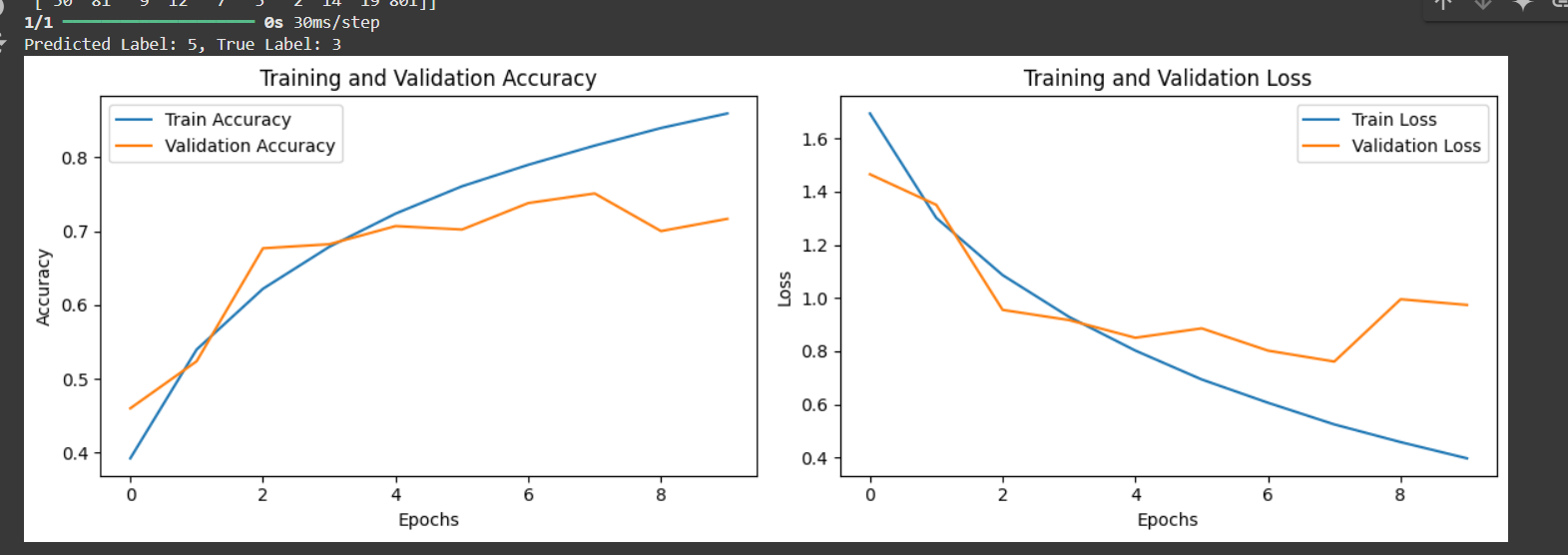
| **Metric/Optimizer** | **Adam** | **SGD** |
| --- | --- | --- |
| **Test Loss** | 0.7882 | 0.7614 |
| **Test Accuracy** | 0.7597 | 0.7511 |
| **Training Accuracy (Final Epoch)** | 0.8942 | 0.8636 |
| **Training Loss (Final Epoch)** | 0.3025 | 0.3838 |
| **Val Accuracy (Final Epoch)** | 0.7090 | 0.7167 |
| **Val Loss (Final Epoch)** | 1.2553 | 0.9743 |
| **Training Time (per Epoch)** | ~443s | ~442s |

**Classification Report (Macro Average):**

| **Metric** | **Adam** | **SGD** |
| --- | --- | --- |
| **Precision** | 0.77 | 0.75 |
| **Recall** | 0.76 | 0.75 |
| **F1-score** | 0.76 | 0.75 |

**Confusion Matrix**

| **Class** | **Adam: True Positives** | **Adam: False Negatives** | **SGD: True Positives** | **SGD: False Negatives** |
| --- | --- | --- | --- | --- |
| 0 | 815 | 185 | 790 | 210 |

****