A Neural Network Approach to Similarity Learning

Stefano Melacci, Lorenzo Sarti, Marco Maggini, and Monica Bianchini

DII, Università degli Studi di Siena
Via Roma, 56 — 53100 Siena (Italy)
{mela,sarti,maggini,monica}@dii.unisi.it

Abstract. This paper presents a novel neural network model, called similarity neural network (SNN), designed to learn similarity measures for pairs of patterns. The model guarantees to compute a non negative and symmetric measure, and shows good generalization capabilities even if a very small set of supervised examples is used for training. Preliminary experiments, carried out on some UCI datasets, are presented, showing promising results.

1 Introduction

In many pattern recognition tasks, appropriately defining the distance function over the input feature space plays a crucial role. Generally, in order to compare patterns, the input space is assumed to be a metric space, and Euclidean or Mahalanobis distances are used. In some situations, this assumption is too restrictive, and the similarity measure could be learnt from examples.

In the last few decades, the perception of similarity received a growing attention from psychological researchers [1], and, more recently, how to learn a similarity measure has attracted also the machine learning community. Some approaches are proposed to compute iteratively the similarity measure, solving a convex optimization problem, using a small set of pairs to define the problem constraints [2,3]. Other techniques exploit EM–like algorithms [4], Hidden Markov Random Fields [5], and constrained kernel mappings [6,7]. However, the existing approaches are generally strictly related to semi–supervised clustering, and the presence of some class labels or pairwise constraints on a subset of data is exploited to improve the clustering process.

In this paper a similarity learning approach based on SNNs is presented. The SNN architecture guarantees to learn a non negative and symmetric function, and preliminary results, carried out using some UCI datasets, show that the generalization performances are promising, even if a very small set is used for training.

The paper is organized as follows. In the next section, the network architecture and its properties are presented. In Section 3 some experimental results are reported, comparing them with commonly used distance functions. Finally, some conclusions are drawn in Section 4.

L. Prevost, S. Marinai, and F. Schwenker (Eds.): ANNPR 2008, LNAI 5064, pp. 133–136, 2008.

2 Similarity Neural Networks

A SNN consists in a feed–forward multilayer perceptron (MLP) trained to learn a similarity measure between couples of patterns $\boldsymbol{x}_i, \boldsymbol{x}_j \in I\!R^n$.

Humans are generally able to provide a supervision on the similarity of object pairs in a dyadic form (similar/dissimilar), instead of to associate intermediate degrees of similarity, that cannot be easily defined in a coherent way. Hence, SNNs are trained using dyadic supervisions. In detail, a training set $T = \{(\boldsymbol{x}_i, \boldsymbol{x}_j, t_{i,j});\ i = 1, ..., m;\ j = 1, ..., m\}$, collects a set of triples $(\boldsymbol{x}_i, \boldsymbol{x}_j, t_{i,j})$, being $t_{i,j}$ the similar/dissimilar label of $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in I\!R^{2n}$, which represents the input vector to the SNN.

The SNN model is composed by a single hidden layer with an even number of units and by a unique output neuron with sigmoidal activation function, that encloses the output range in the interval $[0, 1]$. The hidden neurons are fully connected both with the inputs and the output. The training is performed using Backpropagation for the minimization of the squared error function. Learning a similarity measure is a regression task but, due to the dyadic supervision, it could also be considered as a two-class classification task by applying a threshold to the output of the network.

If $sim() : I\!R^{2n} \rightarrow [0, 1]$ is the function computed by a trained SNN, then the following properties hold for any pair $\boldsymbol{x}_i, \boldsymbol{x}_j$: $sim(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$, and $sim(\boldsymbol{x}_i, \boldsymbol{x}_j) = sim(\boldsymbol{x}_j, \boldsymbol{x}_i)$. The first property is guaranteed by the sigmoidal activation function of the output unit. The second one is forced by exploiting weight sharing along the structure of the network. In Fig. 1(a) the shared weights can be observed, while Fig. 1(b) shows the unfolding of the network over the shared weights. The SNN is essentially composed by a "duplicated" input layer, formed by the original and the exchanged pair, and by two networks that share the corresponding weights (see Fig. 1(b)).

The learnt function is a similarity measure but not necessarily a metric, since $sim(\boldsymbol{x}_i, \boldsymbol{x}_i) = 0$ and the triangle inequality are not guaranteed. Those properties could be learnt from data, but they are not forced in any way by the structure of the network. SNNs are an instance of the neural networks proposed in [8] to process Directed Acyclic Graphs, hence it can be shown that they are universal approximators.

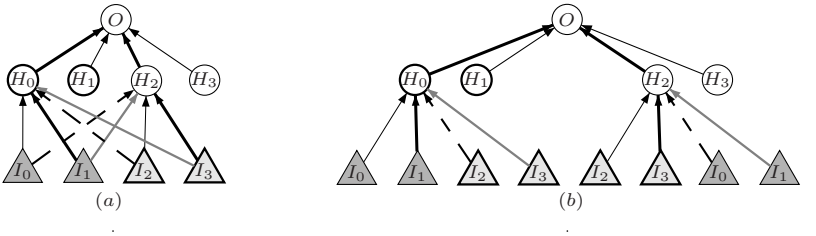

Fig. 1. The SNN architecture. Shared weights between two layers of neurons are drawn with the same gray level and mark. For the sake of simplicity, only some connections are depicted. (a) SNN for pairs of bidimensional vectors, $\boldsymbol{x}_i = [I_0, I_1]', \boldsymbol{x}_j = [I_2, I_3]'$. (b) SNN unfolded over the shared weights.

3 Experimental Results

SNNs were trained over some datasets from the UCI repository, whose patterns are divided in distinct classes. In order to approach the described similarity measure learning problem, pairs of patterns that belong to the same class were considered similar, otherwise they were labeled dissimilar. The training set for SNNs contains both similarity and dissimilarity pairs, collected in the sets S and D respectively. Such sets have been iteratively created by adding a new randomly selected similarity or dissimilarity pair until a target number of connected components of $G_S = (Dataset, S)$ and $G_D = (Dataset, D)$ (the *similarity* and *dissimilarity graphs*) has been obtained, following the sampling criterion of similarity pairs defined in [2].

Moreover, the S set has been enriched by applying the transitive closure over the similarity graph, whereas other dissimilarity pairs were added to D exploiting both the similarity and dissimilarity relationships (if a is similar to b and b is dissimilar to c, then a is dissimilar to c), as suggested in [4]. The training set generation allows also to define the test set; as a matter of fact, given a training set T, the test set collects all the pattern pairs except those belonging to T.

The accuracies of many SNNs were evaluated varying both the network architecture (the number of hidden units) and the amount of supervision; the obtained results are reported in Table 1. Accuracy has been computed by rejecting all outputs $o_{i,j}$ such that $|o_{i,j} - t_{i,j}| > \epsilon$, where $t_{i,j}$ is the target label, in order to evaluate the capability of the network to correctly separate examples belonging to different classes. Constraints size is expressed by $Kc * |Dataset|$, that represents the number of connected components of the similarity and dissimilarity graphs generated with the given supervisions.

The quality of the learned similarity measure has been compared against common distance functions (Euclidean and Mahalanobis distances), over the *cumulative neighbor purity* index. Cumulative neighbor purity measures the percentage of correct neighbors up to the K-th neighbor, averaged over all the data points.

Table 1. SNNs accuracy on 3 UCI datasets. Results are averaged over 20 random generations of constraints for each Kc. Best results for each architecture/supervision are reported in bold.

Kc	ϵ	**Dataset**											
		Iris				Balance				Wine			
		Hidden											
		4	6	10	16	10	12	16	30	18	22	28	36
0.9	0.3	**90.9**	90.7	90.4	89.1	81.1	**81.9**	80.4	79.8	78.6	79.6	79.5	**80.1**
	0.2	**90.3**	90.1	89.6	88.2	80.4	**81.1**	79.3	78.2	75.9	**76.9**	76.5	76.9
	0.1	**89.7**	89.1	88.5	86.9	79.3	**79.9**	77.4	75.7	71.7	**72.4**	71.6	71.6
0.8	0.3	92.3	**93.1**	92.3	92.4	85.4	84.7	84.9	**85.0**	**90.4**	89.7	90.4	90.2
	0.2	92.1	**92.7**	91.9	91.9	**84.6**	84.0	84.3	84.4	**88.8**	87.9	88.5	88.4
	0.1	91.6	**92.1**	91.2	91.3	83.3	82.9	83.4	**83.5**	**86.0**	84.9	85.2	84.8
0.7	0.3	93.4	**93.4**	93.4	92.9	87.1	86.8	86.7	**87.9**	94.9	**95.1**	95.0	95.1
	0.2	93.1	**93.2**	93.1	92.5	85.9	85.9	85.9	**87.4**	94.1	**94.2**	94.1	94.1
	0.1	92.7	92.8	**92.8**	91.9	84.1	84.3	84.5	**86.5**	**92.6**	92.5	92.4	92.5

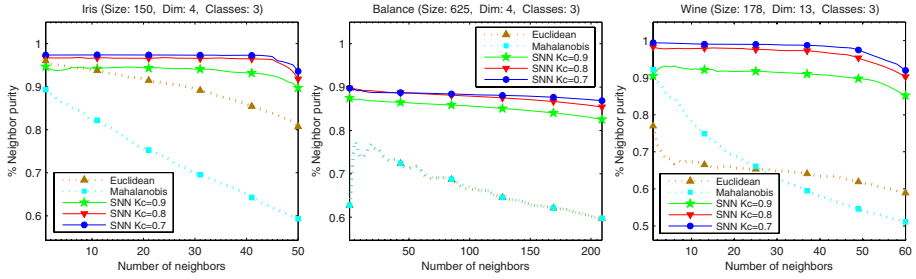

Fig. 2. Cumulative neighbor purity calculated over 3 datasets from the UCI repository. For each dataset, three results obtained by the SNNs trained with three differently sized sets of constraints, are shown. Each result is averaged over 10 random realization of constraints.

The maximum number of neighbors has been chosen such that $K \approx \frac{|Dataset|}{3}$. Results are reported in Fig. 2, showing that SNNs outperform common distance functions even if a small supervision is used.

4 Conclusions and Future Work

In this paper a neural network approach to similarity learning has been presented, showing encouraging results compared to common distance functions even with a small supervision. The proposed architecture assures to learn symmetric and non negative similarity relationship, and can also be trained to incorporate other properties of the data. Future work includes the application of the proposed similarity measure to clustering tasks.

References

1. Tversky, A.: Features of Similarity. Psychological Review 84(4), 327–352 (1977)
2. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning, with application to clustering with side-information. Advances in Neural Information Processing Systems 15, 505–512 (2003)
3. De Bie, T., Momma, M., Cristianini, N.: Efficiently learning the metric using side-information. In: Proc. Int. Conf. on Algorithmic Learning Theory, pp. 175–189 (2003)
4. Bilenko, M., Basu, S., Mooney, R.: Integrating constraints and metric learning in semi-supervised clustering. In: Proc. Int. Conf. on Machine Learning, pp. 81–88 (2004)
5. Basu, S., Bilenko, M., Mooney, R.: A probabilistic framework for semi-supervised clustering. In: Proc. Int. Conf. on Knowledge Discovery and Data Mining, pp. 59–68 (2004)
6. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a Mahalanobis Metric from Equivalence Constraints. J. Machine Learning Research 6, 937–965 (2005)
7. Tsang, I., Kwok, J.: Distance metric learning with kernels. In: Proc. Int. Conf. on Artificial Neural Networks, pp. 126–129 (2003)
8. Bianchini, M., Gori, M., Scarselli, F.: Processing directed acyclic graphs with recursive neural networks. IEEE Trans. on Neural Networks 12(6), 1464–1470 (2001)