OpenAIR @RGU ROBERT GORDON UNIVERSITY ABERDEEN

This publication is made freely available under GREEN open access.

AUTHOR(S): MARTIN, K., WIRATUNGA, N., SANI, S., MASSIE, S. and CLOS, J.

TITLE: A convolutional Siamese network for developing similarity knowledge in the SelfBACK dataset.

YEAR: 2017

Publisher citation: MARTIN, K., WIRATUNGA, N., SANI, S., MASSIE, S. and CLOS, J. 2017. A convolutional Siamese network for developing similarity knowledge in the SelfBACK dataset. In Sanchez-Ruiz, A.A. and Kofod-Petersen, A. (eds.) Proceedings of the 25th International case-based reasoning conference (ICCBR 2017): case-based reasoning and deep learning workshop (CBRDL 2017), 26-28 June 2017, Trondheim, Norway. Trondheim: ICCBR [online], pages 85-94. Available from: http://ceur-ws.org/Vol-2028/paper8.pdf

OpenAIR citation: MARTIN, K., WIRATUNGA, N., SANI, S., MASSIE, S. and CLOS, J. 2017. A convolutional Siamese network for developing similarity knowledge in the SelfBACK dataset. In Sanchez-Ruiz, A.A. and Kofod-Petersen, A. (eds.) Proceedings of the 25th International case-based reasoning conference (ICCBR 2017): case-based reasoning and deep learning workshop (CBRDL 2017), 26-28 June 2017, Trondheim, Norway. Trondheim: ICCBR, pages 85-94. Held on OpenAIR [online]. Available from: https://openair.rgu.ac.uk

Publisher copyright statement:

This is the AUTHOR ACCEPTED version of proceedings originally published by ICCBR (ORGANISERS) and presented at 25th International case-based reasoning conference (ICCBR 2017): case-based reasoning and deep learning workshop (CBRDL 2017), 26 June 2017, Trondheim, Norway. (ISBN _____; eISBN _____; ISSN 1613-0073).

OpenAIR takedown statement:

Section 6 of the "Repository policy for OpenAIR @ RGU" (available from http://www.rgu.ac.uk/staff-and-current-students/library/library-policies/repository-policies) provides guidance on the criteria under which RGU will consider withdrawing material from OpenAIR. If you believe that this item is subject to any of these criteria, or for any other reason should not be held on OpenAIR, then please contact openair-help@rgu.ac.uk with the details of the item and the nature of your complaint.

This publication is distributed under a CC BY-NC 4.0 license.

https://creativecommons.org/licenses/by-nc/4.0

OpenAIR at RGU Digitally signed by OpenAIR at RGU DN: cn=OpenAIR at RGU, o=Robert Gordon University, ou=Library, email=publications@rgu.ac.uk, c=GB Date: 2018.07.02 14:59:59 +01'00'

A Convolutional Siamese Network for Developing Similarity Knowledge in the SelfBACK Dataset

Kyle Martin, Nirmalie Wiratunga, Sadiq Sani, Stewart Massie, Jeremie Clos

School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{ k.martin | n.wiratunga | s.sani | s.massie | j.clos }@rgu.ac.uk

Abstract: The Siamese Neural Network (SNN) is a neural network architecture capable of learning similarity knowledge between cases in a case base by receiving pairs of cases and analysing the differences between their features to map them to a multi-dimensional feature space. This paper demonstrates the development of a Convolutional Siamese Network (CSN) for the purpose of case similarity knowledge generation on the SelfBACK dataset. We also demonstrate a CSN is capable of performing classification on the SelfBACK dataset to an accuracy which is comparable with a standard Convolutional Neural Network.

Keywords: Case-Based Reasoning · Siamese Neural Networks · Categorisation · SelfBACK

1 Introduction

Similarity knowledge is an essential component of an effective Case-Based Reasoning (CBR) system, but its generation can be a daunting task. Large complex datasets, where inter-feature relationships may exist, present a challenge to traditional similarity generation measures. Although similarity-based retrieval can offer numerous advantages during the retrieval phase, this can have a large initial cost. It is little wonder that recent research is targeting methods of harnessing deep learning methods to improve similarity knowledge generation.

A Siamese Neural Network (SNN) is a deep learning architecture which can learn similarity knowledge at a case-to-case level. SNNs have proven effective at learning similarity knowledge for a range of different domains including smartphone gesture classification and face verification [2, 4]. This paper presents the application of an SNN architecture to the SelfBACK[1] dataset[2], which contains

[1]The SelfBACK project is funded by European Union's H2020 research and innovation programme under grant agreement No. 689043. More details available: http://www.selfback.eu

[2]The SelfBACK dataset associated with this paper is publicly accessible from https://github.com/selfback/activity-recognition

the accelerometer data for 34 users labelled with one of 6 activities. The main contribution of this paper is to demonstrate the successful application of an SNN as a means to develop similarity knowledge within a case base. In addition, this paper demonstrates that an SNN can perform a classification task on a level which is competitive with a typical Convolutional Neural Network (CNN).

This paper is organised into the following sections. Section 2 gives an overview of the research regarding learning similarity measure for use in CBR, as well as the SNN architecture and how it may be used as a means to develop similarity knowledge between cases which can be used for classification. Section 3 contains a description of the SelfBACK dataset and how it was used within the context of the presented research. Section 3 also describes our evaluation and details the setup of our experiments, including our classification method and network architecture, as well as pair creation method. Section 4 contains the results of our experiments and Section 5 highlights further work we aim to complete within this research area.

2 Related Works

2.1 Learning Similarity Measures

Learning effective similarity measures between cases can counter many of the issues that plague the retrieval phase of CBR systems, such as retrieving suitable results from extremely large and complex case bases, or retrieving results for cases where some features cannot be explicitly described [10]. However, the process of learning similarity knowledge can itself present an issue, and as such it has been the focus of much research.

Knowledge-Intensive Similarity Measures (KISMs) have been shown to improve retrieval in case bases where domain-specific knowledge is a key component [11]. While the main intention of standard similarity measures is to numerically quantify the simillitude between two cases based upon explicit feature values, K-ISMs use domain-specific knowledge to weight more important features for return [5]. This has been shown to improve retrieval accuracy in complex domains, and domains that rely on expert knowledge to query. However, the acqusition and encoding of domain-knowledge into similarity measures is an extremely expensive process which can often require the input of a domain expert. One of the advantages of the presented SNN architecture is that it can weight features automatically without the input of a domain expert and is significantly less time-consuming.

2.2 The Siamese Neural Network Architecture

An SNN architecture consists of two neural networks that share identical weights and are joined at one or more layers. SNNs receive case pairs as input to both the training and testing phases to develop simlarity knowledge at an object-to-object level. An example architecture is shown in Figure 1. During the training phase, these pairs are labelled as either 'genuine' (if the examples share the

same class) or 'impostor' (if the examples are of different classes). This allows the network to develop a multi-dimensional space based upon cases features, where 'genuine' pairs are pushed closer together and 'impostor' pairs are pulled further away from each other. The output of the identical neural networks (or 'sub-networks') are feature vectors for each member of the input pair. The distance between these vectors is measured at the similarity layer to ascertain whether they belong to the same class based upon a threshold.

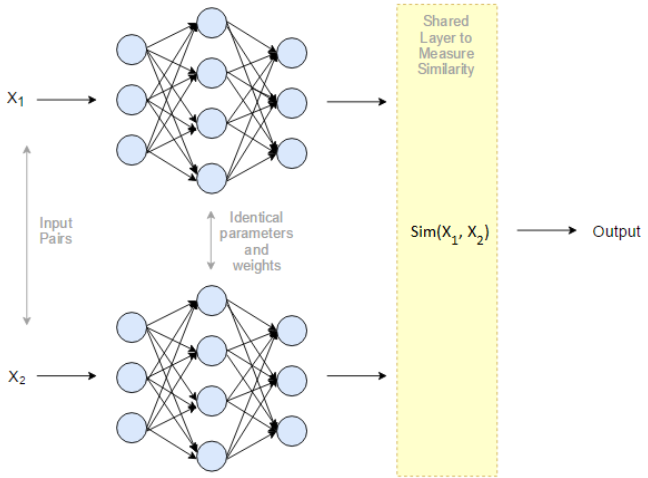

Figure 1: Siamese Neural Network Architecture

SNNs use 'contrastive loss', which was introduced in [4]. Contrastive loss is calculated by summing the results of the individual loss formulas for genuine and impostor pairs. Genuine pairs are penalized by loss L_G for being too far apart, while negative pairs are penalized by L_I if their distance falls within the given margin value. Sub-network weights are then updated by backpropagating the loss with respect to the weights. This means that genuine pairs are pushed closer together over the course of training, whilst ensuring that impostor pairs maintain at least a set distance apart. The similarity metric is therefore directly learned by the network, as it is implicitly defined by the loss function.

The equations for contrastive loss are detailed in Equations (1), (2) and (3). Y_A and Y_P are binary values which are equal to 0 for genuine pairs and 1 for impostor pairs, where Y_A is the actual label, Y_P is the predicted label and M is the margin.

$$L_G = (1 - Y_A){Y_P}^2 \tag{1}$$

$$L_I = Y_A(max(M - Y_P, 0))^2 \tag{2}$$

$$L = L_G + L_I \tag{3}$$

2.3 Similarity and Classification in SNNs

Initially made popular by [4] to identify similarities for face verification, many research efforts have taken advantage of an SNN's capability to develop similarity knowledge, in areas ranging from smartphone gesture classification [2], to similar text retrieval [11]. In [11], the authors demonstrate that their SNN can outperform state-of-the-art text similarity measures by mapping term vectors to a low dimensional space. Their results indicate that the SNN can significantly outperform other methods on both low and high dimensional data. The drawback was that the algorithm did not scale well to large amounts of examples.

Although introduced as a method of signature verification and binary classification [3], recent research has shown that SNNs are able to generalise to multiclass classification. In [6], the authors demonstrate that a Convolutional Siamese Network (CSN) can achieve very close to the state of the art and human levels of recognition in a one-shot learning setting on the omniglot dataset, which contains 40 distinct classes. A CSN is a type of SNN where the parallel neural networks are replaced with two identical Convolutional Neural Networks (CNNs). A CNN itself is a feed forward neural network which arranges its neurons in multiple dimensions in order to operate effectively on high dimensional data. One of the main advantages of using CNNs is that they can learn local feature detectors and are fairly robust to distortions of network input [7].

In [1], the authors demonstrated visual search on multiple domains by performing nearest neighbour on the output feature vectors from an CSN. Their findings showed that CSNs could be used to learn the similarity between images and that a nearest neighbour algorithm could be performed to retrieve the most similar images for a given example. Their findings demonstrated that the feature vectors produced by a CSN have potential use as a means to increase utility of more conventional classification techniques.

We can observe from the range of examples above that SNNs are capable of learning similarity knowledge and performing classification upon a wide range of domains. However, there remain areas which require further exploration. In particular, literature regarding the structuring of pair creation is lacking, as is research which utilises SNNs within ensemble classifiers.

3 Evaluation

The aims of this paper are two-fold; to show that an SNN could generate similarity knowledge within a case base, and to demonstrate the performance of an SNN as a method of human activity classification. To this end, we performed two experiments upon the SelfBACK dataset.

3.1 The Dataset

The SelfBack dataset consists of time series data collected from 34 users performing different activities over a short period of time. Data was collected by

mounting a tri-axial accelerometer on the thigh and right-hand wrist of participants at a sampling rate of 100Hz as they completed a script of set activities, performing each for an average of three minutes [9]. Frequency coefficients were obtained by applying Discrete Cosine Transforms (DCT) and Discrete Fourier Transforms (FFT) to the raw accelerometer data.

Our experiments used the thigh dataset due to time limitations. Data was split into 5 second windows, meaning that there were between 160 and 180 cases per user and 1,500 features per case. This resulted in 6,084 cases of thigh data. These were then labelled as one of 6 activities (standing, upstairs, downstairs, walking, jogging, sitting) to create the full dataset.

3.2 Experimental Setup

Firstly, we implemented a Convolutional Siamese Network (CSN) upon the raw accelerometer thigh data, DCT thigh data and FFT thigh data. Pairs of cases were fed into the CSN and the convolutional sub-networks of the CSN learned to produce representative feature vectors of each case. We tested that the network had learned feature vectors which were representative of the original cases by measuring the euclidean distance between pair vectors at the similarity layer and comparing this to a threshold to identify whether a pair of examples belonged to the same class (a genuine pair) or to different classes (an impostor pair). If we identified that the distance between the genuine pair was less than a certain threshold (i.e. the space that should exist between cases of opposing classes) and the distance between the impostor pair was greater than this threshold, then we could reasonably assume that the case had been mapped to the correct space (or a very close approximation of it). We therefore used the percentage of correctly identified pairs as our accuracy metric.

Secondly, we implemented a CSN and CNN to perform classification on the raw thigh accelerometer data. The raw data was used in order to demonstrate a comparison between the two architectures which was unaffected by preprocessing of the data. For this experiment, we completed a similar process to the previous experiment until the CSN had learned representative feature vectors for each case in the test set. Each feature vector from the test set was then compared with 6 randomly selected class representative vectors generated from cases in the training set. The distance between the unlabelled test vector and each class representative vector was measured, and the test case was identified as belonging to the same class as the nearest class representative vector. The accuracy of the experiment was the percentage of correctly classified cases.

3.3 Network Architecture

A CSN was constructed from 2 sub-networks, which had 2 convolutional layers, a flattening layer and 2 fully connected layers. The first layers used tanh activation functions, while the final layer used a softmax function. The network was optimised using Stochastic Gradient Descent and the hyperparameters in Table 1. The output of the sub-networks was a representative feature vector for each

case member of the pair. Euclidean distance between these vectors could then be measured and compared with the threshold.

Parameter	Value
Learning Rate	0.01
Learning Decay	0.000001
Nesterov Momentum	0.9

Table 1: CSN Hyperparameter Settings

In order to be comparable with the CSN, the CNN created for comparison purposes was a close replica of one of the sub-networks outlined above. The only major difference was the use of categorical cross-entropy for the loss function and backpropagating this loss with respect to the weights.

Implementations were run for10 epochs as the loss had reached a sufficiently low value by this point. Experiments were repeated 5 times and a mean percentage of accuracy calculated. There were many random elements at numerous stages of all implementations and so running for multiple iterations and taking the mean of the result accuracy was the only method to ensure that results were legitimately indicative of network performance.

3.4 Splitting the Dataset into Training and Testing

Data was split between train and test sets using leave-p-out cross-validation (LPOCV), meaning that the test set comprised of cases from p users, while all remaining users made up the training set. LPOCV was used due to the real-world constraints of the SelfBACK project, which involves being able to identify users' activities based upon their similarity to other users. This offered an improvement in accuracy over randomly splitting the dataset, though the larger training set caused minor overfitting. Experiments were completed with p set to 5 and 10, to test the effect that increasing test set size had on results.

At run time, the dataset is normalised using standard normal distribution, Equation (4), where μ is the mean of the training set and σ is the standard deviation of the training set. These values were taken from the training set because the test data represented a population of unknown size and distribution.

$$x \in X \mid \frac{x - \mu}{\sigma} \tag{4}$$

3.5 Pair Creation

Pair creation in all experiments was completed after the data had been split into training and test sets to ensure that there was no cross contamination which could effect the results. For pair creation, we defined d as the number of cases

in the full dataset and p as the number of cases to be left out for testing. The training set therefore contained $n = d - p$ cases.

Initially, we attempted to exhaustively create all pairs by matching every case with every other case. However, this resulted in the creation of $\sum_1^d$ cases, which made pair formation extremely slow. Instead, two pairs were created for every case in the dataset; a genuine pair (with a random case of the same class) and an impostor pair (with a random case of a different class). This meant that every case was represented at least twice. We enforced equal genuine and impostor pair creation because generating truly random pairs led to an imbalance of more impostor pairs than genuine, at a ratio of approximately 5:1, and had a negative effect on classification. The number of training and test pairs were therefore $2n$ and $2p$ respectively.

4 Results

4.1 Learning Similarity Knowledge with a CSN

The CSN was able to develop good similarity knowledge for all three time and frequency representations of the thigh dataset, though best results were obtained from the thigh DCT data. This is indicated by the percentage of correctly identified pair relationships, which is shown in Table 2.

SelfBACK Thigh Data	Test User Set	Pair Identification Accuracy
Raw	5	93.57
DCT	5	**94.33**
FFT	5	93.87
Raw	10	92.17
DCT	10	**94.30**
FFT	10	93.00

Table 2: CSN Pair Identification Accuracy on the Thigh Dataset

Even the minimum result of 92.17% obtained on the raw thigh dataset using L10OCV demonstrates that more than 92% of test cases have been mapped to appropriate feature vectors. With this in mind, distance between these vectors can act as a proxy for similarity measurements at a case-to-case level. These results support the argument that cases of the same class are grouped closer together within the feature space and lend evidence to the idea that the CSN can be used to form the basis for similarity-based retrieval in a CBR system.

4.2 Comparing a CSN and CNN on the SelfBACK Dataset

As a classifier, the CSN did not perform as well as the CNN, although it acheived over 90% classification accuracy on both experiments. Although the CSN per-

formed competitively on the L5OCV, the CNN displayed much higher accuracy on the L10OCV experiments, as shown in Table 3.

Architecture	Test User Set	Classification Accuracy
CSN	5	90.75
CNN	5	**91.77**
CSN	10	90.03
CNN	10	**92.60**

Table 3: CSN and CNN Comparison on Human Activity Classification

These results indicate that the CSN requires more training data to be able to classify cases than a typical CNN does. The low variance in results across iterations, and resistance to increasing test set size, by both architectures, supports the idea that they generalise well even to large test sets. Although the CSN did not perform as well at classifying cases as the CNN, we argue that the generation of similarity knowledge as a by-product of the classification process is a non-negligible contribution. The main benefit of using the CSN over a traditional CNN implementation is that the output of the CSN produces feature vectors of the original case base which are good representations of how each case fits into the case base as a whole and allows direct, accurate distance measurements as a proxy for measuring similarity between cases.

On reflection, there may be a couple of reasons that the CSN did not perform as strongly on the classification task as the CNN. A more structured method of classifying the output test feature vector, such as exhaustive k-nn sorting or informed class representative selection, could potentially offer better classification results and may be worth further study. It is a distinct possiblity that the class representative which was randomly selected for comparison with the test vector was a poor representative of the class, and that may have influenced the classification of test cases. Exhaustively comparing the test case with all training cases, or using a method of selection to pick class representatives may improve classification accuracy.

5 Conclusion and Further Work

We have demonstrated that a CSN is capable of learning similarity knowledge on the SelfBACK dataset. In addition, we have demonstrated that a CSN can use this similarity knowledge to perform human activity classification on the SelfBACK dataset and can perform competitively with a CNN on this task.

In future work we would like to further explore SNN's capacity to develop simlarity knowledge between cases in order to determine whether this could be used in some manifestation to develop similarity knowledge between features. Our end goal is to use this similarity knowledge in order to populate the simi-

larity arcs which exist between information entities in a Case Retrieval Network (CRN) [6]. If this could be applied, it would offer an inexpensive method to develop efficient coverage of extensive case bases and reduce the initial cost often associated with CRNs in this task. In addition, it would be interesting to explore different methods of pair generation and different methods of utilising the similarity knowledge generated by a CSN for classification.

References

[1] Bell, S. and Bala, K. Lerning Visual Similarity for Product Design with Convolutional Neural Networks. In: Proceedings of the 42nd International Conference and Exhibition on Computer Graphics and Interactive Techniques, SIGGRAPH 2015. Los Angeles, CA, USA. 9 - 13 August 2015. New York, NY, USA: ACM

[2] Berlemont, S., Lefebvre, G., Duffner, S. and Garcia, C. 2015. Siamese Neural Network Based Similarity Metric for Inertial Gesture Classification and Rejection. In: 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015. Ljubljana, Slovenia. 4 - 8 May 2015. Red Hook, NY, USA: Curran Associates, Inc

[3] Bromley, J., Guyon, I., LeCun, Y., Sackinger, E. and Shah, R. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. International Journal of Pattern Recognition and Artificial Intelligence, Vol 7(4) pp. 669 - 688.

[4] Chopra, S., Hadsell, R. and LeCun, Y. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Patter Recognition, CVPR 2005. San Diego, CA, USA. 20 - 25 June 2005. Washington, DC, USA: IEEE Computer Society

[5] Dalal, S., Athavale, V. and Jindal, K. 2011. Case Retrieval Optimisation of Case-Based Reasoning Through Knowledge-Intensive Similarity Measures. International Journal of Computer Applications, Vol 34(3) pp. 12 - 18

[6] Koch, G., Zemel, R. and Salakhutdinov, R. 2015. Siamese Neural Networks for One-Shot Learning. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015. Lille Grand Palais, Lille. 6 âĂŞ 11 July 2015. Red Hook, NY, USA: Curran Associates, Inc

[7] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, Vol 86(11) pp. 2278 - 2324

[8] Lenz, M. and Burkhard, H. D. 1996. Case Retrieval Nets: Basic Ideas and Extensions. In: Proceedings of the 20th Annual German Conference on Artificial Intelligence, KI-96. Dresden, Germany. 17 - 19 September 1996. Berlin, Germany: Springer-Verlag

[9] Sani, S., Wiratunga, N., Massie, S. and Cooper, K. 2016. SELFBACK - Activity Recognition for Self-management of Low Back Pain. In: Proceedings of the 36th SGAI International Conference on Artificial Intelligence, AI-2016. Cambridge, England. 13 - 15 December 2016. Cham, Switzerland: Springer Nature

[10] Stahl, A. 2006. Combining Case-Based and Similarity-Based Product Recommendation. In: Proceedings of the 8th European Conference of Advances in Case-Based Reasoning, ECCBR 2006. Fethiye, Turkey. 4 - 7 September 2006. Berlin, Germany: Springer-Verlag

[11] Stahl, A. and Gabel, T. 2006. Optimising Similarity Assessment in Case-Based Reasoning. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI-06. Boston, Massachusetts. 16 - 20 July 2006. Berlin, Germany: Springer-Verlag

[12] Yih, W. et al. 2011. Learning Discriminative Projections for Text Similarity Measures. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CONLL 11. Portland, OR, USA. 23 - 24 June 2011. Stroudsburg, PA, USA: ACL