

SML assignment 4 Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>

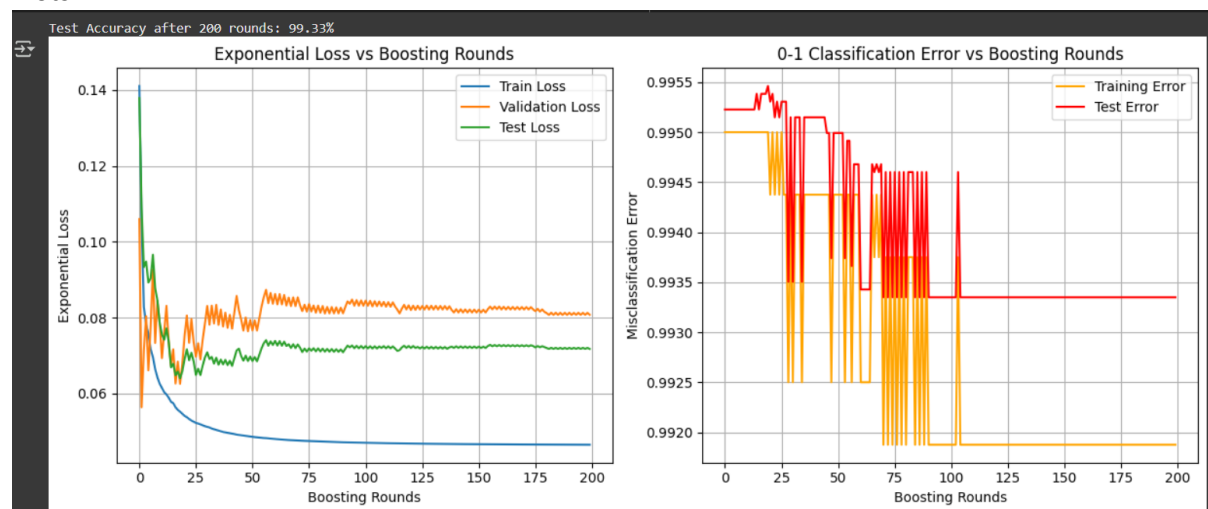
Q1

Test Accuracy after 200 rounds: 99.33%

loss calculation for each round:

Metric	Formula
Exponential Loss	$\sum e^{-y_i f(x_i)}$
0-1 Classification Error	$\frac{1}{N} \sum \mathbf{1}(y_i \neq \text{sign}(f(x_i)))$

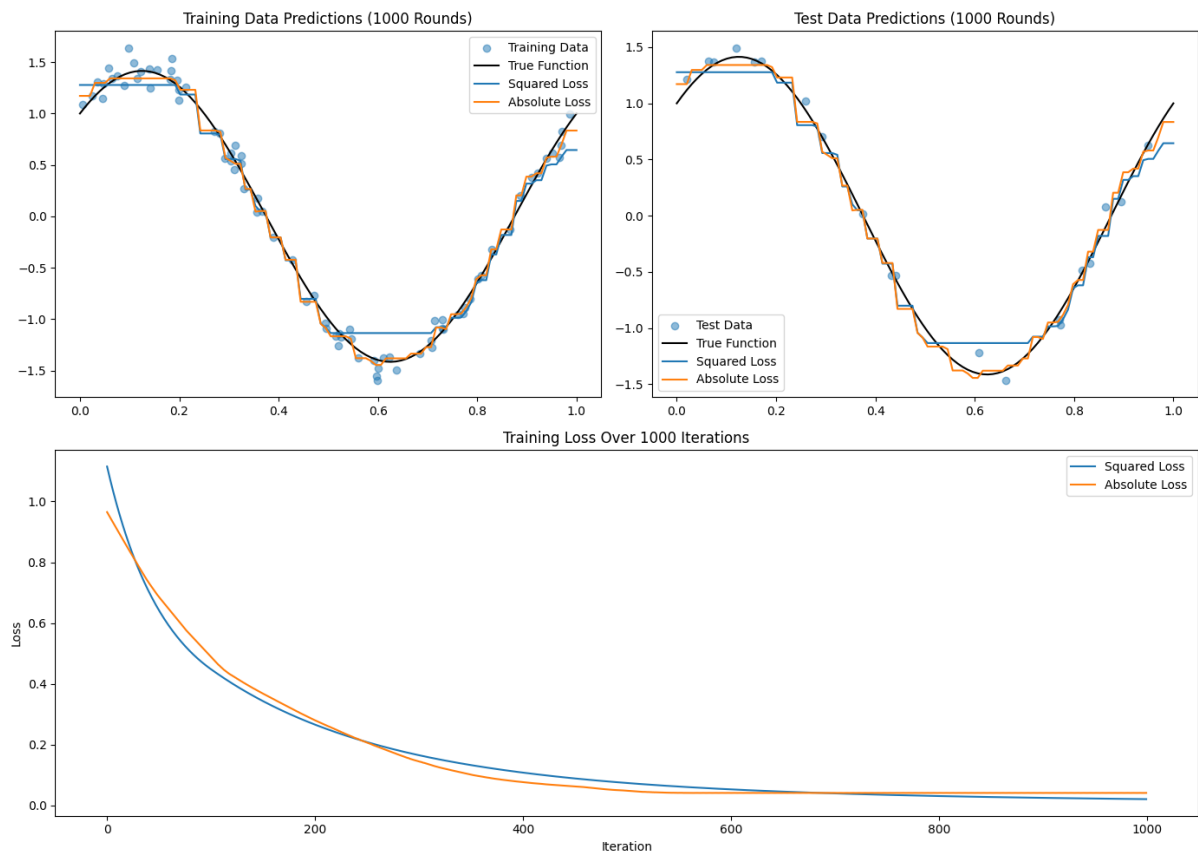
Plots:



Interpretation:

- Validation/test loss decrease and then increase means model is overfitting
- even when classification 0-1 error plateaus the exponential loss keeps on decreasing, that is better margins and more confident predictions.
- The gap between correctly classified and decision boundary keeps on increasing with iteration.

Q2



### Interpretation: Absolute Loss vs Squared Loss (1000 Rounds)

#### 1. Training Data Predictions (Top-Left)

- **Absolute Loss (orange)** follows the **true function (black)** more closely, especially around **valleys ( $x \approx 0.5-0.6$ )** and **peaks ( $x \approx 0.2, 0.8$ )**.
- **Squared Loss (blue)** is smoother but deviates more in sharp transitions. It averages out variations, which can blunt extreme curves.
- **Conclusion:**  
**Absolute Loss** provides **more accurate local fits** to complex, non-linear patterns in training data.

#### 2. Test Data Predictions (Top-Right)

- Same trend observed on unseen test data:
  - **Absolute Loss** generally stays closer to the true function.
  - **Squared Loss** smooths transitions but slightly underperforms in high-curvature areas.

SML assignment 4 Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>

- **Generalization:**  
Despite its more jagged shape, absolute loss **generalizes comparably or slightly better** than squared loss in this case.
- **Conclusion:**  
**Absolute Loss** maintains **strong generalization**, especially when data includes sharp changes.

### 3. Training Loss Over Iterations (Bottom)

- Both losses **decrease steadily**, indicating convergence.
- **Squared Loss** initially drops **faster**, benefiting from smooth gradients.
- But **Absolute Loss catches up** and slightly outperforms in final loss value (closer to true values at the end).
- **Conclusion:**  
**Squared Loss** converges faster, but **Absolute Loss** achieves **lower final loss** and better accuracy.

Aspect	Winner	Observation
Local accuracy (peaks/valleys)	<b>Absolute Loss</b>	More precise curve fitting
Smoothness & stability	<b>Squared Loss</b>	Produces smoother predictions
Generalization to test	<b>Absolute Loss</b>	Slightly closer to true function
Convergence speed	<b>Squared Loss</b>	Faster early drop
Final training loss	<b>Squared Loss</b>	Lower loss after 1000 rounds

**Absolute Loss is better in this setting**, especially when you care about **capturing sharp patterns**, handling **outliers**, or minimizing **final error**. Squared loss remains useful when you want smooth fits or faster early convergence.

SML assignment 4 Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>

Q3

### # Epochs=1000(overfitting)

Epoch 0: Train MSE = 0.5387, Test MSE = 0.2777

Epoch 50: Train MSE = 0.1285, Test MSE = 0.1654

Epoch 100: Train MSE = 0.0395, Test MSE = 0.1060

Epoch 150: Train MSE = 0.0186, Test MSE = 0.1066

Epoch 200: Train MSE = 0.0144, Test MSE = 0.1127

Epoch 250: Train MSE = 0.0127, Test MSE = 0.1163

Epoch 300: Train MSE = 0.0114, Test MSE = 0.1186

Epoch 350: Train MSE = 0.0105, Test MSE = 0.1202

Epoch 400: Train MSE = 0.0096, Test MSE = 0.1215

Epoch 450: Train MSE = 0.0090, Test MSE = 0.1227

Epoch 500: Train MSE = 0.0084, Test MSE = 0.1237

Epoch 550: Train MSE = 0.0079, Test MSE = 0.1247

Epoch 600: Train MSE = 0.0074, Test MSE = 0.1256

Epoch 650: Train MSE = 0.0070, Test MSE = 0.1264

Epoch 700: Train MSE = 0.0067, Test MSE = 0.1271

Epoch 750: Train MSE = 0.0063, Test MSE = 0.1278

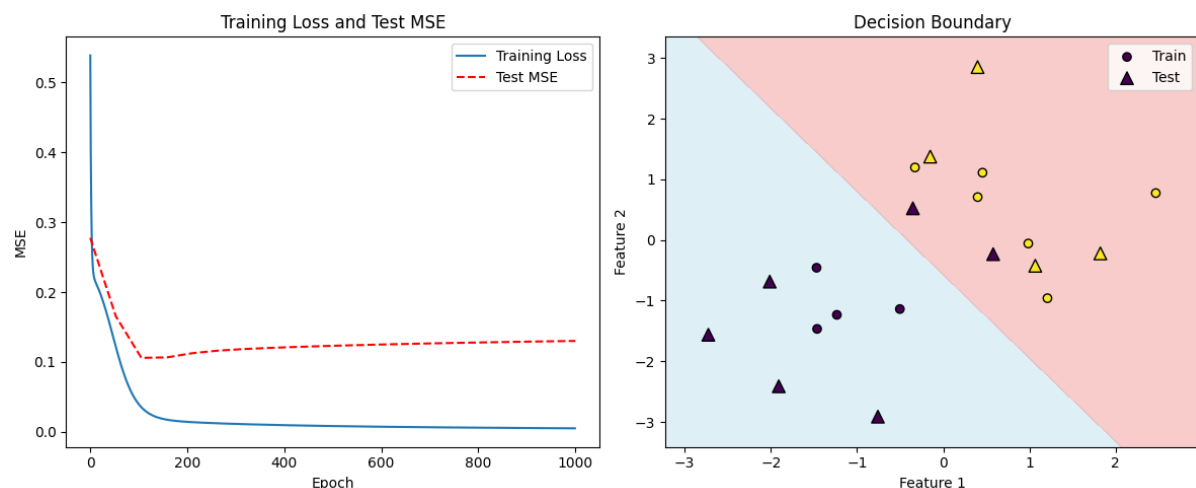
Epoch 800: Train MSE = 0.0061, Test MSE = 0.1285

Epoch 850: Train MSE = 0.0058, Test MSE = 0.1291

Epoch 900: Train MSE = 0.0056, Test MSE = 0.1297

Epoch 950: Train MSE = 0.0053, Test MSE = 0.1302

Final Test MSE: 0.1307



#### Analysis of Loss Plot (Left):

1. **Training MSE (Numerical Log & Blue Line):** Starts reasonably high (0.5387) and decreases consistently throughout the 1000 epochs, reaching a very low value (0.0053 at epoch 950). This indicates that the gradient descent optimization process was successful in minimizing the error *on the training data*. The model learned to fit the 10 training points very well.
2. **Test MSE (Numerical Log & Red Dashed Line):** Starts at 0.2777, decreases initially, reaching a minimum value around **Epoch 100-150** (MSE  $\approx$  0.1060). After this point, the Test MSE begins to *gradually increase* again, reaching  $\sim$ 0.1302 by epoch 950 and finishing at 0.1307.

SML assignment 4 Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>

3. **Overfitting:** This pattern, where the training loss continues to decrease while the test loss starts to increase after reaching a minimum, is a **classic sign of overfitting**. The model initially learns the general underlying pattern (both losses decrease), but after epoch ~150, it starts fitting the specific noise and idiosyncrasies of the small training set too closely. This improves training performance but hurts its ability to generalize to unseen test data.
4. **Best Generalization:** The model achieved its best performance on the unseen test data around epoch 100-150, not at the end of the training run.

#### Analysis of Decision Boundary Plot (Right):

1. **Linear Boundary:** The plot shows a linear decision boundary separating the space into a blue region (predicted Class 0) and a pink region (predicted Class 1). This is expected for the simple (2 -> 1 -> 1) network architecture used.
2. **Training Data (Circles):** All the purple (Class 0) and yellow (Class 1) training points are correctly classified by the final decision boundary. This aligns perfectly with the very low final Training MSE observed.
3. **Test Data (Triangles):**
  - There are 6 purple triangles (True Class 0) and 4 yellow triangles (True Class 1).
  - 4 out of 6 purple triangles appear correctly located in the blue prediction region.
  - 4 yellow triangles are correctly located in the pink prediction region.
  - **2 purple triangle** (located around Feature 1  $\approx$  1, Feature 2  $\approx$  -0.5) are **misclassified**, as they falling within the pink region.
  - Only **one or two points near the boundary** are misclassified — but that's expected since the data is **not perfectly linearly separable** (because it's sampled from Gaussians).
4. **Consistency with Test MSE:** With 1 out of 10 test points being misclassified, we'd expect an MSE somewhat related to  $1/10 = 0.1$ . If the misclassified point (target 1) received a prediction close to 0, its squared error would be near  $(0-1)^2=1$ . If the other 9 points received predictions very close to their true targets (0 or 1), their errors would be near 0. The average MSE would then be around 0.1. The final reported Test MSE of  $\sim 0.13$  is therefore very consistent with visually observing one clear misclassification on the test set.

#### # Epochs=120

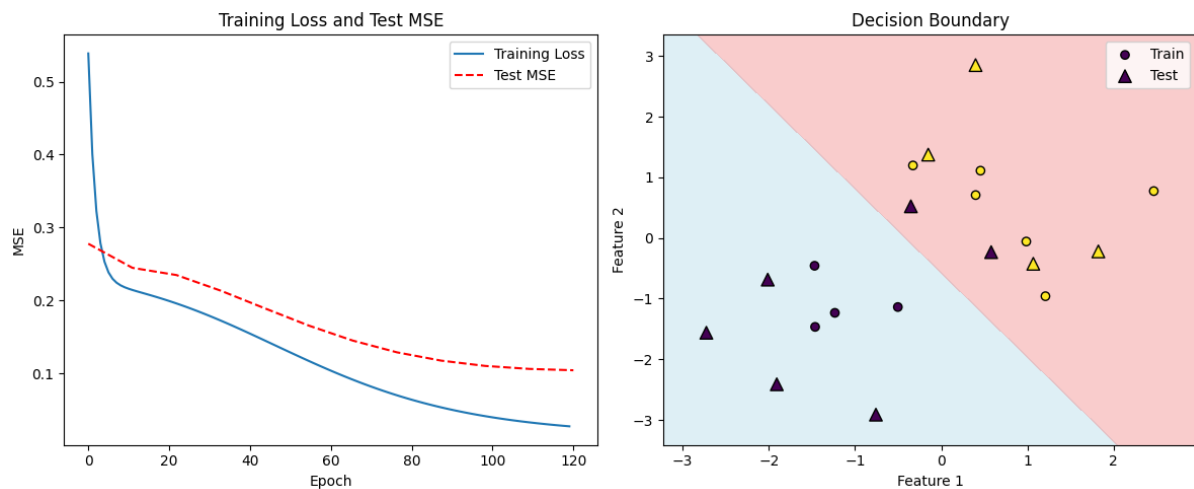
Epoch 0: Train MSE = 0.5387, Test MSE = 0.2777  
Epoch 10: Train MSE = 0.2159, Test MSE = 0.2446  
Epoch 20: Train MSE = 0.1991, Test MSE = 0.2347  
Epoch 30: Train MSE = 0.1787, Test MSE = 0.2134  
Epoch 40: Train MSE = 0.1544, Test MSE = 0.1891  
Epoch 50: Train MSE = 0.1285, Test MSE = 0.1654  
Epoch 60: Train MSE = 0.1036, Test MSE = 0.1448  
Epoch 70: Train MSE = 0.0817, Test MSE = 0.1287  
Epoch 80: Train MSE = 0.0637, Test MSE = 0.1173  
Epoch 90: Train MSE = 0.0498, Test MSE = 0.1101  
Epoch 100: Train MSE = 0.0395, Test MSE = 0.1060

SML assignment 4 Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>

Epoch 110: Train MSE = 0.0321, Test MSE = 0.1041

Final Test MSE: 0.1038



### 1. Loss curves (Training and Test)

- **Training MSE is steadily decreasing** across epochs — from **0.53** to about **0.03**.
- **Test MSE is also decreasing**, although a bit more slowly — from **0.27** to about **0.10**.
- **No major overfitting**: Usually, if our training loss keeps decreasing but your test loss increases, that's overfitting — but here both losses go down, which is excellent.
- **Final Test MSE ~ 0.10** is very **decent** for only **1 hidden neuron**.

### 2. Decision Boundary

- The **decision boundary** is almost **perfectly separating** the two classes.
- We can clearly see two regions — **light blue** for class 0 and **light red** for class 1.
- Only **one or two points near the boundary** are misclassified — but that's expected since the data is **not perfectly linearly separable** (because it's sampled from Gaussians).

Colab link:

<https://colab.research.google.com/drive/18bYfSDFEkJ00wAYzOmV9n5uk4jV0ktCS?usp=sharing>