Department of Artificial Intelligence and Human Interfaces
University of Salzburg

PS Natural Computation
SS 24

Dynamic Optimization of Target Values in Neural Network Classification

June 29, 2025

Project Members:
Andrassik Leon, 12209906, leon.andrassik@stud.plus.ac.at
Bhuiyan Amreen, 12203597, amreen.bhuiyan@stud.plus.ac.at
Yakar Betül, 12205751, betuel.yakar@stud.plus.ac.at
Zauner Sofia, 12118492, sofia.zauner@stud.plus.ac.at

Academic Supervisor:
Helmut MAYER
helmut@cosy.sbg.ac.at

Correspondence to:
Universität Salzburg
Fachbereich AIHI
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Abstract

The aim of our project is to implement an alternative method of choosing target values for classification with neural networks. Instead of using traditional one-hot encoding, which uses fixed values of 0 and 1, this method involves assigning custom target values to each class.

These target values are dynamically optimized during the training process, which distinguishes our approach from conventional fixed-target training. Our idea is to capture the model's natural predictions and use these to refine the target values.

This approach requires a new rule for determining the predicted class: rather than choosing the neuron with the highest value, we select the neuron whose output value is closest to its corresponding target value and therefore has the lowest error; a method we will refer to as "minimum distance classification".

The goal of our project is to increase training efficiency while maintaining acceptable classification accuracy. To evaluate the performance of our approach, we will test it on the MNIST dataset and compare it with the one-hot encoding method.

1 Introduction

Artificial neural networks (ANNs) have become essential tools in modern classification tasks, mapping complex input data to discrete predefined output categories. During training, the error is calculated as the difference between the desired output and the actual output produced by the network. The objective of the training process is to minimize this error through iterative adjustments of the network's weights and biases. Following the training phase, performance is evaluated using a test set that contains previously unseen data, providing a neutral measure of its generalizability and classification accuracy.

Traditionally, the learning process of such networks is based on one-hot encoded target values, where the correct class is represented by the value 1, and all others are assigned a 0. This binary approach became standard due to its simplicity and compatibility with widely used loss functions. However, enforcing these extreme target values could potentially lead to challenges depending on the system's underlying activation function. It may negatively affect gradient flow, reduce training efficiency, and encourage overconfident predictions.

One-hot encoding enforces a "winner-takes-all" (WTA) dynamic, where only the neuron with the highest output is considered correct. Since the target vector consists of a 1 for the correct class and a 0 everywhere else, this is effectively equivalent to selecting the output that is closest to its respective target value. Thus, WTA can be seen as a special case of a more general strategy, which is known as nearest target or minimum distance classification. Here we make a general comparison between the entire output distribution given by the network and the available target vectors for each class. Whichever class's target the output is closest to (using a distance metric such as Euclidean distance) will be chosen as the correct class.

In this project, we propose a more flexible approach for target value assignment in classification networks. Instead of defining fixed values of 0 and 1, we generate target vectors composed of custom class and non-class values. Here, class values refer to the target values assigned to the positions which would ordinarily be filled with 1 in a one-hot vector, while the non-class values fill all other positions. These alternative targets are not chosen arbitrarily, but rather inferred from the model's own inherent output tendencies during training. By adapting target values dynamically in response to the network's evolving behavior, we aim to guide learning in a way that is more aligned with the model's natural predictions.

This change also calls for a rethinking of how predictions are interpreted. Instead of relying on the activation of the highest neuron to determine the predicted class as in WTA, we introduce a "nearest target classification" approach. We select the class whose output vector value is closest to the corresponding target vector, thereby minimizing the error. This nearest target classification strategy reflects the idea that a well-trained model should not necessarily produce a maximal value in any particular output neuron, but rather accurately match the overall desired response.

Our hypothesis is that this method improves the training efficiency, while maintaining the classification accuracy. We empirically evaluate this method using the MNIST dataset, comparing its performance against the traditional one-hot encoding approach.

2 Methodology

In this section, we describe our approach to implementing dynamic target value optimization for neural network classification. First, we break down the concrete architecture of the target vectors used for training and classification. We then discuss how the class and non-class values that make up these target vectors are initially constructed and subsequently optimized dynamically during training.

2.1 Target Value Architecture

To establish our methodology clearly, we first review the traditional one-hot encoding approach before defining our alternative target value structure.

2.1.1 Traditional One-Hot Encoding

In standard classification with C classes, the target vector $\mathbf{t}^{(i)}$ for a sample belonging to class i is defined as:

$$\mathbf{t}^{(i)} = [0, 0, \ldots, 0, \underbrace{1}_{i\text{-th position}}, 0, \ldots, 0]$$

For example, with 3 classes, the target vectors are:

$$\mathbf{t}^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{t}^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{t}^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2.1.2 Dynamic Target Value Structure

Our approach replaces the fixed values 0 and 1 with adaptive class and non-class values. For a classification problem with C classes, we define:

- c_i: the class value for class i (replaces 1 in one-hot encoding)
- $\bar{c}_i$: the non-class value for class i (replaces 0 in one-hot encoding)

The target vector $\mathbf{t}^{(i)}$ for class i becomes:

$$\mathbf{t}^{(i)} = [\bar{c}_i, \bar{c}_i, \ldots, \bar{c}_i, \underbrace{c_i}_{i\text{-th position}}, \bar{c}_i, \ldots, \bar{c}_i]$$

Using the same 3-class example:

$$\mathbf{t}^{(1)} = \begin{bmatrix} c_1 \\ \bar{c}_1 \\ \bar{c}_1 \end{bmatrix}, \quad \mathbf{t}^{(2)} = \begin{bmatrix} \bar{c}_2 \\ c_2 \\ \bar{c}_2 \end{bmatrix}, \quad \mathbf{t}^{(3)} = \begin{bmatrix} \bar{c}_3 \\ \bar{c}_3 \\ c_3 \end{bmatrix}$$

Importantly, in our implementation:

- Each class has its own class and non-class values c_i and $\bar{c}_i$
- All non-class values within a class are the same
- All non-class values $\bar{c}_i$ (and in special cases class values) are optimized dynamically during training

2.2 Target Value Initialization and Optimization

Since a sample's classification depends on finding the nearest target vector to the network's output, we need target vectors that are reasonably well-separated from one another. Given our construction method, this separation can be achieved by ensuring that each class's class and non-class values are sufficiently spaced apart.

Our approach accomplishes this through two stages: first, we select initial class and non-class values based on the network's natural output tendencies, and second, we dynamically increase the spacing between these values during training. This section describes both processes.

2.3 Initial Target Value Selection

The key challenge in selecting the initial class and non-class values is balancing two competing requirements: the values should reflect the model's innate output tendencies rather than being chosen arbitrarily, while also providing a distribution that enables sufficient separation, either initially or through the subsequent optimization process.

We explored several strategies to achieve this balance:

- Uniform...
- Soft...
- Pref...
- Redistributed Pref...

Based on our experiments we found that the [...] approach yielded the best results.

Explain best warm-up method...

2.4 Dynamic Target Optimization Algorithm

Note: Mention uni directional and bi directional sigma...

3 Experimental Setup

4 Results and Analysis

5 Discussion and Conclusion

6 Milestones

- **26.03.25 - Project Kickoff and Theoretical Outline**
  Create a rough outline of the project's content based on the given paper. Include loose definitions and a short summary of the introduced methods.
- **09.04.25 - Goal Specification**
  Clarify the project idea, define concrete project goals, and deepen the understanding of the topic. Focus especially on precise definitions of target values, class values, and non-class values. Write the abstract and milestone plan for the semester.
- **23.04.35 - Algorithm Design and Planning**
  Start working on the paper. Deepen research on alternative target value methods for classification with neural networks. Select and define an approach for alternative target encoding. Plan implementation steps for generating "class-" and "non-class" values.
- **07.05.25 - Initial Implementation and Testing of target value algorithm**
  Initial implementation of the new target encoding method. Before testing, allocate time for experimental design to ensure solid and interpretable results. Conduct simple experiments using simplified datasets to validate functionality.
- **21.05.25 - Compelete Implementation and Evaluate on MNIST-Experiments**
  Complete implementation and testing of the alternative target value methods. Prepare and conduct experiments on the MNIST dataset. Analyze model predictions to evaluate performance compared to traditional one-hot encoding. Discuss results and possible improvements.
- **04.06.25 - Optimization and Comparative Analysis**
  Complete improvements and experiments. Analyze experimental results and compare training efficiency and classification accuracy with traditional methods. Document findings and prepare initial draft of project report.
- **18.06.25 - Finalization**
  Final preparations for submission. Complete report and presentation.

7 Progress of Work

7.1 Week 1, Tuesday, 26.03.2025

7.2 Week 2, Tuesday, 02.04.2025

7.3 Week 3, Tuesday, 09.04.2025

7.4 Week 4, Tuesday, 16.04.2025

7.5 Week 5, Tuesday, 23.04.2025

7.6 Week 6, Tuesday, 30.04.2025

7.7 Week 7, Tuesday, 07.05.2025

7.8 Week 8, Tuesday, 14.05.2025

7.9 Week 9, Tuesday, 21.05.2025

7.10 Week 10, Tuesday, 28.05.2025

7.11 Week 11, Tuesday, 04.06.2025

7.12 Week 12, Tuesday, 11.06.2025

7.13 Week 13, Tuesday, 18.06.2025

8 Subproject Responsibilities

- **Programming of Target Value Calculation Methods (Phyton)**
  Leon Andrassik
- **Research and Preparation of Target Encoding Methods**
  Betül Yakar, Sofia Zauner
- **Evaluation and Result Analysis**
  Amreen Bhuiyan, Betül Yakar, Sofia Zauner
- **Project Documentation and Preparation of Presentations**
  Leon Andrassik, Amreen Bhuiyan, Betül Yakar, Sofia Zauner
- **Presenting Progress Presentations**
  Leon Andrassik, Betül Yakar, Sofia Zauner
- **Presenting Final Presentation**
  Amreen Bhuyian

9 Links

- Project Page: `http://student.cosy.sbg.ac.at/???`
- PS Page: `http://www.cosy.sbg.ac.at/~helmut/Teaching/PSRobotik/`