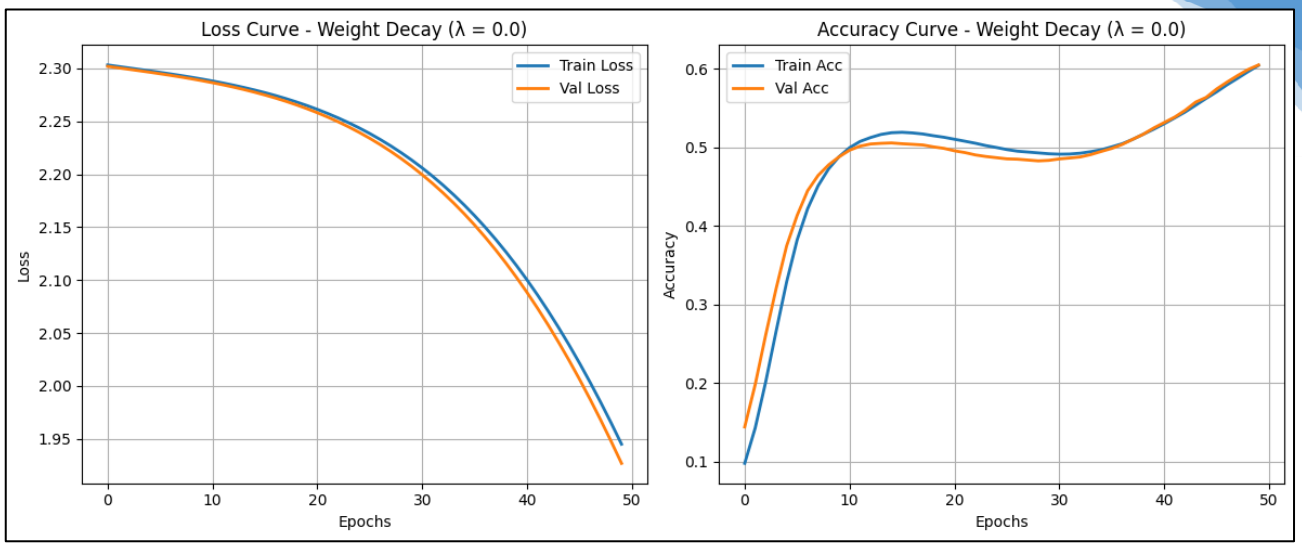
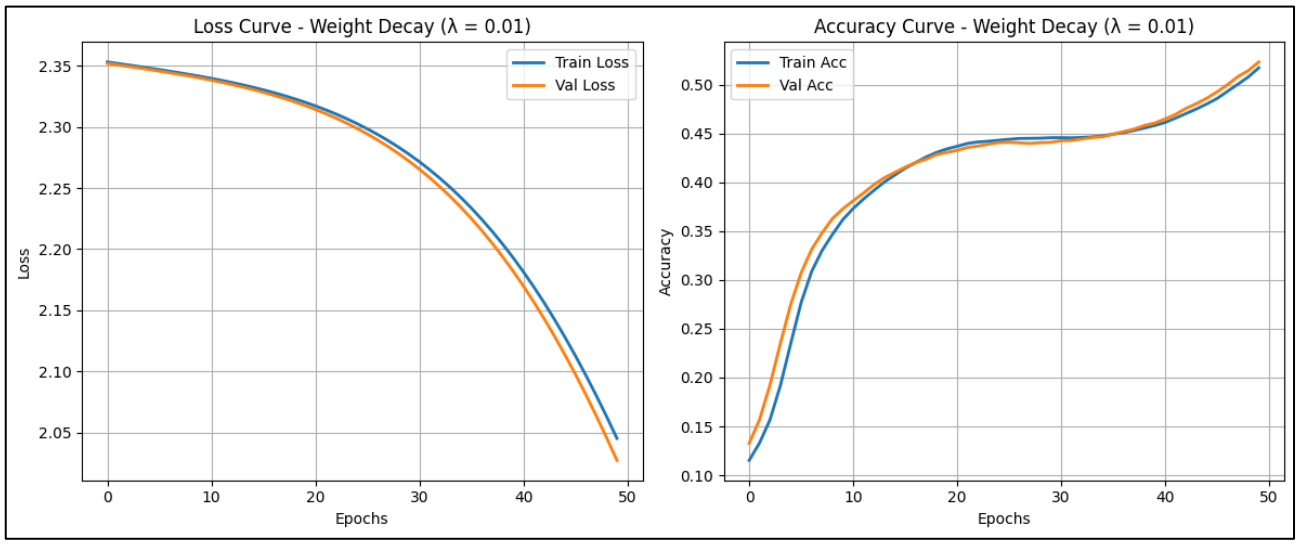
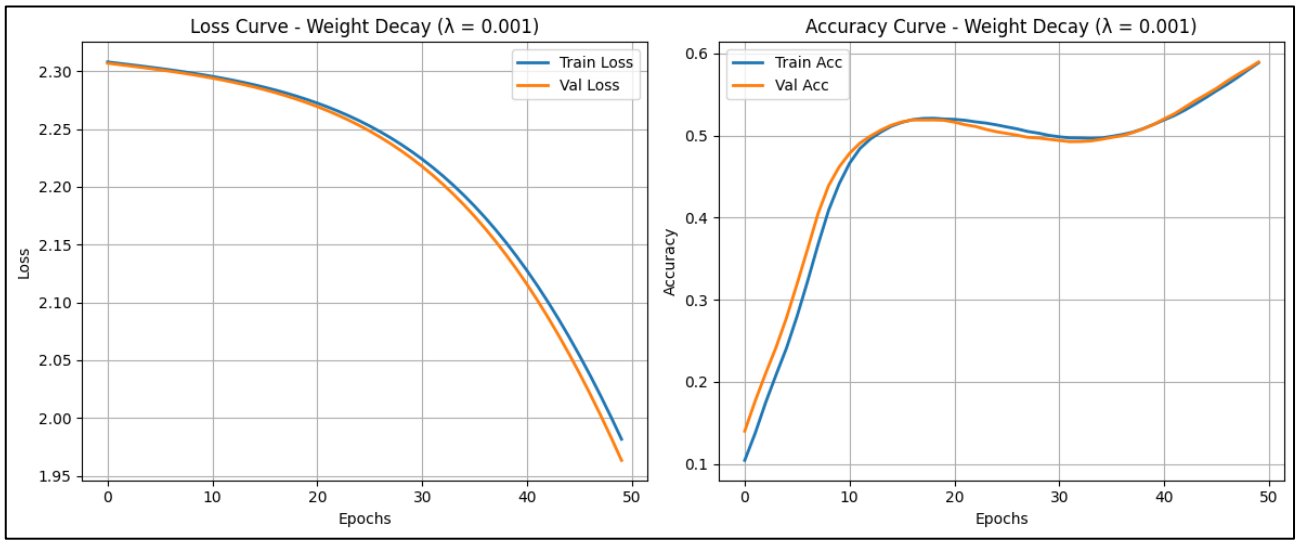
Put Your Code Results Here:







Lecture: Prof. Hsien-I Lin
TA: Satrio Sanjaya and Muhammad Ahsan

Question (20%):

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?

Look at the image at last page, Early Stopping and Weight Decay (with λ = 0.0, 0.01 and 0.001) didn't make a huge difference in final validation accuracy. But Weight Decay with λ = 0.001 did a little better overall.

Its validation accuracy go up more clearly later in training and keep getting better, ending up a bit higher than the others.

Why? Weight Decay helps keep the model's weights smaller, which means it doesn't try to make super complicated decisions. That helps it do better on new data, not just the training stuff. Also, using λ = 0.001 is kind of the sweet spot—it stops the model from overfitting, but it doesn't make it too simple either.

2. (7%) 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:
From the chart, Early Stopping didn't actually kick in during the 50 epochs, which means the model didn't start overfitting. The training and validation loss curves went down together almost the whole time and kept dropping, so the model wasn't done learning yet. It might even still be underfitting a little.

Weight Decay (λ = 0.0):
This is basically no regularization at all. The training and validation loss curves looked really close and kept going down. There's no sign of overfitting here either, and the model might still be underfitting just like before.

Weight Decay (λ = 0.01, λ = 0.001):
The curves for these look super smooth and also kept going down. The training and validation lines matched up really well, and sometimes the validation loss was even lower than the training one. That's another sign the model was still underfitting, not overfitting.

So all 4 experiment show that the model hasn't fully finished learning yet — it's still kind of underfitting. That probably means we could make the model a bit more complex or train it for more epochs to get better accuracy.

3. (6%) **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?

Weight Decay (how different λ values affect things):

- **λ = 0.0:**
  There's no regularization here, so the model acts pretty much like the baseline.
- **λ = 0.001:**

This one gives just the right amount of regularization. It ended up doing the best — it helps stop overfitting but still lets the model learn well.

- **λ = 0.01:**
  This one was a bit too much. The regularization was too strong, so the model couldn't learn as fast, and its final accuracy wasn't as good as with λ = 0.001.

So yeah, **λ = 0.001** is the sweet spot. It strikes a nice balance between keeping the model simple and letting it learn well.

Early Stopping (how the patience setting matters):

- The patience was set to 5, but Early Stopping never actually kicked in. That means 5 was big enough to let the model keep training all the way through 50 epochs without stopping early.
- If we had set patience lower (like 2 or 3), it might've stopped too early, before the model was fully trained.
- Setting it higher wouldn't have changed much in this case (since Early Stopping didn't trigger), but in real life, it could just waste time by training longer than needed.

So best is keep patience around 5 — it gives the model enough time to learn while also helping avoid super long training.