Human Activity Recognition with Deep Metric Learners

Kyle Martin[0000−0003−0941−3111] ✉ and Anjana Wijekoon[] ✉ and Nirmalie Wiratunga[0000−0003−4040−2496]

Robert Gordon University, Aberdeen, Scotland
{k.martin, a.wijekoon, n.wiratunga}@rgu.ac.uk

Abstract. Establishing a strong foundation for similarity-based return is a top priority in Case-Based Reasoning (CBR) systems. Deep Metric Learners (DMLs) are a group of neural network architectures which learn to optimise case representations for similarity-based return by training upon multiple cases simultaneously to incorporate relationship knowledge. This is particularly important in the Human Activity Recognition (HAR) domain, where understanding similarity between cases supports aspects such as personalisation and open-ended HAR. In this paper, we perform a short review of three DMLs and compare their performance across three HAR datasets. Our findings support research which indicates DMLs are valuable to improve similarity-based return and indicate that considering more cases simultaneously offers better performance.

Keywords: Human Activity Recognition · Deep Metric Learning · Deep Learning · Metric Learning · Matching Networks

1 Introduction

Establishing a strong foundation for similarity-based return is a top priority in Case-Based Reasoning (CBR) systems. Without a firm understanding of the similarity between cases, CBR systems are poorly placed to offer solutions from previous knowledge and become increasingly reliant on their adaptation component [2]. With this in mind, it is reasonable to claim that a strong similarity component can be central to the success of a CBR system.

Deep Metric Learners (DMLs) are a group of neural network architectures which learn to optimise case representations for similarity-based return. This is achieved by training on multiple cases simultaneously. Early examples of DMLs trained upon pairs of input cases [1], while more recent learners use triplets [3] or a representative from each cluster [15] to better incorporate contextual knowledge of the feature space. Due to their lack of reliance on class knowledge, DML algorithms are traditionally applied to matching problems with potentially very large numbers of classes, such as signature verification [1] or face re-identification [13]. However, recent research has demonstrated that DMLs have great potential within the Human Activity Recognition (HAR) domain. Feature representations learnt with Siamese Networks produced better classification results in [8]. Well established robustness in Matching Networks [15] when tackling few-shot or one-shot learning problems [5,15] was exploited in achieving personalisation of HAR [12] efficiently. Also recent work highlighted strong performance

to support Open-ended HAR [17]. It is challenging in the domain of HAR to collect large datasets with sensory equipment, it is also desirable learn personal nuances of human activities with the limited amount of data available. Learning feature representations that are similarity driven holds significance as it enables incorporating personal nuances with limited number of cases.

Despite their growing impact on this domain, there remains a lack of literature examining the quality of representations obtained by different DML architectures for a HAR task. With this in mind, we are motivated to do a review of the performance of three DMLs (Siamese Neural Network, Triplet Network and Matching Network) across three HAR datasets (SelfBACK, PAMAP2 and MEx). The contributions of this paper are therefore as follows: (1) we offer an introductory review of three DML architectures and (2) present a comparison of their performance on three HAR datasets.

The paper is structured in the following manner. In section 2 we provide an introductory review of each of the DML architectures that we consider in this paper. In section 3 we detail the experimental setup of our comparative evaluation and give details of the three HAR tasks, while in section 4 we analyse and discuss the results. Finally in section 5 we present some conclusions.

2 Deep Metric Learners

In this section we explore the unique traits of several DML architectures; Siamese Neural Networks, Triplet Networks and Matching Networks. Though individual architectures possess distinct nuances, there are several themes which are consistent across DMLs. With this in mind, let us introduce some general notation used throughout this paper. Let $\mathcal{X}$ be a set of labelled cases, such that for $x \in \mathcal{X}$, the function $y(x)$ returns the class label, y, of case x. In the context of this paper, we will define matching cases as those which have the same class while non-matching cases will have differing classes. The embedding function θ is an appropriate parameterisation of any function used to create the vectorised representation of a given x, while the function D_W represents an arbitrary metric function to measure the distance between two vector representations.

2.1 Siamese Neural Networks

Siamese Neural Networks (SNN) are deep metric learners which receive pairs of cases as input. The SNN architecture consists of two identical embedding functions, enabling the SNN to generate a multi-dimensional embedding for each member of a pair. Input pairs are labelled as either matching or non-matching respectively. Correspondingly, the objective of training is to minimise the distance between the generated embeddings for matching pair members while maximising the distance between embeddings for non-matching pair members. Thus the overall goal of the network is the development of a space optimised for similarity-based return.

To achieve this goal, each training pair, $p \in \mathcal{P}$, consists of two cases from the training set, $p = (\hat{x}, \ddot{x})$. Whether the pair is matching or non-matching is governed by the relationship of the pivot case, $\hat{x}$, to the passive case, $\ddot{x}$. In the context of this work, the pair's relationship class is established by comparing class labels of its members

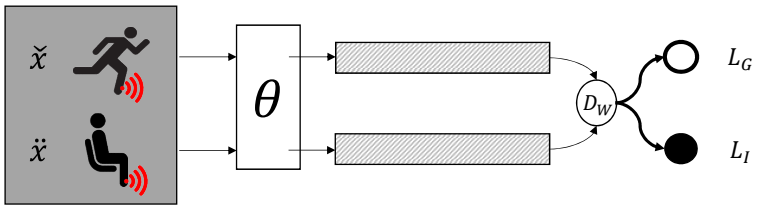


Fig. 1: Siamese Neural Network Architecture

(e.g. $y(\hat{x})$ with $y(x')$). For this we use function $Y(p)$, which returns p's relationship class label, such that $Y(p) = 0$ to symbolise a matching pair when $y(\hat{x}) = y(\ddot{x})$, and $Y(p) = 1$ to symbolise a non-matching pair when $y(\hat{x}) \neq y(\ddot{x})$.

After input to a network, the generated embeddings for each member of a pair can be compared using a distance metric, $D_W(\theta(\hat{x}), \theta(\ddot{x}))$. This distance metric plays a key role in the unique contrastive loss used by SNN (as in Equation 3); it penalises members of matching pairs until they occupy the exact same spot in the space (using L_G in Equation 1) and penalises members of non-matching pairs until they exist at least a set margin distance of α apart (using L_I in Equation 2) to calculate error for model update (see Figure 1). The error is then backpropagated over both embedding functions to ensure they remain identical.

$$L_G = (1 - Y_A) \cdot D_W(\theta(\hat{x}), \theta(\ddot{x}))^{\,2} \tag{1}$$

$$L_I = Y_A \cdot (max(0,\ \alpha - D_W(\theta(\hat{x}), \theta(\ddot{x}))))^2 \tag{2}$$

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

Using both errors means the similarity metric can be directly learned by the network through the comparison of the actual pair label Y_A (which, as above, is equal to 0 for matching and 1 for non-matching pairs respectively) and the distance between pair members, while using the generated embeddings of pair members during distance comparisons ensures iterative model refinement. It is also these learned embeddings which act as the improved representation for similarity-based return after training is complete.

Designed for matching tasks such as signature verification [1] or similar text retrieval [9], SNNs generalise well to classification tasks when supported by a similarity-based return component such as k-Nearest Neighbour (kNN). Research into SNNs highlighted their capacity to support one-shot learning [5], an area of research where recent innovations on deep metric learning such as Matching Networks still demonstrate state-of-the-art results [15,17].

2.2 Triplet Networks

Triplet networks (TN) are DMLs which learn from three input cases simultaneously. Together described as a triplet, these inputs are the anchor case (x^a), a positive case (x^+) and a negative case (x^-). The anchor case acts as a point of comparison, meaning that the positive and negative cases are dictated by their relationship to the anchor (i.e.

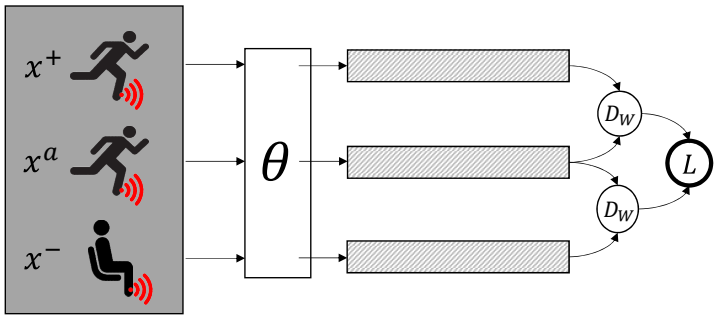


Fig. 2: Triplet Network Architecture

matching and not matching respectively). Similar to SNN, the objective during training is to minimise the distance between an anchor and its associated positive case and maximise the distance between an anchor and its associated negative case. However, considering three cases at once ensures that update of weights is more focused. This is because SNN are learning based on only one aspect at any given time (e.g. either pair members are alike, or not), meaning that more pairs are required to build the full picture. Considering three cases at once allows the triplet network to better understand the context of the anchor case.

A triplet network is comprised of three identical embedding functions, each of which creates an embedding for one input (see Figure 2) before the error is calculated using triplet loss:

$$L = D_W(\theta(x^a), \theta(x^+)) - D_W(\theta(x^a), \theta(x^-)) + \alpha \tag{4}$$

Like contrastive loss (see Equation 3), triplet loss is a distance based function. The formula will generate a loss value in situations where the distance between the anchor case and the negative case, $D_W(\theta(x^a), \theta(x^-))$, is less than the distance between the anchor case and the positive case, $D_W(\theta(x^a), \theta(x^+))$. The network is therefore penalised until matching cases are closer than non-matching cases. A minimum boundary between non-matching cases is enforced by the margin α.

Unlike SNNs, TNs were designed to be supported by a similarity-based return component [3] and cannot perform classification tasks on their own. This means that despite common use in matching problems such as facial recognition [6,13] or image-based search [16], TNs are very capable of establishing an effective basis for similarity-based return on multi-class problems [7]. Though convergence of these networks can be achieved through creation of random triplets, recent work has demonstrated that a training strategy which optimises triplet creation can improve training efficiency [13,16].

2.3 Matching Networks

Matching Networks (MNs) [15] are unique in that they an be used flexibly as either a classifier or a DML. MNs learn to match a query case to members of a support set

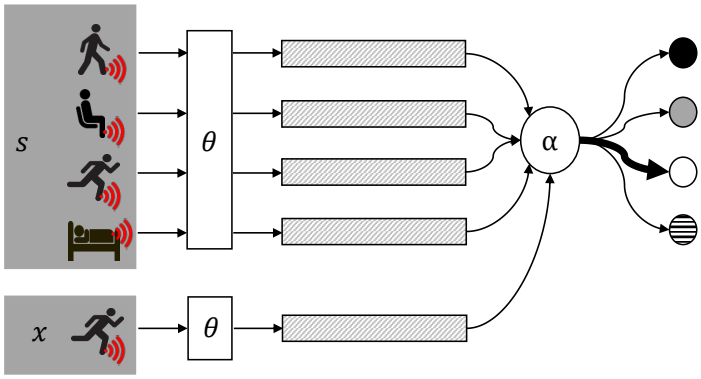


Fig. 3: Matching Network Architecture

(S) which contains both matching and non-matching cases. Throughout training the network iteratively updates weights of the feature embedding function (θ) to maximise the pair similarity between the query case and the corresponding match in the support set (see Figure 3). Originally designed as a classification algorithm, S is conventionally populated with one or more representatives of each class. However, MNs can be trivially adapted to deep metric learning by considering cluster representatives to populate S. Each cluster can have more than a single representative, meaning that the cardinality of S is equal to $k \times n$; where k is the number of representatives per cluster and n is the number of clusters in the dataset. The similarity between the query case and the support set case pairs (x', x'_i) are calculated with a suitable similarity metric ($D_W(x', x'_i)$), and an attention mechanism in the form of weighted majority vote estimates the class distribution (see Equation 6 and 7). This is enforced by the loss function categorical cross-entropy, which quantifies the difference between the estimated and actual distributions.

$$\theta(x) = x' \tag{5}$$

$$a(x', x'_i) = \frac{e^{D_W(x', x'_i)}}{\sum^{|s|} e^{D_W(x', x'_i)}} \tag{6}$$

$$\hat{y} = \sum^{|S|} a(x', x'_i) \times y \tag{7}$$

MN was first applied in the domain of one-shot and few-shot learning with image recognition [15]. Prototypical Networks by [14] is a adaptation of MN applied in the same domain. Here the model creates a prototype (by averaging over similar elements in the support set) for each class in the support set, then behaves as a one-shot learning MN model that out performed original MN in few-shot learning. More recently, MN was exploited successfully to achieve personalised HAR[12] and Open-ended HAR [17] where they successfully utilise support set to enforce personal traits of human activities.

3 Evaluation

In this section, we offer details of our evaluation of three deep metric learners - SNN, TN and MN. We perform an empirical comparison of the representations gained from each network architecture to determine their effectiveness within the HAR domain. For both SNN and TN we use k-NN accuracy as a proxy for representation goodness, while for MN we use classification accuracy from the model's final parametric comparison between query and support set to indicate the quality of representations. We evaluate this by performing a one-tail t-test to establish statistical significance at a confidence level of 95% on classification accuracy.

3.1 Datasets

SelfBACK [11] features time series data [1] collected with 34 users each performing 9 different ambulatory and sedentary activities. Activities include jogging, lying, sitting, standing, walking downstairs, walking upstairs, walking in slow, medium or fast pace. Data was collected by mounting a tri-axial accelerometer on the right thigh and right-hand wrist of participants and data was recorded at a sampling rate of 100Hz for 3 minutes.

MEx (Multi-modal Exercise Dataset) [2] dataset is a Physiotherapy exercise dataset that contains data collected using a pressure mat, a depth camera and two accelerometers placed on the wrist and the thigh. Data was recorded for 7 exercises with 30 users. Each user performed one exercise for a maximum of 60 seconds. Seven exercises included in the dataset are Knee rolling, Bridging and Pelvic tilt, The Clam, Extension in lying, Prone Punches and Superman. The pressure mat, the depth camera and the accelerometers data record frequencies are $75Hz$, $15Hz$ and $100Hz$ respectively. In this work we only work with the data from two accelerometers for the purpose of comparability with other datasets.

PAMAP2 [10] is a Physical Activity Monitoring dataset [3] which contains data from 3 IMUs located on wrist, chest and ankle. Data was recorded with 9 users approximately at $9Hz$ for 18 activity classes by following a pre-defined protocol. Activities include that are ambulatory, sedentary and activities of daily living. One user and 10 activities were filtered out of this dataset due to insufficient data. The refined dataset contained 8 users and 8 activity classes.

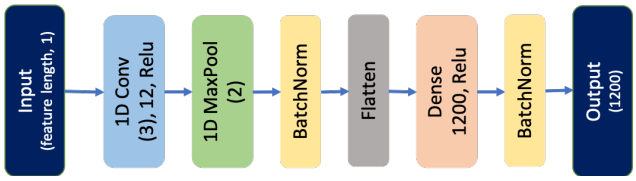
3.2 Data Pre-processing

The following pipeline was applied on all datasets as pre-processing and to form cases by progressively converting a raw signal to a Discrete Cosine Transformation (DCT) feature vector.

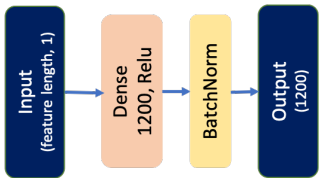
[1] Public dataset available at https://github.com/rgu-selfback/Datasets

[2] Public dataset available at https://data.mendeley.com/datasets/p89fwbzmkd/1

[3] Public dataset available at http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring



(a) Convolutional θ



(b) Dense θ

Fig. 4: Embedding θ

1. Use a sliding window of 500 timestamps to segment the original raw sensor signal. (no overlap for SelfBACK and PAMAP2, 2 second overlap for MEx)
2. Extract 3-dimensional (x, y, z) raw accelerometer data from each sensor.
3. Apply DCT and extract most significant 60 features from each dimension.
4. Concatenate all DCT feature vectors from each dimension of all sensors to form the final feature vector. Lengths of resulting feature vectors for SelfBACK, PAMAP2 and MEx are 360, 540 and 360 respectively.

3.3 Experimental Setup

We define a nearest-neighbour classification task for all three HAR datasets - SelfBACK, PAMAP2 and MEx. We use Leave-One-Person-Out (LOPO) validation with each dataset and perform 34, 8 and 30 experiments for SelfBACK, PAMAP2 and MEx respectively. We record the accuracy of each experiment and present mean value as the performance metric.

All feature embedding functions (Figure 4a and Figure 4b) used the ReLU activation and were trained using the Adam optimiser [4]. We implemented an empirical evaluation to identify the best performing hyper-parameters for all datasets (see Table 1). These parameters are kept constant across all experiments, meaning that SNN, TN and MN all used the same embedding function to allow fair comparison. To ensure the comparability between DMLs, we also enforced random pair, triplet and subset creation. Each training case was represented within two pairs (one matching and one non-matching), a single triplet and a single query for comparison to a subset respectively. This was to mitigate the fact that TNs and MNs inherently consider more information than SNNs. In future, it would be interested to do a co-ordinated examination of the networks invoking pair, triplet or subset mining strategies.

	Parameter	Value
General	Batch size	60
	Training epochs	10
	Output length - feature embedding function	1200
	Number of neighbours for kNN	3
TN/ SNN	Mini-batch size	200
	α	TN- 1, SNN- 15
MN	Samples per class	1
	Classes per set	number of classes in the dataset
	Training cases	$500 \times number\ of\ users$

Table 1: Hyper-parameters

4 Results

Each dataset was evaluated against the kNN algorithm which sets the baseline for DML algorithms. Table 2 presents the comparative results we obtained with algorithms detailed in the previous section. An asterisk indicates that the performance improvement is statistically significant at 95% confidence interval compared to the kNN baseline and we highlight the best performing algorithm with bold text.

It is evident that the representations learned with DML algorithms are better for similarity-based return than the raw representation. With a single exception, all representations learned with DMLs outperform the baseline. In the SelfBACK dataset, all DML algorithms significantly outperform kNN baseline with both MLP and CNN feature embedding functions. MN algorithm achieves the best performance and the MLP embedding function earns a 1.34% performance improvement over CNN embedding function. Similarly, results on the MEx dataset show that all DML algorithms significantly outperform the kNN baseline and the MN algorithm achieves the best performance. Unlike the SelfBACK results, CNN feature embedding function edge out MLP embedding function with a performance improvement of 1.30%. These results suggest that considering more cases simultaneously is advantageous to the algorithm. Furthermore, the choice of MLP or CNN to operate as an embedding function seems to be problem specific.

We observe a distinct difference on the PAMAP2 dataset where TN algorithm with CNN feature embedding function outperforms all DML algorithms and baseline. In addition, the MN algorithm performs the poorest and does not significantly outperform the baseline. This dataset is comparatively smaller compared to other two datasets which can be a contributing factor for the reduced performance with MN algorithm. It is interesting that the TN algorithm significantly outperform the baseline even with limited training data. We plan to explore this insight further in future work.

Overall the results are indicative that deep metric learning methods can learn an effective feature representation for similarity calculations. Most importantly, the im-

Dataset	Network	Accuracy(%)		
		Raw	MLP	CNN
SelfBACK	kNN	76.70	-	-
	SNN	-	80.41*	81.34*
	TN	-	80.60*	81.57*
	MN	-	**88.35***	87.01*
MEx	kNN	68.56	-	-
	SNN	-	77.27*	72.31*
	TN	-	79.32*	79.27*
	MN	-	94.19*	**95.49***
PAMAP2	kNN	81.40	-	-
	SNN	-	82.14	85.88*
	TN	-	86.80*	**87.32***
	MN	-	80.72	81.77

Table 2: Summary of results

provements to performance were statistically significant in almost every circumstance. This is an important insight for when using similarity based algorithms where we previously relied on hand-crafted feature engineering.

5 Conclusion

In this paper we review three DML algorithms to learn feature representations and evaluate them against hand-crafted feature representations in a similarity based classification task. We select HAR as the classification task as similarity based classification holds special significance in improving performance by understanding personal nuances. Our results show that the feature representations learnt with DML algorithms significantly outperform hand-crafted feature representations in the selected domain. These results highlight the potential of DML algorithms to create effective feature representations efficiently, which is crucial in domains such as case-based reasoning. In future we plan to extend this review to other domains and compare further aspects of DMLs, such as the performance of different case mining approaches.

References

1. Bromley, J., Guyon, I., LeCun, Y.: Signature verification using a 'siamese' time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7**(4), 669 – 688 (August 1993). https://doi.org/doi:10.1142/S0218001493000339
2. Ganesan, D., Chakraborti, S.: An empirical study of knowledge tradeoffs in case-based reasoning. In: Proceedings of the Twenty-Seventh International Joint Conference on Ar-

tificial Intelligence, IJCAI-18. pp. 1817–1823. International Joint Conferences on Artificial Intelligence Organization (7 2018). https://doi.org/10.24963/ijcai.2018/251, https://doi.org/10.24963/ijcai.2018/251
3. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) Similarity-Based Pattern Recognition. pp. 84 – 92. Springer International Publishing, Cham (2015)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Deep Learning Workshop. ICML '15 (July 2015)
6. Liao, W., Ying Yang, M., Zhan, N., Rosenhahn, B.: Triplet-based deep similarity learning for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 385–393 (2017)
7. Liu, Y., Huang, C.: Scene classification via triplet networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(1), 220–237 (Jan 2018). https://doi.org/10.1109/JSTARS.2017.2761800
8. Martin, K., Wiratunga, N., Sani, S., Massie, S., Clos, J.: A convolutional siamese network for developing similarity knowledge in the selfback dataset. p. 85–94 (2017)
9. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In: Rep4NLP@ACL (2016)
10. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: Wearable Computers (ISWC), 2012 16th International Symposium on. pp. 108–109. IEEE (2012)
11. Sani, S., Wiratunga, N., Massie, S., Cooper, K.: Selfback - activity recognition for self-management of low back pain. In: Research and Development in Intelligent Systems XXXIII. pp. 281 – 294. SGAI '16, Springer Nature, Cham, Switzerland (December 2016)
12. Sani, S., Wiratunga, N., Massie, S., Cooper, K.: Personalised human activity recognition using matching networks. In: Cox, M.T., Funk, P., Begum, S. (eds.) Case-Based Reasoning Research and Development. pp. 339–353. Springer International Publishing, Cham (2018)
13. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 815 – 823. CVPR '15, IEEE Computer Society, Washington, DC, USA (June 2015). https://doi.org/doi:10.1109/cvpr.2015.7298682
14. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. pp. 4077–4087 (2017)
15. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. pp. 3630–3638 (2016)
16. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: Proc. of the 2014 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 1386 – 1393. CVPR '14, IEEE Computer Society, Washington, DC, USA (June 2014). https://doi.org/doi:10.1109/cvpr.2014.180
17. Wijekoon, A., Wiratunga, N., Sani, S.: Zero-shot learning with matching networks for open-ended human activity recognition. (2018)