Dynamical Loss Function Implementation Documentation

1. Load and Preprocess Data

- The code loads the MNIST dataset and preprocesses the images and labels.

2. Neural Network Architecture

- The neural network architecture is defined using Keras. It's a simple feedforward network.

3. Dynamical Loss Function (Weighted Loss)

- A custom dynamical loss function `weighted_loss` is defined. The weights for the loss function are determined by the function `c_fn`.

Formulas: - The dynamical weight function $c_{fn}(t, i, w_{max})$ is defined as follows:

$$w(t,i) = 1 + t \cdot slope \quad for\ t < \frac{T}{2}$$

$$w(t,i) = 2 \cdot w_{max} - t \cdot slope - 1 \quad for\ t \geq \frac{T}{2}$$

$$slope = \frac{2 \cdot (w_{max} - 1)}{T}$$

$$w = \frac{1}{C} \cdot normalize\left(1 + (w_{main\_class} - 1) \cdot one\_hot(i, C)\right)$$

Where: - C is the number of classes. - 1 is a vector of ones. - $one\_hot(i, C)$ is a one-hot encoded vector for class i. - w_{main_class} is the main class weight. - The final weight vector w is normalized to ensure that its elements sum to C.

4. Standard Loss Function

- A standard cross-entropy loss function is defined for comparison during training.

Formula:

$$StandardCross - EntropyLoss = -\frac{1}{C} \sum_{j=1}^{C} y_j \cdot \log(softmax(predictions))$$

Where: - C is the number of classes. - y_j is the one-hot encoded ground truth label for class j. - $softmax(predictions)$ is the softmax activation of the neural network predictions.

5. Accuracy Calculation

- A function to calculate accuracy is defined.

Formula:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} I(true\_label_i = predicted\_label_i)$$

Where: - N is the number of samples. - $I(condition)$ is the indicator function.

6. Training Loop

- The training loop uses the custom loss function and updates the model's weights accordingly.

Formulas: - The training loop iterates over time steps and updates the weights dynamically based on the current time t and class index c.