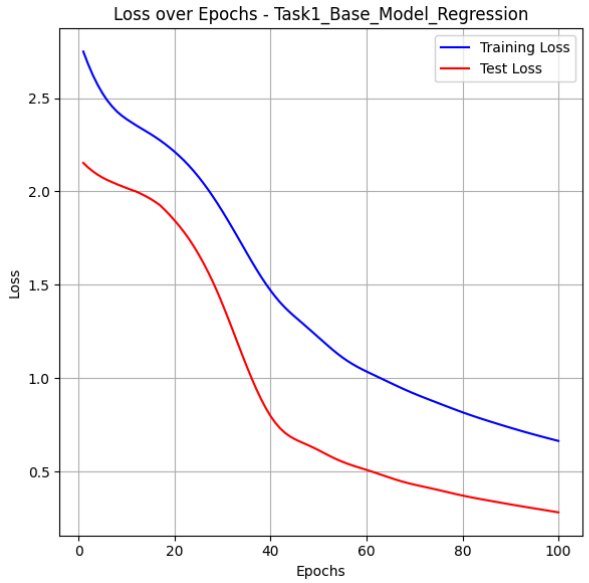
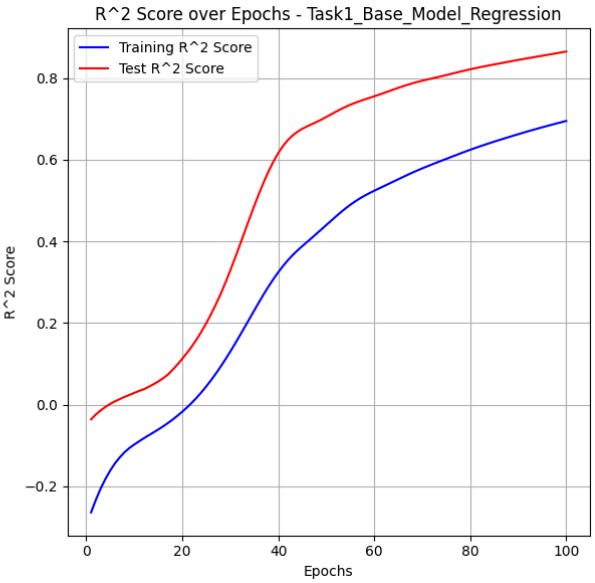
DL - LAB 1

Analysis of the obtained results

Task 1:

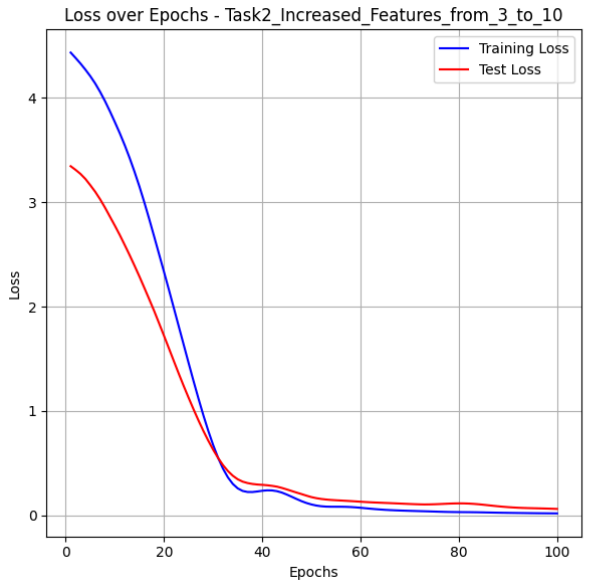


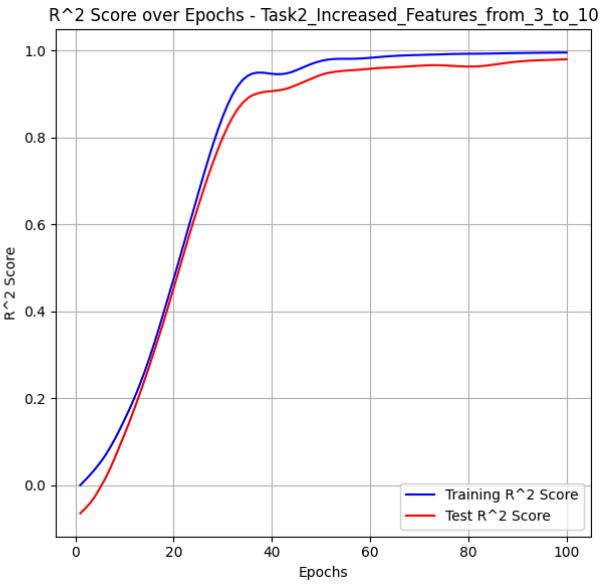


- The significant gap between training and test metrics indicates clear overfitting.
- The model achieves R^2 of 0.6951 on training data but 0.8650 on test data, which is unusual as typically we see better performance on training data than test data.
- This suggests the test set may be inherently easier to predict than the training set, possibly due to the specific data split with seed 42 or characteristics of the synthetic data generation process

Task 2:

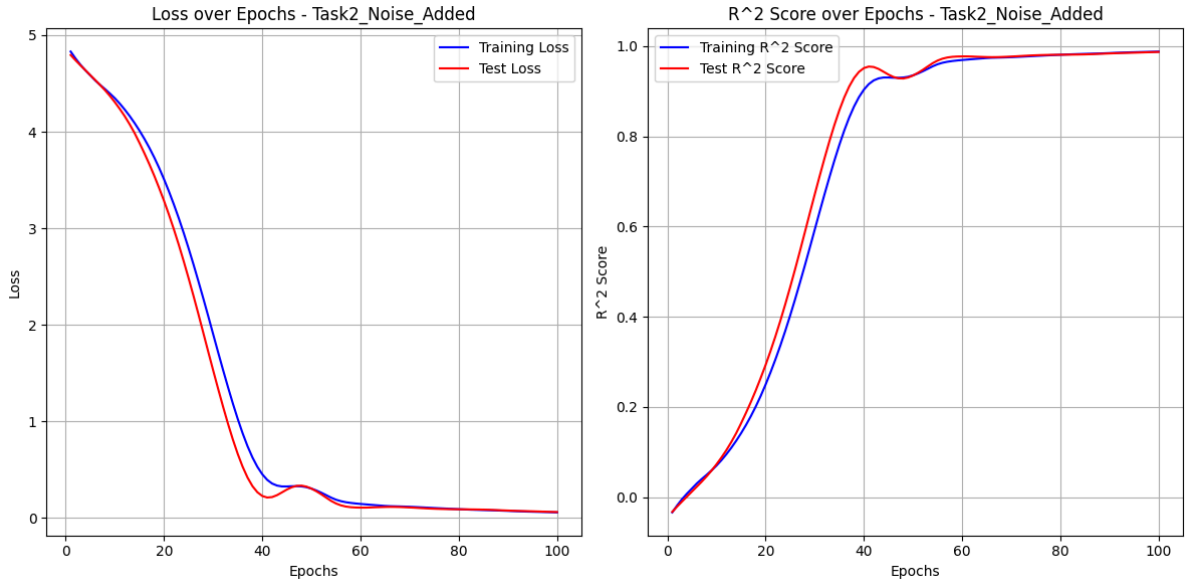
A. Feature increase (3 to 10):





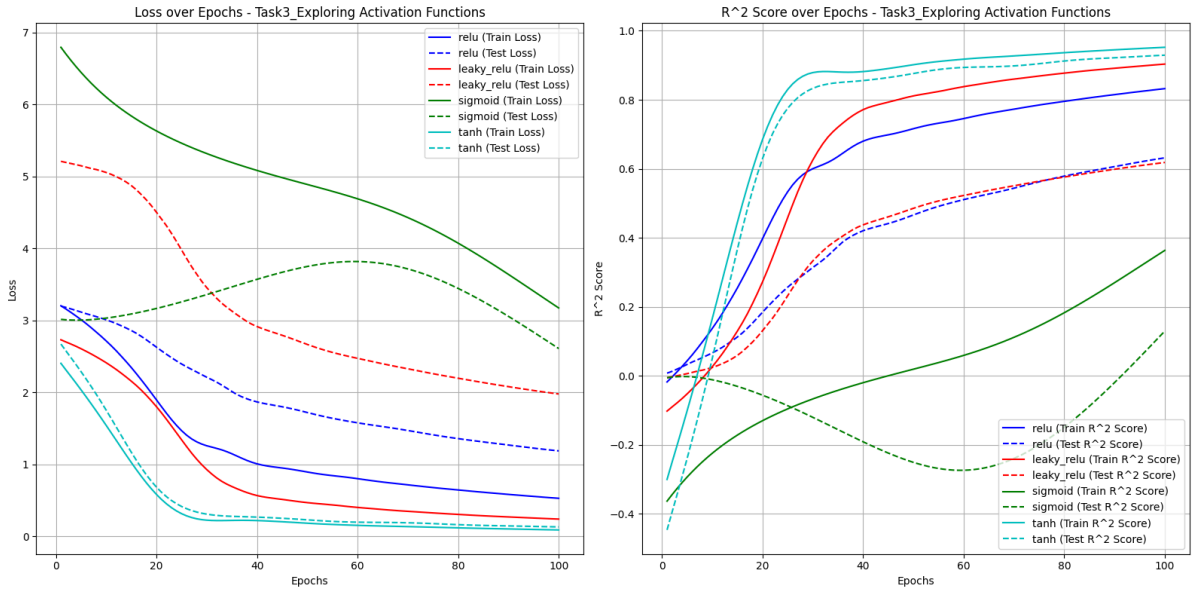
- Increasing features from 3 to 10 allows the model to capture more complex relationships, resulting in significantly better performance
- Final metrics improved dramatically: R^2 increased from ~0.69/0.86 to ~0.99/0.98 (train/test)
- The model converges much faster, reaching $R^2 > 0.9$ by epoch 40 compared to the base model which never reaches this level
- The close alignment between training and test curves indicates good generalization without overfitting
- The final loss decreased by an order of magnitude (from ~0.66/0.28 to ~0.02/0.06)

B. Feature increase + Noise effect (0.1 standard deviation):



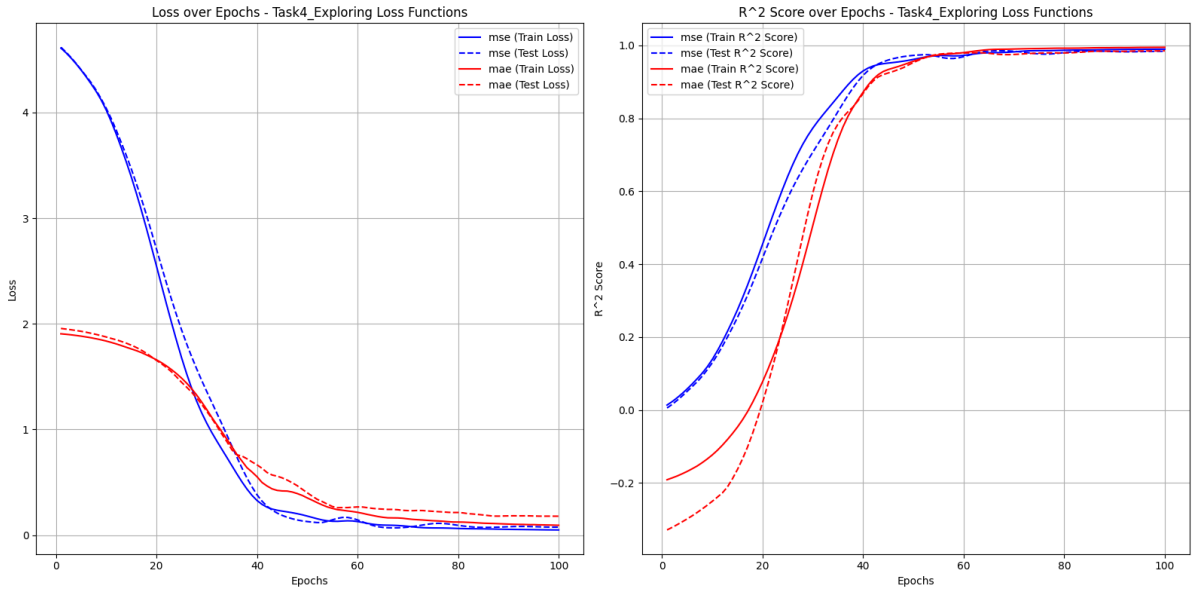
- Adding noise (0.1 standard deviation) slightly slows down initial learning speed
- Despite noise, the model still achieves excellent final performance (R^2 ~0.99/0.99)
- The training and test curves remain closely aligned, suggesting the model effectively learns to ignore the noise
- The final performance is comparable to the noise-free case, demonstrating the model's robustness

Task 3:



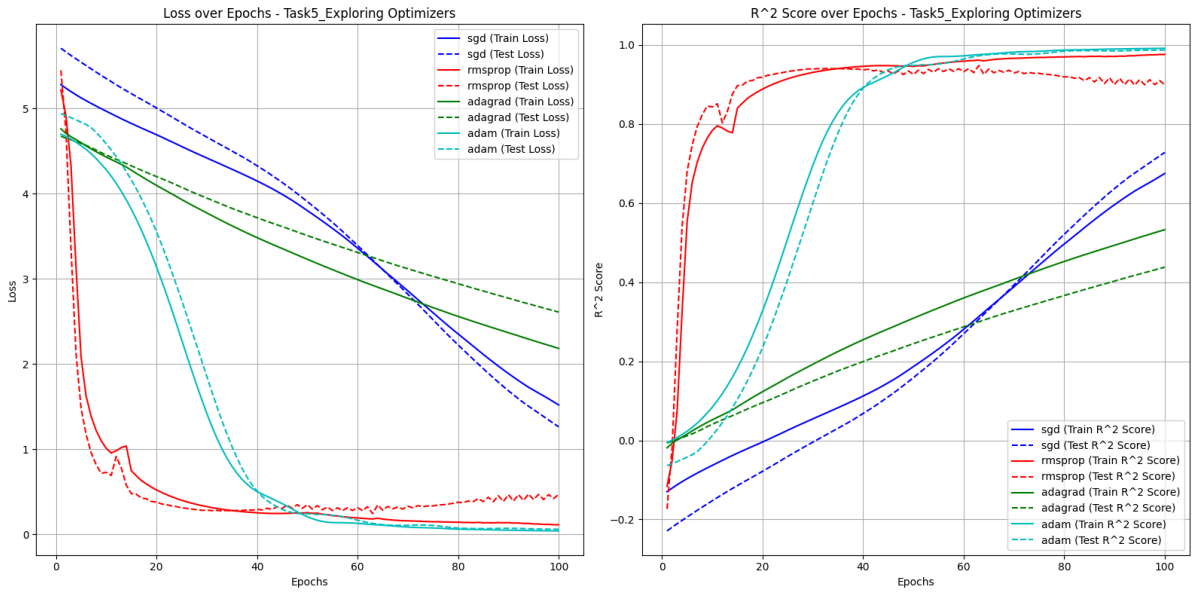
- **ReLU vs Leaky ReLU:**
    - Both achieve strong performance, with Leaky ReLU slightly outperforming ReLU (Test Loss: 0.1183 vs. 0.0943; R^2: 0.9745 vs. 0.9797).
    - The improvement is modest but consistent, likely due to Leaky ReLU's ability to maintain small gradients for negative inputs.
- **Sigmoid**:
    - Significantly worse performance (Test Loss: 0.9785; R^2: 0.7893), showing both higher loss and lower R^2 scores.
    - The graph shows sigmoid struggling with very slow convergence and poor final performance. This aligns with theoretical expectations of vanishing gradients due to saturation in the limited output range [0,1].
- **Tanh**:
    - Performs better than Sigmoid but still underperforms compared to ReLU variants (Test Loss: 0.3442; R^2: 0.9259).
    - The learning curve shows faster initial convergence than sigmoid but plateaus earlier than ReLU variants. This is consistent with tanh addressing some sigmoid limitations through its zero-centered output range [-1,1], but still suffering from saturation effects.

Task 4:



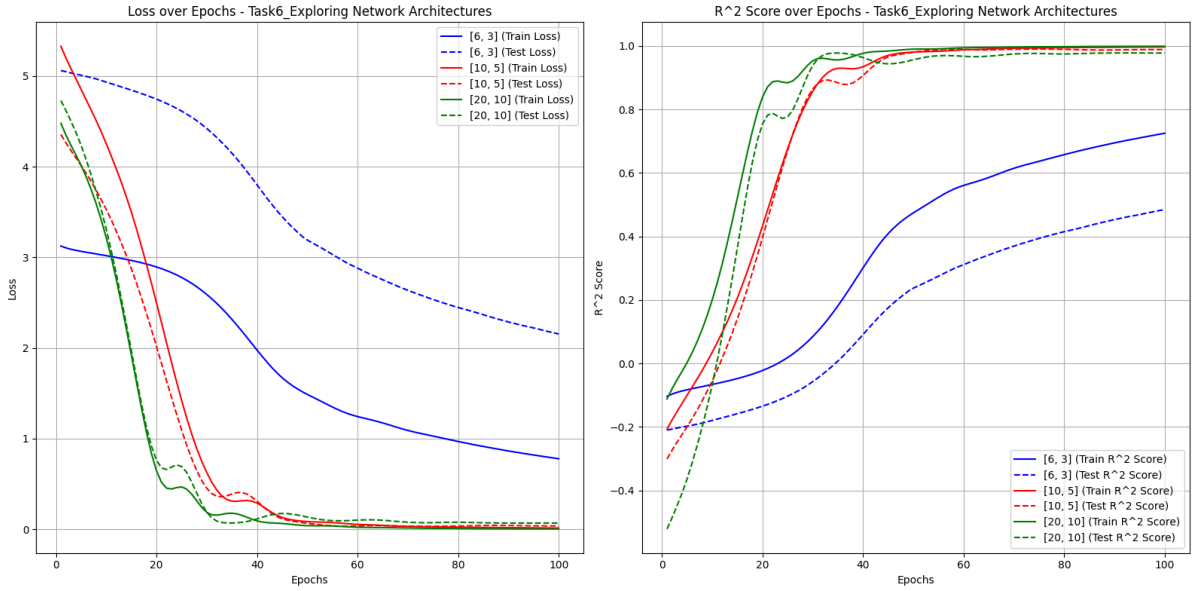
- Both converge to similar final R^2 scores (~0.98-0.99), but MAE achieves lower final loss values (0.08/0.14 vs 0.58/0.65 train/test).
- MSE shows faster initial convergence in the first 20 epochs, but MAE catches up and surpasses MSE around epoch 40.
- MAE training shows more stable progression with smoother curves, while MSE exhibits more fluctuations, particularly in the test metrics.
- MAE demonstrates better generalization with test R^2 occasionally exceeding training R^2, while MSE maintains a more typical gap between training and test performance.
- MAE is less influenced by outliers and provides more stable gradients, while MSE penalizes larger errors more heavily, leading to potentially more aggressive updates.

Task 5:



- **Adam** -> Demonstrates the best balance of speed and stability, achieving the highest final performance (R^2 ~0.99/0.99)
- **RMSprop** -> Shows rapid initial convergence but exhibits instability in later epochs with slight performance degradation
- **SGD** -> Slowest convergence but steady improvement throughout training
- **AdaGrad** -> Shows diminishing learning rates over time, resulting in incomplete convergence

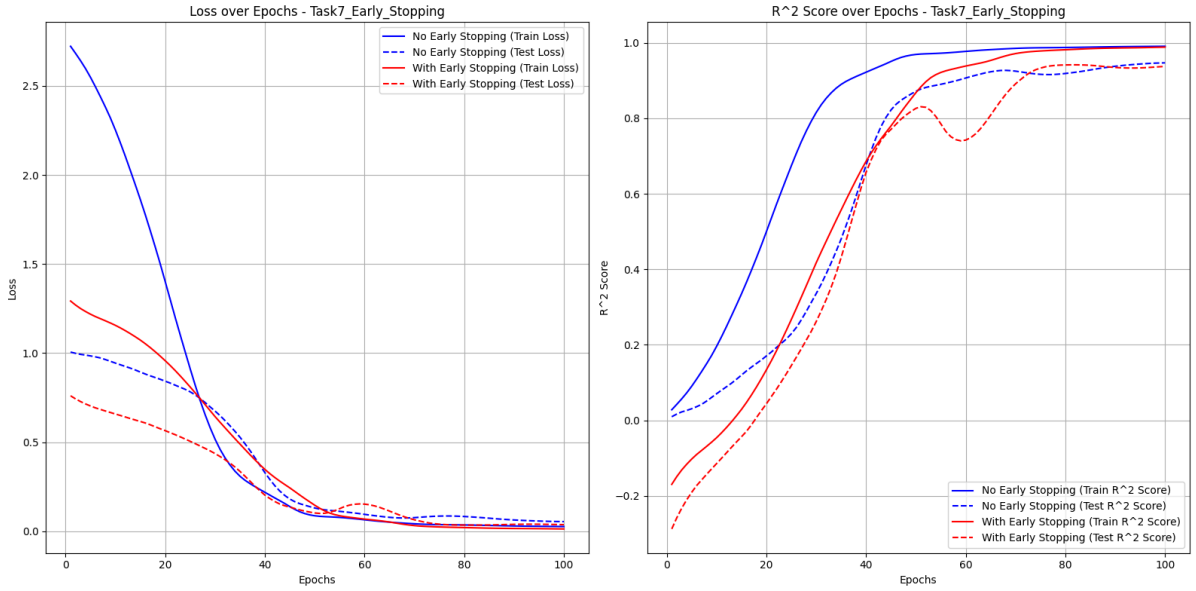
Task 6:



- [6,3] (original model):
    - Achieves moderate performance (R^2 ~0.73/0.48) but shows a significant gap between training and test performance, suggesting limited capacity to capture the underlying patterns.
    - The learning curve shows slow, steady improvement but never reaches the performance of wider networks.
- [10,5] (wider model):
    - Shows dramatically faster convergence and much better final performance (R^2 ~0.997/0.989).
    - The learning curves indicate this architecture has sufficient capacity to model the data effectively, with minimal gap between training and test performance after convergence.
- [20,10] (much wider model):
    - Demonstrates the fastest convergence of all architectures, reaching $R^2 > 0.9$ by epoch 30, and achieves the best final performance (R^2 ~0.999/0.978).

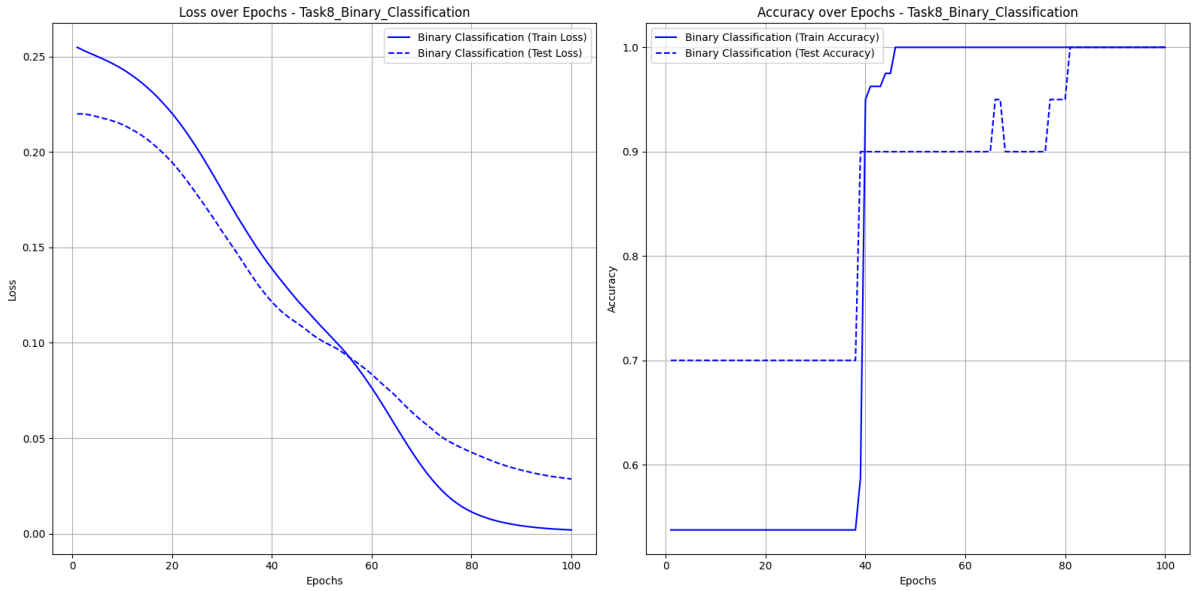
- However, it shows a slightly larger gap between training and test metrics than the [10,5] model, potentially indicating the beginning of overfitting due to excessive capacity.

Task 7:



- Without early stopping
    - The model continues training for all 100 epochs, gradually improving throughout.
    - The final performance is strong (R^2 ~0.99/0.95), but the extended training comes with computational costs.
- With early stopping:
    - Training terminates earlier when validation performance plateaus.
    - The learning curves show comparable final performance to the full training, but with fewer epochs (likely around 60-70 based on the graph).
- Both approaches show similar learning trajectories initially, but the early stopping model shows a temporary dip in test R^2 around epoch 50-60 before recovering. This highlights how early stopping can sometimes prevent the model from working through temporary plateaus.

Task 8:



- The model starts with modest accuracy (~54%/70% train/test) and dramatically improves around epoch 40, achieving 95-100% accuracy on training data and 90% on test data.
- By epoch 81, the model achieves 100% accuracy on both training and test sets, indicating complete separation of the classes.
- Loss continues to decrease substantially even after accuracy reaches 100%, dropping from ~0.08 at epoch 60 to ~0.002 by epoch 100. This indicates the model is increasing its confidence (margin) in predictions even after achieving perfect classification.
- Test accuracy initially exceeds training accuracy (70% vs 54% at epoch 20), suggesting the test set might contain more easily separable examples. The gap closes as training progresses.
- The step-like pattern in the accuracy curve (particularly visible in test accuracy) suggests the model makes discrete improvements as it learns to correctly classify specific challenging examples.