

Presentation 4

- Links to presentation(s) and code(s) on GitHub
 - [Presentation](#)
 - [Code](#)
- What did you do?
 - I modified the training pipeline by adding several new functionalities to improve model robustness and convergence. First, I incorporated a learning rate scheduler that automatically reduces the learning rate by a factor of 0.3 whenever the validation loss plateaus, allowing the model to escape flat regions and refine convergence. I also implemented early stopping, which halts training when validation loss fails to improve for 8 epochs, helping prevent overfitting. In addition, I added dropout, which randomly deactivates neurons during training to enhance generalization and reduce reliance on specific features. Finally, I applied a stronger weight decay (L2 regularization) to further constrain model complexity and serve a similar regularization role as dropout. After integrating these improvements, I tested the updated training pipeline and observed generally favorable results.
- How does it help the project?
 - Successfully improved the robustness of the training pipeline by reducing chances of overfitting. Provided a framework for better convergence, but the current limited number of observations (only 51 cases in train and 17 cases in validation) is also another significant factor for wobbly validation loss. I believe that the current framework is widely used in deep learning and would show huge potentials if we get more cases.
- Issues faced (if any)
 - I encountered an issue where the validation loss did not accurately reflect the model's true performance. During the early stages of training, the model was able to achieve a low validation loss, but this did not translate into strong performance on the test set. In fact, later epochs (despite having higher validation losses) produced better and more reliable test results. This suggests that validation loss alone may not be a sufficient signal for selecting the best checkpoint in this setting.
- Attempts to resolve issues (if any)

- o To better assess generalizability and detect possible overfitting, the team began running cross-validation. I was not directly involved in this step, but it provides a clearer picture of the model's true performance beyond a single validation split. In future iterations, we may also introduce hyperparameter tuning using tools such as Optuna.
- Issues resolved (if any)
 - o Fixed the issues discussed during the last presentation.
- Next steps
 - o Optuna hyperparameter tuning
- References (Mention if you built up on someone else's work)
 - o ChatGPT & Claude (coding)