Optimization of Target Values in an Artificial Neural Network

Andrassik, Bhuyian, Yakar, Zauner

Recap: Definitions

- Target values → expected output a neural Network is trained to produce for a specific given input
- Class values → values assigned to the correct class position in the target vector
- Non-class values → values used for all incorrect class positions in the target vector

Recap: Encoding Methods

One-hot encoding

$$\text{Target}_{\text{cat}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \text{Target}_{\text{dog}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{Target}_{\text{rat}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Class values

- 1 (cat / dog / rat)

Non-class value

- 0

These could be anything in [0, 1]!

→ Introduce adaptive assignment via Sigma-Adaptation

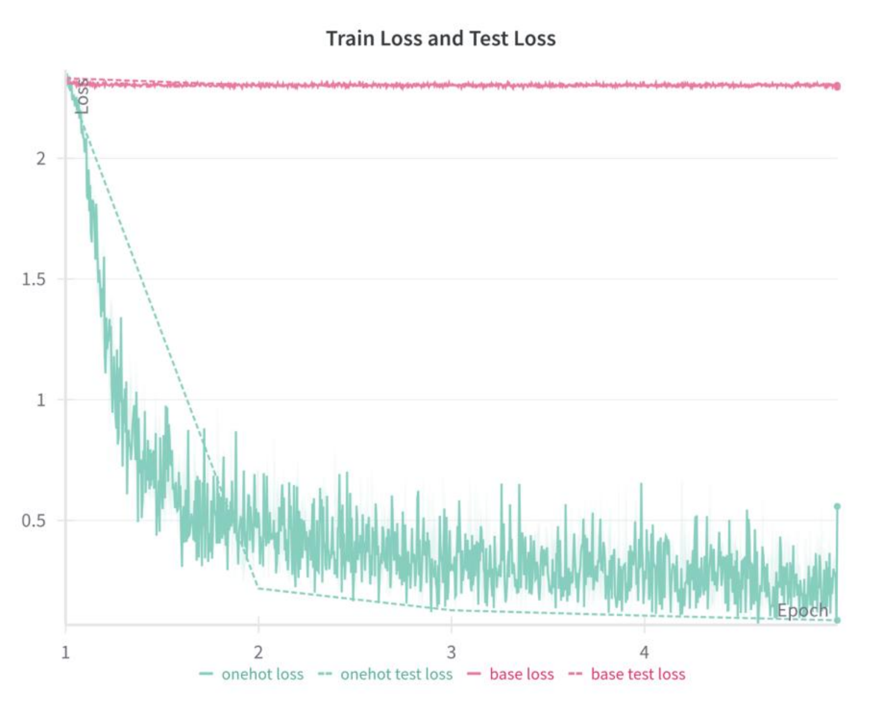
Adaptive Refinement - General Approach

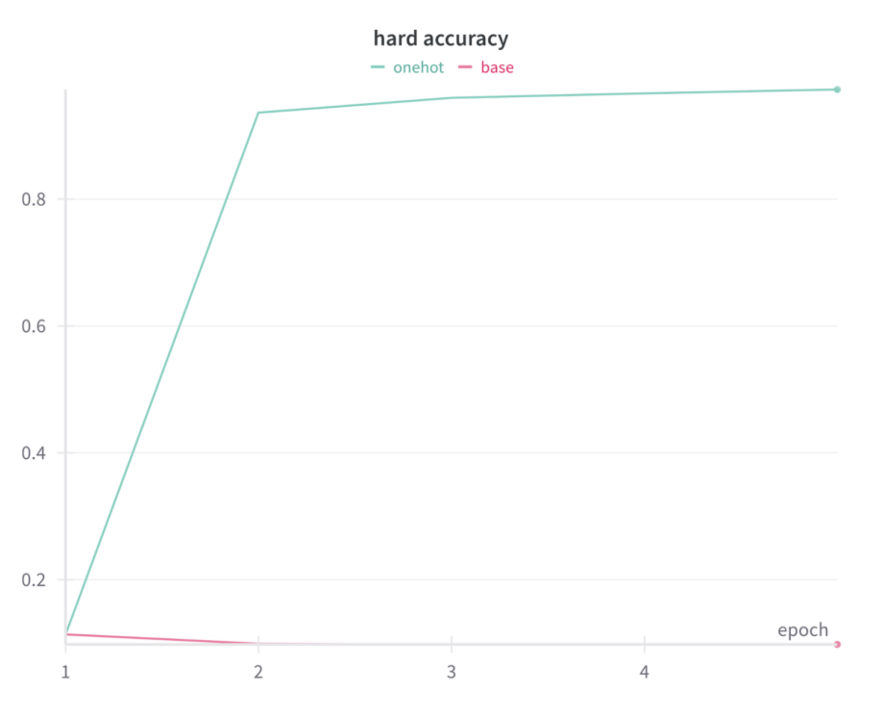
Each epoch:

- Record all outputs
- Adjust the class value by averaging it with the most consistent class output seen so far
- Adjust the non-class value by averaging all non-class outputs across the epoch (and previous guesses)

This method dynamically adapts the target values to match the evolving output behavior of the network

Base Approach Performance





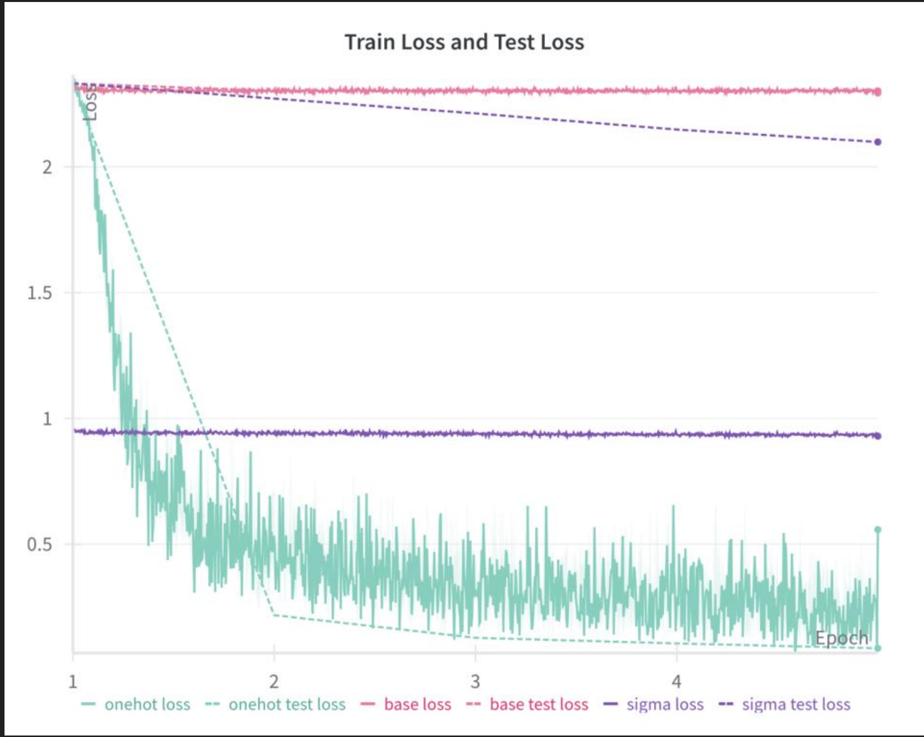
σ-Adaptation Prototype

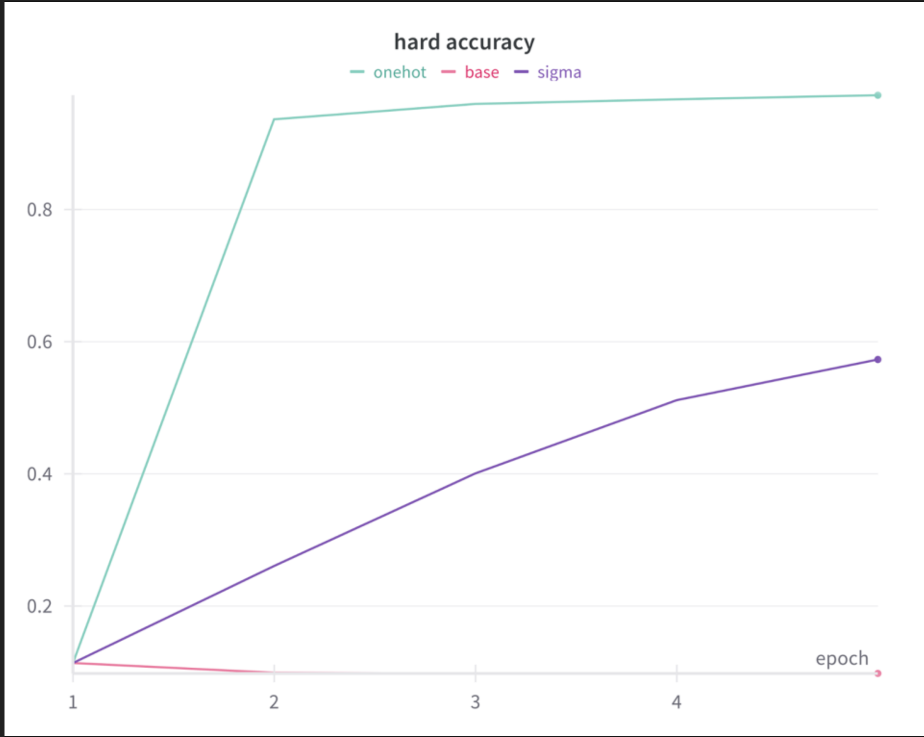
Uses the variability in outputs (standard deviation) to dynamically enforce separation

Steps:

1. Compute the sum of standard deviations for current outputs: σ_{sum}
2. For every class ensure:
   - All values remain within `[0, 1]`
   - $|class - non\text{-}class| \geq \sigma_{sum}$
3. If too close:
   - Move non-class **down** by σ_{sum}
4. If this would exceed bounds clip to 0 or 1

σ-Adaptation Performance





Alternative σ-Adaptation Methods

Adjust both class and non-class values →

- $class_{adjusted} = class_{previous} + \sigma_{sum}$
- $non\text{-}class_{adjusted} = non\text{-}class_{previous} - \sigma_{sum}$

Use only half of sigma-sum →

- $class_{adjusted} = class_{previous} + (\sigma_{sum} * ½)$
- $non\text{-}class_{adjusted} = non\text{-}class_{previous} - (\sigma_{sum} * ½)$

Next Steps

We can increase accuracy by:

- Improving the adaptation
  - Move more in early training
  - Adjust the calculation logic
  - …
- Reducing Overfitting
  - Regularization techniques
  - …