

# Learning From Differences

Oliver Linger

February 19, 2024

**Final-Year Project - B.Sc. in Computer Science**

**Supervisor:** Dr. Derek Bridge

**Department of Computer Science**

**University College Cork**

## Abstract

# Declaration of Originality

In signing this declaration, you are confirming, in writing that the submitted work is entirely your own original work, except where it clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award.

1. This body of work is all my own, unless clearly stated otherwise, with the full and proper accreditation;
2. With respect to my work: none of it has been submitted to any educational institution contributing in any way to an educational award.
3. With respect to anyone else's work: all diagrams, text, code, or ideas have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or print.

*Signed*

---

*Date: February 19, 2024*

## 0.1 Acknowledgements

# Contents

0.1	Acknowledgements . . . . .	1
<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Literature Review</b>	<b>7</b>
2.1	KNN + ANN for Regression, utilizing (CBR) . . . . .	8
2.1.1	Learning From Differences Paper . . . . .	8
2.2	Learning Adaptations for Case-based classification . . . . .	11
2.2.1	An NN-CDH Approach for Classification . . . . .	12
<b>3</b>	<b>Design</b>	<b>13</b>
3.1	Linger regression model . . . . .	13
3.1.1	Objective . . . . .	13
3.1.2	Architecture Overview . . . . .	14
3.1.3	Components . . . . .	14
3.1.4	Parameterization . . . . .	15
3.1.5	Training and Prediction . . . . .	15
3.1.6	Data Collection and Preprocessing . . . . .	15
3.1.7	Feature Engineering . . . . .	15
3.1.8	Model Selection . . . . .	15

3.1.9	Training Procedure . . . . .	15
3.1.10	Evaluation Metrics . . . . .	15
3.1.11	Cross-Validation and Testing . . . . .	16
3.1.12	Hardware and Software Specifications . . . . .	16
3.1.13	Ethical Considerations . . . . .	16
3.1.14	Limitations and Future Work . . . . .	16
<b>4</b>	<b>Implementation</b>	<b>17</b>
<b>5</b>	<b>Evaluation</b>	<b>18</b>
<b>6</b>	<b>Conclusions</b>	<b>19</b>
<b>7</b>	<b>Experiments</b>	<b>20</b>
7.1	Experiment 1: Abalone dataset Linger Regression . . . . .	20
7.2	Experiment 2: House Prices Dataset Regression . . . . .	20
7.3	Experiment 3: Glass Dataset Classification . . . . .	20
7.4	Experiment 4: Raisin Dataset Classification . . . . .	20
7.5	Experiment 5: Car Safety Dataset Classification . . . . .	20
<b>8</b>	<b>Results</b>	<b>21</b>

# List of Figures

# List of Tables



# Chapter 1

## Introduction

# Chapter 2

## Literature Review

In recent years, the application of neural networks in regression tasks has gained significant attention within the realm of machine learning and artificial intelligence. Applying neural networks to regression tasks has become common practice within the machine learning community [6]. Traditionally, supervised learning approaches involve presenting datasets described by features alongside their corresponding expected values to the artificial neural network (ANN), facilitating the learning process aimed at predicting these expected values.

Inspired by case-based reasoning, which involves solving new problems by adapting solutions from old ones [7], researchers have developed predictive ensemble models based on the principles of case-based reasoning (CBR). These principles involve retrieving the most similar cases, reusing them to attempt to solve the problem, revising the proposed solution if necessary, and retaining the new solution as part of a new case [8].

Researchers have suggested the potential advantages of deviating from this conventional paradigm by training neural networks on differences between sets of features to predict differences between values. This departure from the traditional approach underscores a shift towards a more nuanced understanding of the learning process, emphasizing the exploration of alternative methodologies to enhance the efficiency and efficacy of neural network training. Additionally, it aims to provide more transparency within the learning process to offer insights into the black box that is a neural network.

## 2.1 KNN + ANN for Regression, utilizing (CBR)

The exploration of alternative methodologies for enhancing the performance of neural networks in regression tasks has attracted considerable attention within the machine learning community in recent years. One innovative approach, inspired by the principles of case-based reasoning (CBR), involves training neural networks to predict differences between values based on differences between sets of features.

This departure from the conventional supervised learning paradigm aims to leverage the inherent advantages of learning from differences, potentially leading to more interpretable and efficient models. The study under consideration [2] conducts a systematic factorial study to investigate the efficacy of this approach across various datasets and experimental conditions. The findings suggest that neural networks trained on differences demonstrate comparable or even superior performance to those trained using traditional methods, while requiring significantly fewer epochs for convergence. In this section of the literature review, we consider the utilization of case-based reasoning for regression tasks, with a specific focus on the implementation of KNN (K-Nearest Neighbors) and ANN (Artificial Neural Network) frameworks.

### 2.1.1 Learning From Differences Paper

The paper by [2] introduces the innovative concept of training a neural network based on the disparities between a base case and its closest neighbors. Titled "A Factorial Study of Neural Network Learning from Differences for Regression" [2], the study aims to present novel learning methodologies geared towards enhancing performance and fostering a more interpretable learning process. The research evaluates three model variations: a benchmark neural network, a learning from differences model, and an augmented learning from differences model incorporating the original context. The context here refers to the base case, from which differences with its nearest neighbors are calculated.

According to [2], the study's key findings underscore a significant increase in accuracy when incorporating contextual information in the learning process. Moreover, comparable or superior results were achieved with fewer training epochs compared to the basic neural network. This enhancement is attributed partly to the broader training dataset that encompasses neighbors greater than 1. Notably, the paper advocates for a hybrid approach, combining learning from differences with direct feature-based learning, which yields superior results.

The related work for the paper [2] has addressed the use of neural networks in the case-based reasoning process. These papers have focused on exploiting the base case using the idea that adaptation of information is acquirable from the differences between pairs [3]. Since then, various different approaches have been used. More recently, in relation to using neural networks to predict the differences between problems, a few interesting points have been investigated. These points show the ability of a neural network to correct the solution of the most similar retrieved case. These preliminary works expressed in the paper showed that the use of differences plus context yielded superior results [1]. The previous studies examined in the paper [2] did not examine the impact of different parameters during the (CBR) case-based reasoning experiments. The paper [2] made an effort to explore the effect of epochs in the learning process.

The graphical representations in the paper highlight a distinct trend: rapid attainment of high accuracy levels with the learning from differences method, particularly when contextual information is included. Conversely, the basic neural network requires a substantially larger number of epochs to achieve comparable accuracy, if not worse.

This disparity suggests the efficiency of the learning from differences approach.

In conclusion, the paper suggests a marginal performance enhancement and a substantial reduction in training epochs through the adoption of the differences method and inclusion of contextual information. However, it is crucial to acknowledge the time overhead associated with retrieving and training using similar cases.

The process of learning from differences for regression, as described in the paper by [2], demonstrates the use of the case difference heuristic in the creation of a predictive model. Consider a dataset  $D$  of pairs  $(X_i, Y_i)$ , where  $Y_i$  is a vector of  $n$  numerical values corresponding to the result we expect to predict. Within the paper, the method of executing a regression model follows the normal pattern of training and predicting that a standard model would. However, instead of the features themselves being passed into the neural net, the differences between the features and their nearest neighbors are passed in. The differences set is represented by  $\Delta D^{\text{train}}$ . The network, after being fitted with the above, is used again to make predictions. This time, the test set is used, where nearest neighbors in the training set are retrieved using the items in the training set  $D^{\text{train}}$ . The resultant dataset of test differences is called  $\Delta D^{\text{test}}$ . Predictions are made on this dataset, and the predictions are added to their corresponding differences in  $D^{\text{train}}$ . The average of the resultant label is taken as the regression prediction.

The paper [2] has an associated library where they have implemented a regression-based machine learning model.

The new model proposed in the paper is tested using the factorial study methodology. This process involved finding a network structure through experimental means for each dataset. It entailed training and testing using a wide range of hyperparameters to determine the optimal settings.

Despite its contributions, the paper falls short in adequately addressing classification and exploring additional variations of the learning from differences model beyond context inclusion. Incorporating original values for the base case in the generated differences dataset provides a valuable framework for understanding the observed disparities. While the model with contextual inclusion demonstrates slight improvements over the base model in learning from differences, a more comprehensive examination of the methodology is warranted for further refinement and enhancement.

The paper tested three models using the same factorial methodology: a base version, which is a basic neural network trained empirically; a differences model employing the (CBH) Case-based Heuristic for predictions; and a differences model with its base context included. The results yielded interesting insights, with similar performance for all three models and a slight improvement in the differences + context model. Notably, the differences and differences + context models required significantly fewer epochs to achieve accuracy compared to the basic neural network, which achieved similar accuracy after more epochs.

Another consideration in the paper [2] was the number of neighbors for both training and prediction. The study revealed no consistent trend across datasets, suggesting dataset-specific requirements. This approach could be seen as a method of data augmentation, as choosing a larger number of neighbors dramatically increases the dataset size.

In conclusion, the learning from differences method achieved equal or higher accuracy compared to the base neural network model with far fewer epochs, potentially leading to faster training times. However, careful consideration must be given to the number of neighbors used, as increasing it can lead to significantly higher time overhead.

An improvement on the study [2] would be to vary the network structure, including but not limited to increasing the number of layers and neurons. Additionally, including datasets that require different types of networks could provide valuable insights. Exploring these parameters could lead to improvements in the existing learning from differences regression model. While the study tested a singular variant through the inclusion of context, exploring more variants of the model could yield further enhancements.

## 2.2 Learning Adaptations for Case-based classification

The paper [9] "Learning Adaptations for Case-based Classification" explores the implementation of Case-Based Reasoning (CBR) for classification. It investigates recent advancements in CBR and its replacement of learning rules with Case-Based Heuristic (CBH) network models for adaptation. The paper introduces a novel model with three variations: segmentation of adaptation knowledge based on the classes of the source cases, training a single neural network on differences between problem solutions, and adapting from an ensemble of source cases followed by a majority vote.

This study [9] presents the first neural network-based approach to classification, called "C-NN-CDH" or Classification with Neural Network-based Case Difference Heuristic. The approach uses a neural network to learn adaptation knowledge from pairs of cases. According to the paper, this approach outperforms traditional statistical models.

The first approach involves segmenting the dataset by class and training a neural network for all segmented differences set between base cases. This method offers faster training but slightly less accuracy, according to the paper. The use of these CBR methods provides at least two benefits: inertia-free lazy learning and the ability to assess different cases when running classification. Inertial free means that the model does not make assumptions based on the underlying structure of the data, and lazy learning pertains to an algorithm that will postpone generalization until it is certain a new instance needs to be classified. It will only classify as the data comes; it won't try and learn or adapt preemptively.

In the process of CBR, adaptation is often treated as the most difficult. Various methods have been developed by academics over the years to tackle the issue. For example, [5] extrapolates a method in which the case base adaptation cases are populated from previously successful adaptations. Another approach is to have a stored base case and query and retrieve the case most similar to it, and adapting the retrieved case to the base case [1].

In the paper [9], several design questions had to be addressed. How to calculate problem differences and how to select case pairs for training. The paper in question [9] considers standard pair selection methods as well as a new training per selection approach based on class-class classification.

The paper [9] addresses the application of (CDH) method for classification. It discusses the value distance metric which is probabilistic method to measure the similarity that allows the comparison of nominal values in a single dimensional space.

### **2.2.1 An NN-CDH Approach for Classification**

The "NN-CDH" method learns through pairs of cases. With one being treated as the source and the other as the target. The source is to be adapted to the target. This is often referred to as a case pair. With the incorporation of a neural network, the neural network calculates the difference between the source and target pair. The (CBA) case based reasoning system applies predicted result of this is applied to the source solution and that is taken as the result.

# Chapter 3

## Design

### 3.1 Linger regression model

#### 3.1.1 Objective

The objective of the Linger Regression model is to provide a more dynamic and interpretive model for Regression using differences which implements (CBR) case-based reasoning. This model aims to enhance the interpretability and performance of regression models, facilitating a more transparent learning process [2]. Specifically, the goals are outlined as follows:

The primary aim is to develop a regression model that captures dynamic relationships within the data, leveraging the principles of case-based reasoning. By incorporating differences, the model can adapt to varying contexts and provide nuanced predictions.

The model seeks seamless integration with popular machine learning libraries like Scikit-learn [4]. By providing a codebase compatible with Scikit-learn's pipeline and hyperparameter testing methods, the model ensures accessibility and ease of implementation for practitioners.

An essential objective is to enable users, including academics, researchers, and industry professionals, to interpret the model's predictions effectively. By elucidating the correlation between input features and predictions, the model facilitates deeper insights into the underlying data patterns.

Another of the core objective involves implementing the learning from differences algorithm [2] in a straightforward and testable manner. This implementation includes



various enhancements, such as incorporating original context and introducing additional features like distance metrics between base cases and their nearest neighbors.

The overarching goal is to provide a versatile regression model that not only achieves superior performance than previous (CBR) regression models but also empowers users with insights into the predictive process, fostering trust and understanding in machine learning applications.

### 3.1.2 Architecture Overview

Provide a concise overview of the architecture of the machine learning device. The **LingerRegressor** is a custom regressor that combines the principles of k-nearest neighbors (KNN) and the **MLPRegressor** from **scikit-learn**. It offers flexibility in parameter tuning and incorporates various techniques to enhance prediction accuracy and interpretability.

### 3.1.3 Components

- **K-nearest neighbors (KNN):** The model utilizes KNN for both training and prediction phases. During training, it computes the nearest neighbors of each data point and extracts the differences between them. The number of neighbors for training and prediction can be specified independently (**n\_neighbours\_1** and **n\_neighbours\_2**).
- **MLPRegressor:** The core regression model is implemented using the MLPRegressor from scikit-learn. It consists of multiple layers of neurons and supports various activation functions, solvers, and other hyperparameters. These parameters can be adjusted to optimize model performance.
- **Additional Features:**
  - **Weighted KNN:** Allows for weighted averaging of neighbor predictions based on distances.
  - **Context Addition:** Enables the addition of base case context to the training data, enhancing the model's ability to capture complex relationships.
  - **Additional Results Column:** Incorporates an additional column with distance values associated with each training instance.
  - **Duplication Based on Distance:** Duplicates training instances based on calculated distances to improve model robustness.

### **3.1.4 Parameterization**

The `LingerRegressor` offers extensive parameterization to fine-tune model behavior. Parameters include those for KNN (e.g., number of neighbors), `MLPRegressor` settings (e.g., activation function, solver), and additional features (e.g., weighted KNN, context addition).

### **3.1.5 Training and Prediction**

During training, the model computes differences between neighboring instances and fits the `MLPRegressor` to learn the underlying patterns. In prediction, it identifies nearest neighbors for each test instance and generates predictions using the trained `MLPRegressor`.

### **3.1.6 Data Collection and Preprocessing**

Explain how training and validation datasets were collected and preprocessed.

### **3.1.7 Feature Engineering**

Describe the process of feature extraction or engineering for the machine learning model.

### **3.1.8 Model Selection**

Explain the rationale behind selecting the machine learning model or algorithm.

### **3.1.9 Training Procedure**

Detail the training procedure, including optimization algorithms and hyperparameters.

### **3.1.10 Evaluation Metrics**

Define the evaluation metrics used to assess the performance of the machine learning device.

### **3.1.11 Cross-Validation and Testing**

Describe the methodology used for cross-validation and testing of the machine learning model.

### **3.1.12 Hardware and Software Specifications**

Provide details about the hardware and software infrastructure used in the development and deployment of the machine learning device.

### **3.1.13 Ethical Considerations**

Discuss any ethical considerations associated with the machine learning device.

### **3.1.14 Limitations and Future Work**

Acknowledge limitations and propose future improvements for the machine learning device.

# Chapter 4

## Implementation

## Chapter 5

## Evaluation

## Chapter 6

## Conclusions

# Chapter 7

## Experiments

7.1 Experiment 1: Abalone dataset Linger Regression

7.2 Experiment 2: House Prices Dataset Regression

7.3 Experiment 3: Glass Dataset Classification

7.4 Experiment 4: Raisin Dataset Classification

7.5 Experiment 5: Car Safety Dataset Classification

# Chapter 8

## Results



# Bibliography

- [1] Susan Craw, Nirmalie Wiratunga, and Ray C Rowe. Learning adaptation knowledge to improve case-based reasoning. *Artificial intelligence*, 170(16-17):1175–1192, 2006.
- [2] Mathieu D’Aquin, Emmanuel Nauer, and Justine Lieber. A factorial study of neural network learning from differences for regression. In Mark T. Keane and Nirmalie Wiratunga, editors, *Case-Based Reasoning Research and Development. ICCBR 2022. Lecture Notes in Computer Science*, volume 13405, page PageNumbersHere. Springer, Cham, 2022.
- [3] Kathleen Hanney and Mark T Keane. Learning adaptation rules from a case-base. In *European Workshop on Advances in Case-Based Reasoning*, pages 179–192. Springer, 1996.
- [4] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.
- [5] David B Leake and Andrew Kinley. Acquiring case adaptation knowledge: A hybrid approach. 1996.
- [6] Leorey Marquez, Tim Hill, Reginald Worthley, and William Remus. Neural network models as an alternative to regression. In *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, volume 4, pages 129–135. IEEE, 1991.
- [7] Christopher K Riesbeck and Roger C Schank. *Inside case-based reasoning*. Psychology Press, 2013.
- [8] Ian Watson and Farhi Marir. Case-based reasoning: A review. *The knowledge engineering review*, 9(4):327–354, 1994.
- [9] Xiaomeng Ye, David Leake, Vahid Jalali, and David J Crandall. Learning adaptations for case-based classification: A neural network approach. In *Case-Based Reasoning Research and Development: 29th International Conference, ICCBR 2021, Salamanca, Spain, September 13–16, 2021, Proceedings 29*, pages 279–293. Springer, 2021.