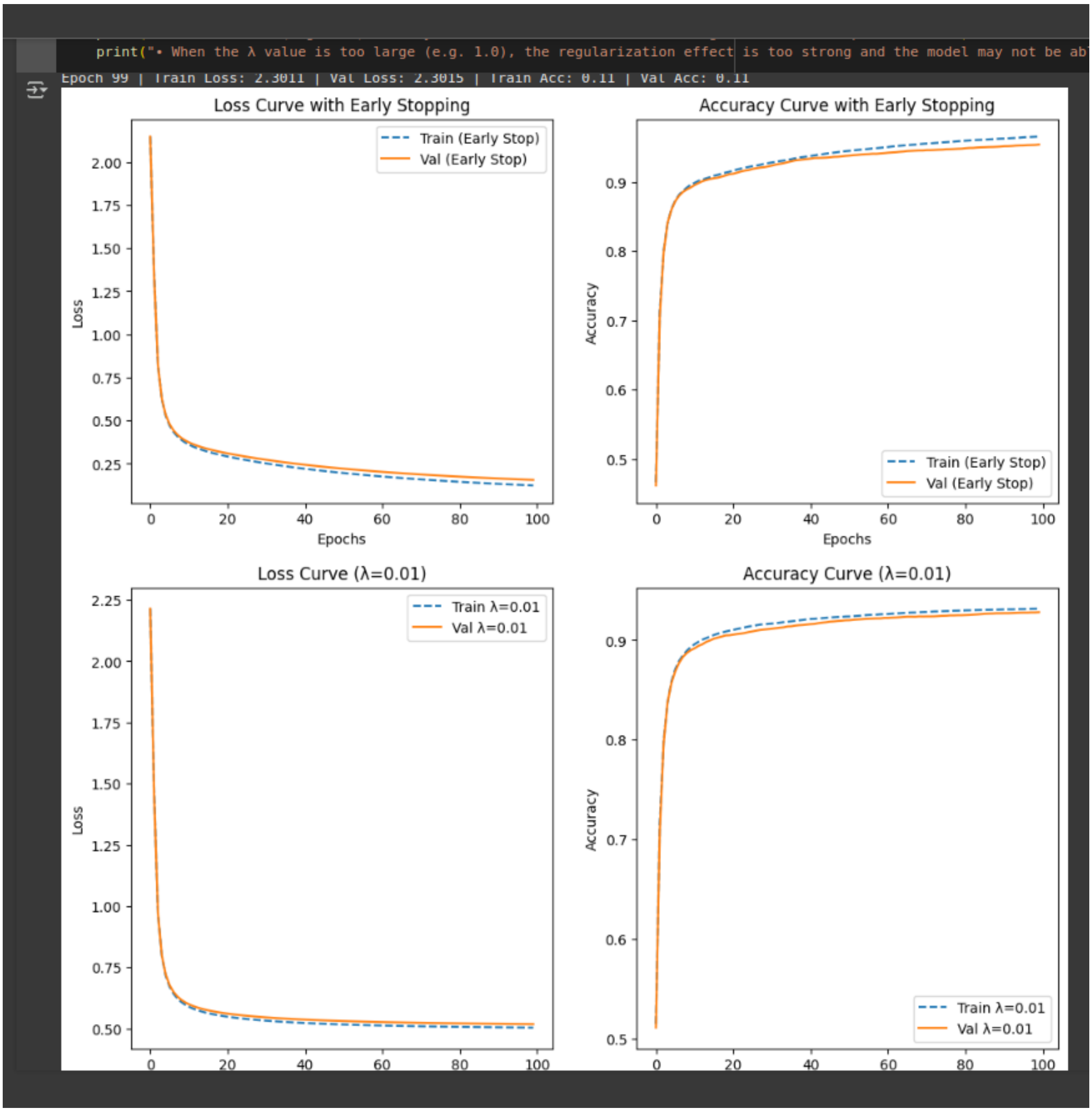
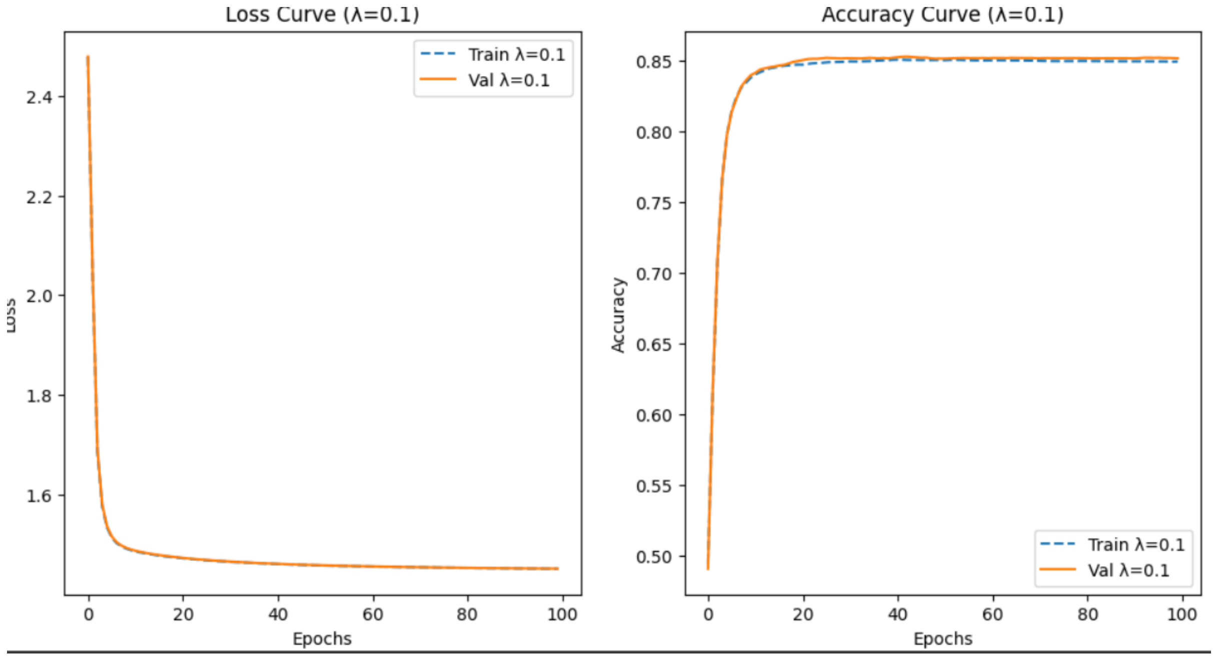
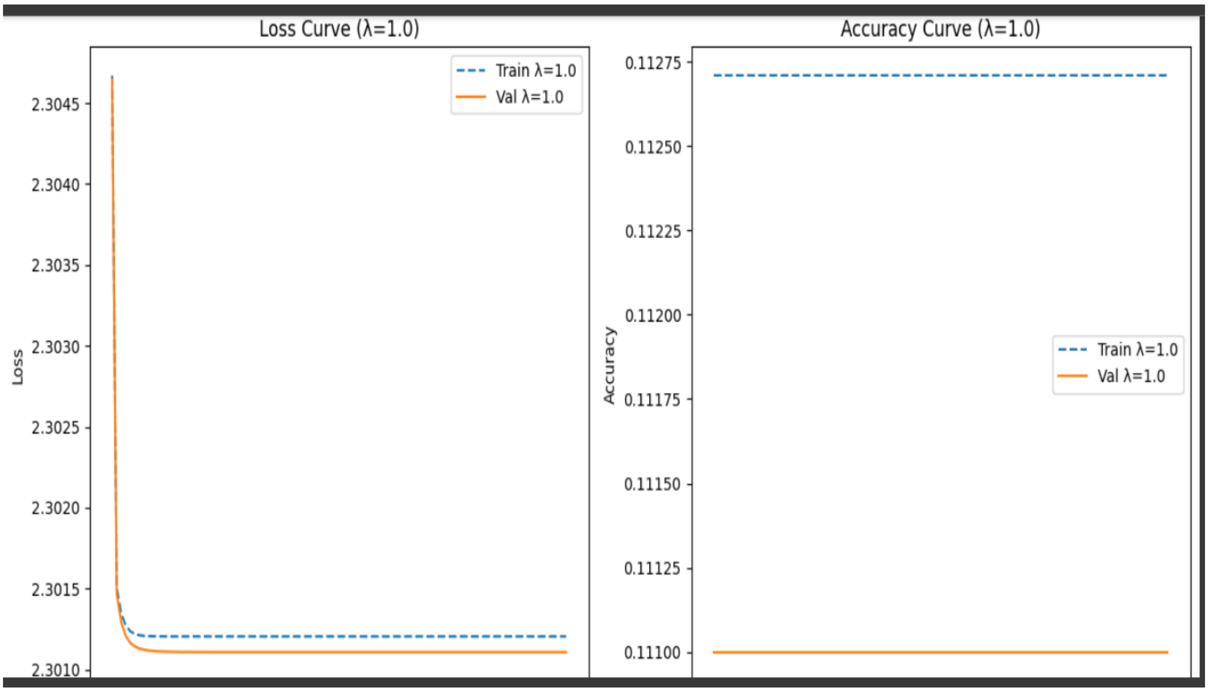
Lab5

code results:




Loss Curve (λ=0.1)
Train λ=0.1
Val λ=0.1
Loss
Epochs
Accuracy Curve (λ=0.1)
Accuracy
Epochs


Loss Curve (λ=1.0)
Train λ=1.0
Val λ=1.0
Loss
Accuracy Curve (λ=1.0)
Accuracy

Question:

1. Which regularization method gave you the best test accuracy? Why do you think it performed better than the other? Was it due to training duration, generalization effect, or another factor?

The **Early Stopping** method gave the best test accuracy among all regularization strategies. This is evident from the accuracy curve with early stopping, where both training and validation accuracies converge and reach above **95%**, outperforming all L2 regularization (weight decay) settings.

The reason it performed better is mainly due to its **generalization effect —** it stops training when the validation performance stops improving, thus preventing overfitting while allowing enough training time to learn meaningful patterns. It also avoids unnecessary over-training, which might otherwise lead to memorization of the training set.

Additionally, early stopping adaptively controls training duration based on actual performance, unlike fixed epochs in weight decay experiments. This flexibility gives it an advantage, especially when the validation metric starts to plateau.

2. Compare training and validation loss curves Which method showed signs of overfitting or underfitting? Use your graphs to justify your answer (e.g., early stopping curve flattens early, weight decay trains longer but smoother).

Early Stopping:

- The training and validation loss curves are very close and both decrease smoothly, with no significant gap, which **indicates neither overfitting nor underfitting**.
- Accuracy curves also converge closely, suggesting a well-generalized model.

- **L2 Regularization (λ = 0.01):**
    - Training and validation losses are slightly separated, especially in accuracy curves — training accuracy is higher than validation accuracy, which **suggests mild overfitting**.
    - However, the gap is small and performance is still good overall.
- **L2 Regularization (λ = 0.1):**
    - The gap between training and validation loss/accuracy is minimal. Both curves converge smoothly, showing **neither overfitting nor underfitting**.
    - However, overall accuracy is slightly lower than λ = 0.01, suggesting that the regularization is slightly stronger and might limit the model's capacity.
- **L2 Regularization (λ = 1.0):**
    - Both training and validation losses stay high and flat, and accuracy remains low (around **11%**, close to random guessing).

- This is a clear case of **underfitting**. The regularization is too strong, preventing the model from learning useful representations.

3. How did your choice of regularization strength (λ) or patience affect the model? What λ or patience value worked best in your experiment? What happened when you increased or decreased it?

Best Regularization Strategy:

Early Stopping worked best, with a patience value that allowed the model to train long enough to learn patterns but stopped before overfitting. This balance made it more adaptive than fixed λ values.

Effect of λ (L2 regularization):

- **λ = 0.01**: Gave good performance; slight overfitting but acceptable.
- **λ = 0.1**: Reduced overfitting and gave stable results, but slightly lower accuracy.
- **λ = 1.0**: Regularization was too strong, leading to **severe underfitting**. The model could not learn and performed poorly.

Conclusion:

Small λ (like 0.01) allows the model to learn effectively but may slightly overfit.

Moderate λ (0.1) improves generalization with a slight drop in accuracy.

Large λ (1.0) overly penalizes weights, preventing learning entirely.

For this task, **early stopping with a well-tuned patience value** was the most effective approach.