Learning From Differences

Oliver Linger

February 21, 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in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or print.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 21, 2024

0.1 Acknowledgements

Contents

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 problems based on differences between features rather than the features themselves.

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 ΔD^{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^{train}. The resultant dataset of test differences is called ΔD^{test}. Predictions are made on this dataset, and the predictions are added to their corresponding differences in D^{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 is a novel approach to classification that operates through pairs of cases, where one serves as the source and the other as the target. This method revolves around adapting the source case to the target case, forming what is commonly known as a case pair.

To achieve adaptation, a neural network is integrated into the process. The difference between the source and target cases is computed, and the neural network is trained on these differences.

The neural network computes the difference between the source and target pairs. This difference calculation is a crucial step in the Case-Based Reasoning (CBR) system.

Once the difference is computed, the predicted result obtained from the neural network is applied to the source solution. This adapted solution is then considered as the final result of the classification process.

The NN-CDH approach represents a fusion of neural network techniques with the principles of Case-Based Reasoning, offering a dynamic and adaptable method for classification tasks.

When calculating the difference between the problem and the solution values it is important to implement a difference function. This presents an interesting challenge for nominal values. The method presented in [9] uses an implicit calculation through machine learning techniques. This replaces tradition (CDH) case difference heuristic approaches for difference calculations. Through the use of a neural network the papers' method hopes to include the base context in the differences' calculation. This method for generating differences has been aptly named ("C-CDH") case difference heuristic approach for classification.

Multiple variants were created with the paper [9].

2.2.2 Variant 0: Non-Network C-CDH

Variant 0, as described in [9], serves as a foundational test bed implementation for classification tasks.

- **Storage of Case Pairs:** The C-CDH system stores case pairs, treating them as adaptation rules. Each pair consists of a source case and a target case.
- **Grouping Based on Source Solution:** Case pairs are grouped based on the source solution, facilitating efficient retrieval and comparison during classification.
- **Classification Process:**
  1. The system retrieves the most similar source case from the stored case pairs.
  2. It selects the case pair with the most similar source problem and target problem among those sharing the same source solution.
  3. The target solution of the retrieved case pair is utilized for the final classification.
- **Final Outcome:** The final classification outcome is determined based on the target solution of the retrieved case pair.

2.2.3 Variant 1: C-NN-CDH

Variant 1, referred to as C-NN-CDH from the paper [9], extends the foundational principles established in Variant 0, aiming to integrate neural networks into the classification process while maintaining the core functionalities of the C-CDH system. Below is the description of Variant 1:

- **Storage of Case Pairs:** Similar to Variant 0, the C-NN-CDH system stores case pairs as adaptation rules. Each case pair comprises a source case, a target case, and additional information related to probabilities. These probabilities could be obtained from the output of the neural network or calculated based on certain features of the data.
- **Grouping Based on Source Solution:** Case pairs are organized based on the source solution, ensuring efficient retrieval and comparison during the classification process.
- **Classification Process:**

1. The system retrieves the most similar source case from the stored case pairs. This can be done with euclidean distance via Nearest Neighbors.
2. In variant 1, probabilities of the source, target, and solution of the target are included. These probabilities could be generated by the neural network model based on the input data.
3. For Variant 1:
   - The system appends pairs of probabilities from the source and target classes.
   - It selects the case pair with the most similar source problem and target problem among those sharing the same source solution.
4. The target solution of the retrieved case pair is utilized for the final classification.

- **Final Outcome:** Similar to Variant 0, the final classification outcome is determined based on the target solution of the retrieved case pair.

2.2.4 Variant 2: C-NN-CDH

Variant 2, is very similar to variant 1 but includes the source solution during the training of its neural network.

- **Storage of Case Pairs:** Similar to Variant 0, and 1, the variant 2 system stores case pairs as adaptation rules. Each case pair comprises a source case, a target case, and additional information related to probabilities. These probabilities could be obtained from the output of the neural network or calculated based on certain features of the data.

- **Grouping Based on Source Solution:** Case pairs are organized based on the source solution, ensuring efficient retrieval and comparison during the classification process.

- **Classification Process:**

  1. The system retrieves the most similar source case from the stored case pairs. This can be done with euclidean distance via Nearest Neighbors.
  2. In variant 2 probabilities of the source, target, solution of the target, and the source solution are included. These probabilities could be generated by the neural network model based on the input data.

3. For Variant 1:
    - The system appends pairs of probabilities from the source and target classes.
    - It selects the case pair with the most similar source problem and target problem among those sharing the same source solution.
4. The target solution of the retrieved case pair is utilized for the final classification.

- **Final Outcome:** Similar to Variant 0, and 1, the final classification outcome is determined based on the target solution of the retrieved case pair.

2.2.5 Variant 3 - 5: C-NN-CDH

The variants 3 - 5 for classification stored in the paper [9] add the grouping of case pairs based on their source solutions. Giving multiple specialized adaptation neural networks, that have learned to adapt cases of specific solutions to all solutions. The variants 3 - 5 all share the same grouping and training method but vary in their testing methods. It has been inspired by the ensemble of adaptations for classification approach (EAC) [4].

Variant 3: Single Adaptation Neural Network

Variant 3 introduces a specialized adaptation neural network approach, which predicts the target solution by retrieving the most similar source case using 1-Nearest neighbor and employing one specialized adaptation neural network for adaptation.

Characteristics:

- **Grouping of Case Pairs:** Case pairs are grouped based on their source solutions. The target solutions within a group are not restricted. Meaning there is no constraint on the possible solutions.
- **Training Procedure:** Each specialized adaptation neural network is trained on a specific group of case pairs, those case pairs with the same class outcome, focusing on learning adaptation knowledge where the source solution is determined. The separation of case pairs based on source solutions enables the incorporation of the source solution as a crucial input because it is already known, guiding the selection of the appropriate specialized adaptation neural network.

- **Homogeneous Training:** Training becomes more manageable as each group of case pairs is more homogeneous, and the knowledge to be learned is more specific. This specialization improves the adaptation process.

Testing Procedure:

- During the testing phase, the system retrieves the most similar source case for each test case using its nearest neighbor (1-NN).
- It utilizes one specialized adaptation neural network corresponding to the retrieved source case.
- The target solution is predicted based on the alteration performed by the selected specialized neural network.

Variant 4: N Adaptation Neural Network

Variant 4 also has a specialized adaptation neural network approach, which predicts the target solution by retrieving the most similar source case using K-Nearest neighbor and employing one specialized adaptation neural network for adaptation. It's use of (k-NN) gives it an advantage in the number of predictions it can make to then take a vote on the most likely solution.

Characteristics:

- **Grouping of Case Pairs:** Is the same as variant 3.
- **Training Procedure:** Is the same as variant 3.
- **Homogeneous Training:** Is the same as variant 3.

Testing Procedure:

- During the testing phase, the system retrieves the most similar source cases for each test case using (K-NN) K-nearest neighbors (K-NN).
- It utilizes one specialized adaptation neural network corresponding to the retrieved source case.
- The target solutions are created using the adaptation created by the specialized neural network, applied to the retrieved cases.
- A majority vote is taken and returned as the final outcome.

Variant 5: N Adaptation Neural Network, taking N neighbors from all classes

Variant 5 is based off variant 4, its main difference being that it uses N neighbors from all classes. Giving it the use of every specialized neural network.

Characteristics:

- **Grouping of Case Pairs:** Is the same as variant 3.
- **Training Procedure:** Is the same as variant 3.
- **Homogeneous Training:** Is the same as variant 3.

Testing Procedure:

- During the testing phase, the system retrieves the most similar source cases for each test case using (K-NN) K-nearest neighbors (K-NN), for every class. It retrieves k neighbors for each distinctive class.
- It utilizes one specialized adaptation neural network corresponding to the retrieved source case.
- The target solutions are created using the adaptation created by the specialized neural network, applied to the retrieved cases.
- A majority vote is taken and returned as the final outcome.

2.2.6 Evaluation of Learning adaptations for case based classification: A Neural Network approach

Gathering Case Based Pairs

- **Neighboring pairs:** Each source case will be paired with its nearest neighbor.
- **Random pairs:** Each source case is paired with 10 random cases. This results in 10n random pairs.
- **Class-to-Class pairs:** each source case is paired with its nearest neighbor in every other class resulting in n(m - 1) pairs, we exclude that source cases class.

Evaluation Summary

Within the paper [9] the experiments conducted on two different datasets provided answers to several key questions.

The effectiveness of neural network learning adaptation knowledge was demonstrated through consistent performance of one or more C-NN-CDHs, which often outperformed the neural network classifier. Despite the possibility of neural networks learning to discard the source problem and rely solely on the target problem, the non-zero weights associated with the source problem and significant performance differences between C-NN-CDHs and the neural network indicate that C-NN-CDHs effectively learn adaptation knowledge. This suggests that if a neural network can handle the classification task directly, it can also learn adaptation knowledge or the relation between pairs of cases in the task domain.

Surprisingly, variant (1) performed almost identically to variant (2), which also considers the source solution in adaptation. This suggests that the source solution, heavily coupled with the source problem, may not provide additional useful information for adaptation.

In terms of segmenting case pairs by source case solution, variants (1, 2) generally outperformed variants (3, 4) in accuracy on most datasets. This could be because a single adaptation neural network in (1, 2) is well-trained with all pairs of cases, while a specialized adaptation neural network in (3, 4) is trained with segmented examples. However, variants (3, 4) showed faster convergence during training due to training on segmented examples.

The impact of the ensemble of adaptations was investigated, with EAC-NN-CDH (variant (4)) performing similarly to its counterpart variant (3) without ensemble. This suggests that the generalization power of C-NN-CDH produces stable predictions that an ensemble version does not significantly alter.

The usefulness of the class-to-class approach for adaptation was demonstrated by C2C-NN-CDH (variant (5)), which performed differently from and often better than the other C-NN-CDHs. C2C-NN-CDH reaches its prediction by collecting evidence from diverse source cases from all classes, which can provide more global support, especially when there are multiple classes. Moreover, the C2C approach offers the possibility of explanation with contrastive evidence.

Conclusions on Learning adaptations for case based classification: A Neural Network approach

The conclusion of the paper [9] being that (variant (5)) is worth investigating using its C2C-NN-CDH approach to classification. As it yielded the most promising results when compared to all the other variants tested in the paper.

2.3 Learning and Applying Case Adaptation Rules for Classification: An Ensemble Approach

The paper [4] introduces the (EAC) ensemble of adaptations for classification.

Chapter 3

Design

3.1 Linger Regression Model

3.1.1 Objective

The Linger Regression model integrates principles of Case-Based Reasoning (CBR) into regression analysis, aiming to enhance interpretability and performance. The objectives are:

- Develop a dynamic regression model using differences to adapt to varying contexts.
- Ensure compatibility with Scikit-learn for accessibility and ease of implementation.
- Enable effective interpretation of predictions for users.
- Implement the learning from differences algorithm with enhancements for improved performance.

Its intended use case is for regression problems that could benefit from a more transparent learning process.

3.1.2 Architecture

The `LingerRegressor` combines K-nearest neighbors (KNN) and `MLPRegressor` from Scikit-learn for flexibility and accuracy. The philosophy behind the regression model is to implement a model that learns from differences between case pairs.

3.1.3 Components

- KNN module: Computes nearest neighbors and extracts differences during both training and prediction phases.
- `MLPRegressor`: Offers flexibility in activation functions, solvers, and hyperparameters for regression.
- Supplementary Features: Weighted KNN, Context Addition, Additional Results Column, and Duplication Based on Distance.

3.1.4 Parameterization

Extensive parameterization allows fine-tuning of KNN, `MLPRegressor`, and additional features.

3.1.5 Training and Prediction

Training

Consider a dataset D. The dataset D will be broken up into D^{train} and D^{test} with a 80% 20% split.

During the training, with dataset D^{train} Two separate sets (X_i, Y_i) are created, (X_i) contains features, and (Y_i) labels. For every source case in (X_i) its corresponding result in (Y_i) this is the `TrainBaseCase`, it's nearest (K-Neighbors) in D^{train} are retrieved.

The differences between each `TrainBaseCase` and their nearest neighbors are calculated by subtracting the retrieved cases, these are then added to the ΔD^{train} dataset. There will be $M * N$ differences added to the ΔD^{train}, M being the number of items in D^{train} and N being the number of Neighbors.

The neural network is then trained on ΔD^{train}.

Prediction

After a successful training of the neural network. The dataset D^{test} is used for prediction. For every `TestBaseCase` within D^{test}, its K-nearest neighbors are retrieved from D^{train}. The differences between each `TestBaseCase` and their nearest neighbors are calculated by subtracting the retrieved cases from the `TestBaseCase`, these are then added to the ΔD^{train} dataset.

3.1.6 Data Collection and Preprocessing

Details on how training and validation datasets were collected and preprocessed.

3.1.7 Infrastructure

Hardware and software specifications for development and deployment.

3.1.8 Ethical Considerations

Discussion on ethical considerations associated with the Linger Regression model.

3.1.9 Limitations and Future Work

Acknowledgement of limitations and proposals for future improvements.

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] Vahid Jalali, David Leake, and Najmeh Forouzandehmehr. Learning and applying case adaptation rules for classification: An ensemble approach. In *IJCAI*, pages 4874–4878, 2017.

[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.