Module 4 - Critical Thinking

Miles Exner

Colorado State University Global

Course Code: CS580-1

Instructor: Bingdong Li

Due Date: 4/13/2025

# Toxicology Testing

To start, we refactored the TensorFlow 1.x code to TensorFlow 2.x with its Keras API, which provided a cleaner implementation for this toxicity prediction task. Through systematic experimentation with different neural network architectures and training approaches, we found a tradeoff between precision and recall when giving more weight to the toxic components. In different use cases, we may want to prioritize recall over precision or vice versa.

In a pharmaceutical safety context, prioritizing recall would be critical as missing a toxic compound (false negative) could have severe health consequences during drug development. In such cases, we would accept more false positives to ensure potentially harmful compounds are flagged. Conversely, in early-stage drug discovery, where thousands of compounds are being screened, prioritizing precision would be more valuable to avoid unnecessarily eliminating promising candidates due to false toxicity predictions. Our models and threshold optimization approach allow stakeholders to adjust this balance according to their risk tolerance and application requirements.

Our baseline model featuring a single hidden layer with 50 neurons achieved high accuracy (96.04%) but suffered from poor recall (9.68%), indicating it missed most toxic compounds despite reasonable precision (50%). We found that increasing the layer width to 100 neurons while strengthening regularization with dropout (0.3) significantly improved precision to 66.67%, with a modest increase in recall to 12.90%, demonstrating the benefit of additional model capacity for capturing complex molecular feature relationships.

Our third approach addressed class imbalance by applying aggressive class weighting (10x for toxic compounds) and implementing early stopping, which doubled recall to 25.81% but reduced precision to 50%, highlighting the fundamental precision-recall tradeoff in toxicity classification.

Our most successful approach combined moderate class weighting (5x) with decision threshold optimization. Rather than using the default classification threshold of 0.5, we systematically evaluated thresholds between 0.10 and 0.85, discovering that 0.40 provided the optimal balance with 70.59% precision and 38.71% recall, yielding an F1 score of 0.50—significantly higher than previous configurations. This threshold analysis revealed that different classification thresholds could be selected depending on the specific application: safety-critical contexts might prioritize recall (using lower thresholds around 0.15-0.20), while discovery applications concerned with minimizing false positives could use higher thresholds (0.60-0.75) to maintain precision above 70%.

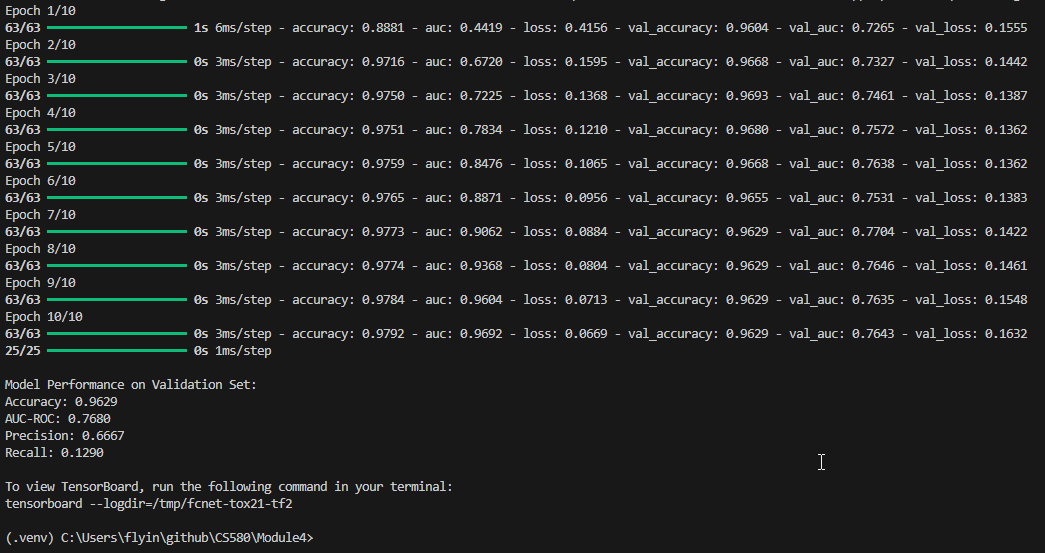
## Conclusion

Our experiments demonstrate that accurate toxicity prediction requires careful consideration of model architecture, class imbalance handling, and classification thresholds. While accuracy remained consistently high across all models due to class imbalance in the dataset, metrics like precision, recall, F1 score, and AUC-ROC (which reached 0.7856 in our final model) provided a more meaningful performance assessment. For real-world applications, our analysis suggests that a single hidden layer with increased width (100 neurons), moderate class weighting (5x), appropriate regularization (dropout 0.3), and a custom classification threshold (0.40) provides the most balanced approach for toxicity prediction. However, stakeholders can adjust the threshold based on their tolerance for false positives versus false negatives.

## Output

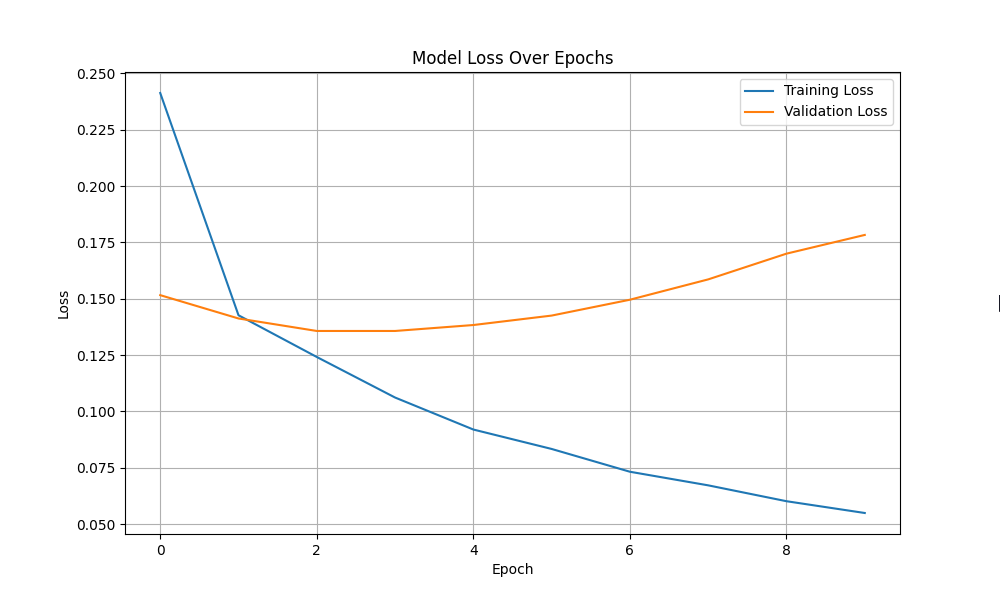
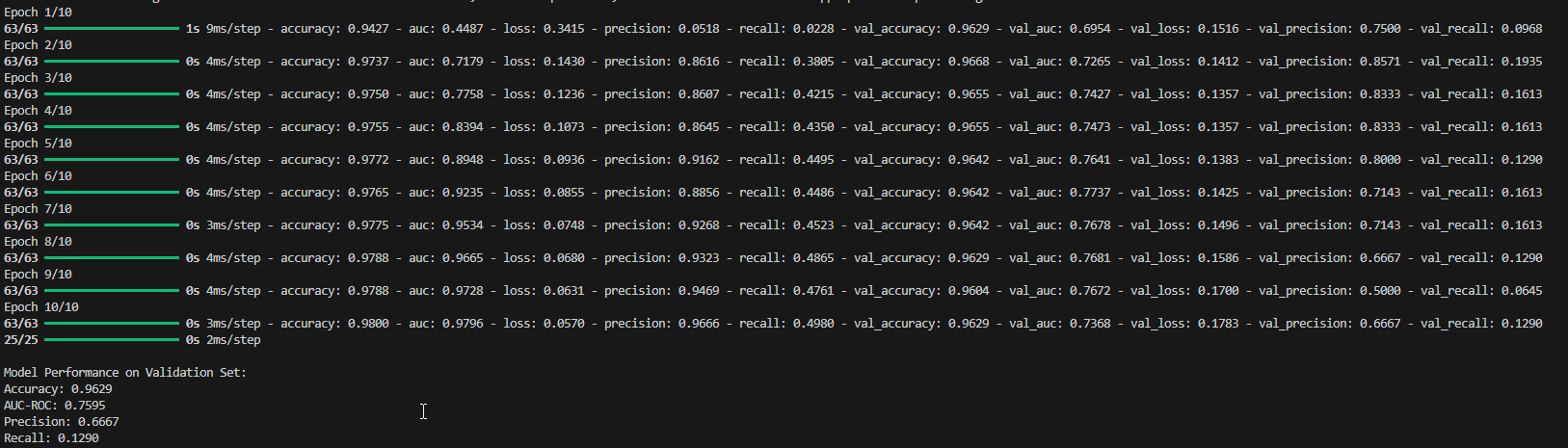
**Run 1 - Original Model**: 10 epochs, 1 hidden layer with 50 neurons, dropout 0.2

* **Results**: Accuracy: 0.9604, AUC-ROC: 0.7720, Precision: 0.5000, Recall: 0.0968
* **Performance**: Accuracy: 0.9604, AUC-ROC: 0.7720, Precision: 0.5000, Recall: 0.0968
* **F1 Score**: 0.1623



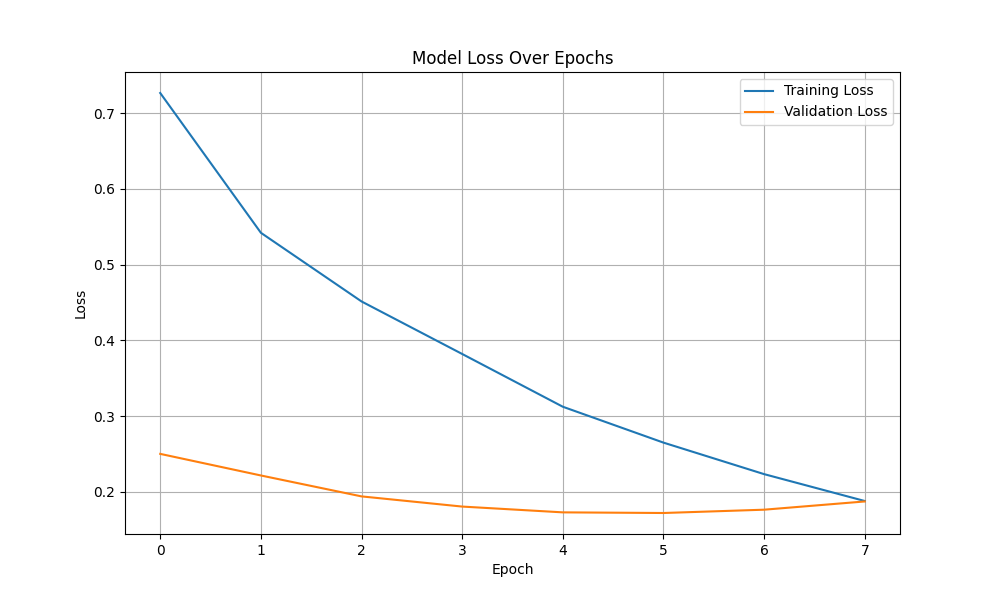
**Run 2 -**  **Increased Width**: 10 epochs, 1 hidden layer with 100 neurons, dropout 0.3

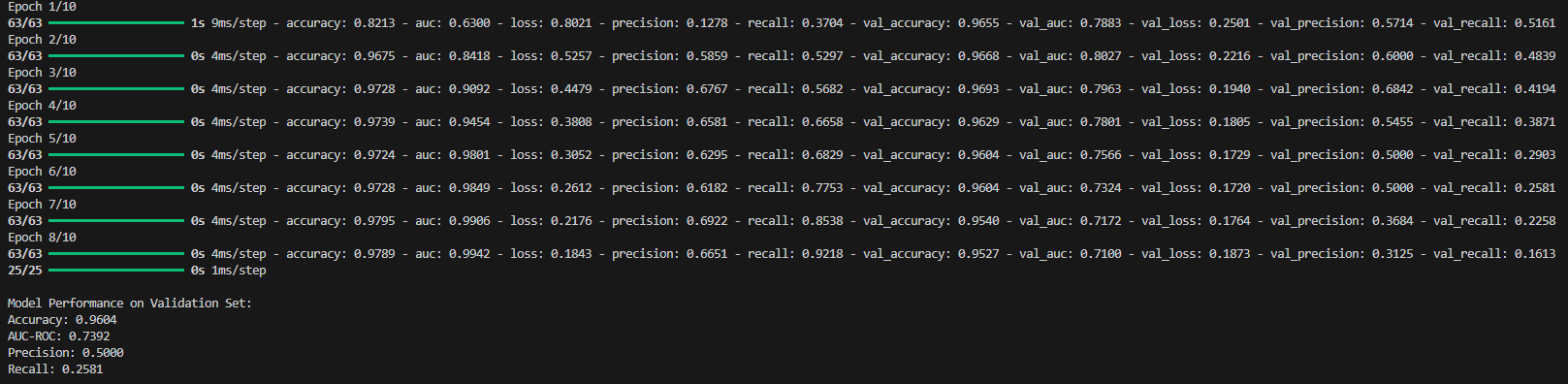
* **Results**: Accuracy: 0.9629, AUC-ROC: 0.7395, Precision: 0.6667, Recall: 0.1290
* **Performance**: Accuracy: 0.9629, AUC-ROC: 0.7395, Precision: 0.6667, Recall: 0.1290
* **F1 Score**: 0.2162

**Run 3 - Class Weights & Early Stopping**: Early stopped at 8 epochs, 1 hidden layer with 100 neurons, dropout 0.3, toxic class weighted 10x

* **Results**: Accuracy: 0.9604, AUC-ROC: 0.7392, Precision: 0.5000, Recall: 0.2581
* **Performance**: Accuracy: 0.9604, AUC-ROC: 0.7392, Precision: 0.5000, Recall: 0.2581
* **F1 Score**: 0.3400

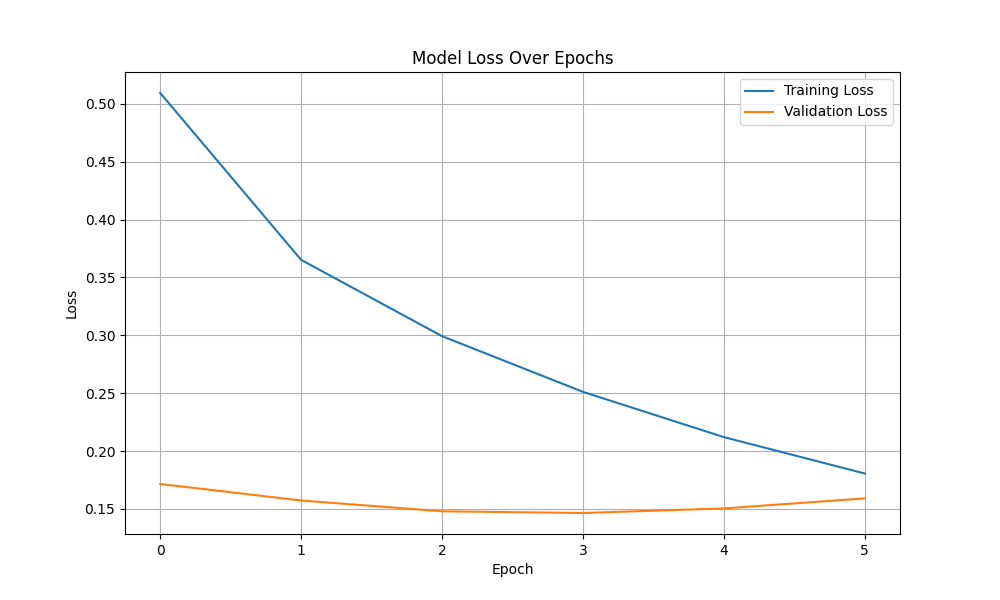


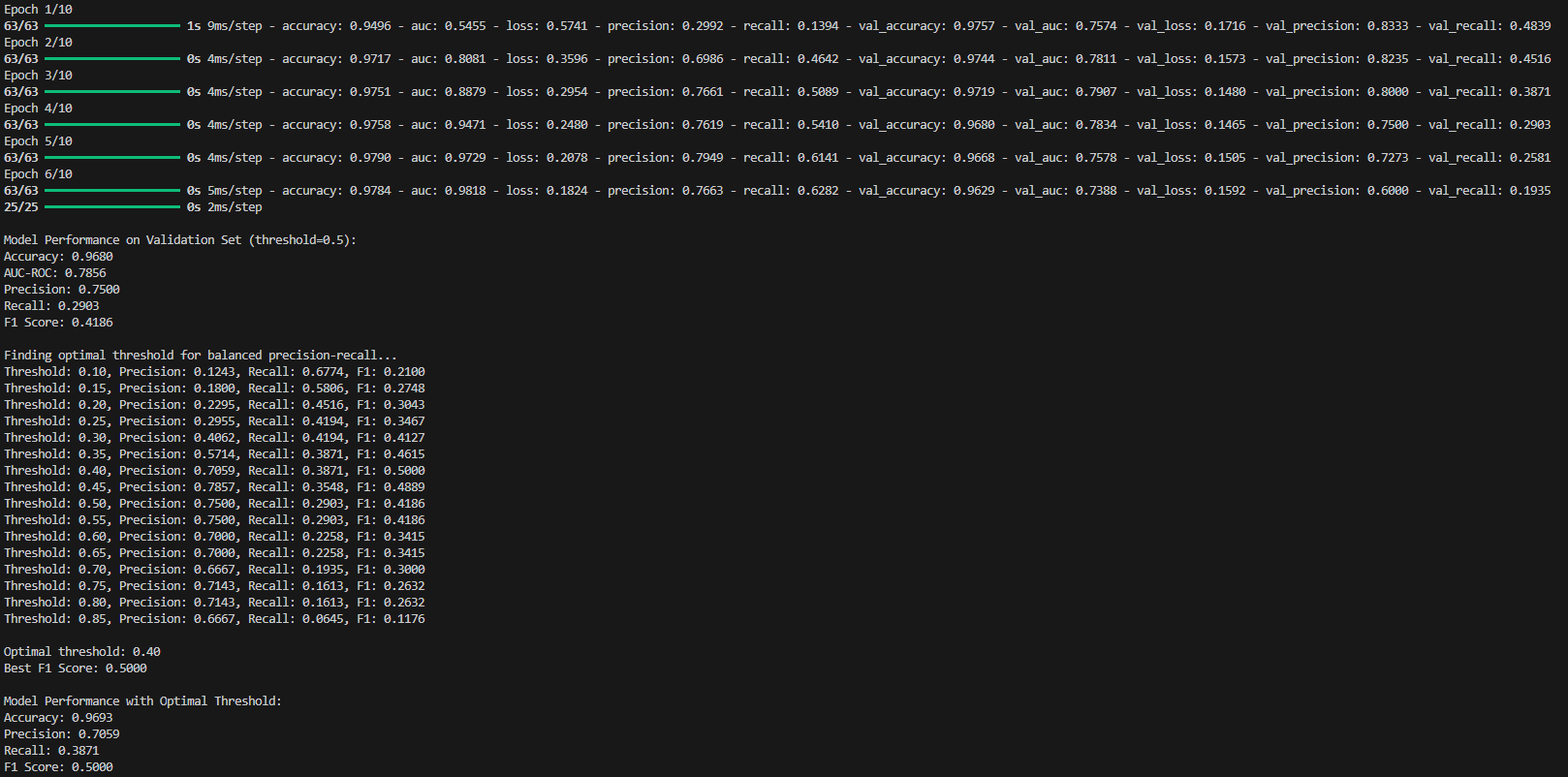


**Run 4 - Balanced Approach with Threshold Optimization**

* **Architecture**: Early stopped at 6 epochs, 1 hidden layer with 100 neurons, dropout 0.3, toxic class weighted 5x
* **Default Performance**: Accuracy: 0.9680, AUC-ROC: 0.7856, Precision: 0.7500, Recall: 0.2903, F1 Score: 0.4186
* **Optimized Performance** (threshold=0.40): Accuracy: 0.9693, Precision: 0.7059, Recall: 0.3871, F1 Score: 0.5000

This balanced approach with moderate class weighting and threshold optimization achieved the best overall performance. By finding the optimal decision threshold (0.40 instead of the default 0.50), we significantly improved the F1 score, providing the best balance between precision and recall.

****

****