Polar Sine Based Siamese Neural Network for Gesture Recognition

Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, Christophe Garcia

Polar Sine based Siamese Neural Network for Gesture Recognition

Samuel Berlemont[1], Grégoire Lefebvre[1],
Stefan Duffner[2], and Christophe Garcia[2]

[1]Orange Labs, R&D, Grenoble, France
{firstname.surname}@orange.com
[2]LIRIS, UMR 5205 CNRS, INSA-Lyon, F-69621, France.
{firstname.surname}@liris.cnrs.fr

Abstract. Our work focuses on metric learning between gesture sample signatures using Siamese Neural Networks (SNN), which aims at modeling semantic relations between classes to extract discriminative features. Our contribution is the notion of polar sine which enables a redefinition of the angular problem. Our final proposal improves inertial gesture classification in two challenging test scenarios, with respective average classification rates of 0.934 ± 0.011 and 0.776 ± 0.025.

Keywords: Siamese Neural Network, Metric Learning, Polar Sine, Gesture Recognition

1 Introduction

As consumer devices become more and more ubiquitous, new interaction solutions are required. In recent years, new sensors called MicroElectroMechanical Systems (MEM) were popularized thanks to their small sizes and low production costs. Two kinds of gestures can be considered for different applications. On the one hand, static gestures correspond to a specific state, described by a unique set of features, with, in the context of Smartphones, a "phone-to-ear" posture for instance. On the other hand, dynamic gestures are more complex, since they are described by a time-series of inertial signals, such as the "picking-up" movement when the user is ready to take a call. Thus, in this study, we explore inertial-based gesture recognition on Smartphones, where gestures holding a semantic value are drawn in the air with the device in hand.

Based on accelerometer and gyrometer data, three main approaches exist. The earliest methods suggest to model the temporal structure of a gesture class, with Hidden Markov Models (HMM) [10]; while another approach consists in matching gestures with reference instances, using a non-linear distance measure generally based on Dynamic Time Warping (DTW) [1]. Finally, features can be extracted from gesture signals in order to train specific classifiers, such as Support Vector Machines (SVM) [11].

Our work focuses thus on metric learning between gesture sample signatures using Siamese Neural Networks (SNN) [3], which aims at modeling semantic relations between classes to extract discriminative features, applied to the Single

Feed Forward Neural Network (SFNN). Contrary to some popular versions of this algorithm, we opt for a strategy that does not require additional class-separating-parameter fine tuning during training. After a preprocessing step where the data is filtered and normalized spatially and temporally, the SNN is trained from sets of samples, composed of similar and dissimilar examples, to compute a higher-level representation of the gesture, where features are collinear for similar gestures, and orthogonal for dissimilar ones. As opposed to the classical input set selection strategies, using similar or dissimilar pairs, or {reference, similar, dissimilar} triplets, we propose to include samples from every available dissimilar classes, resulting in a better structuring of the output space. Moreover, the notion of polar sine enables a redefinition of the angular problem by maximizing a normalized volume induced by the outputs of the reference and dissimilar samples, which results in a non-linear discriminant analysis similar to independant component analysis.

This paper is organized as follows. Section 2 presents related works on SNN. In Section 3, we explain our contributions with a new SNN objective function. Then, Section 4 describes our results for gesture recognition. Finally, our conclusions and perspectives are drawn.

2 Related Studies on SNN

2.1 Training Set Selection

A SNN is trained to project multiple samples coherently. Two identical neural networks with shared weights W take simultaneously two input samples $\mathbf{X_1}$ and $\mathbf{X_2}$ to compute the error relative to a cosine-based objective function, thanks to the respective outputs $\mathbf{O_{X_1}}$ and $\mathbf{O_{X_2}}$ (see Figure 1a). The resulting application of the network depends on the kind of knowledge about similarities one expects. In problems such as face or signature verification [3,4,8,2], the similarity between samples depends on their origin, and the network allows to determine the genuineness of a test sample with a binary classification. In cases involving the learning of a mapping that is robust to specific transformations [6], similar samples differ by slight rotations or translations. However, similarities can be more abstract concepts, such as same documents in different languages [13]. The most common representation consists in a binary relation based on pairs: given two samples $\mathbf{X_1}$ and $\mathbf{X_2}$, the $(\mathbf{X_1}, \mathbf{X_2})$ pair similarity is determined by a tag, which takes two different values whether the relation is similar or dissimilar. However, knowledge about semantic similarities can take more complex forms. Lefebvre *et al.* [8] expand the information about expected neighborhoods with triplets $(\mathbf{R}, \mathbf{P}, \mathbf{N})$, composed of a reference sample $\mathbf{R}$ for each known relation, with $\mathbf{P}$ a *positive* sample forming a genuine pair with $\mathbf{R}$, while $\mathbf{N}$, the *negative* sample, is the member of an impostor pair. Similarities are then represented as much as dissimilarities. With these different knowledge representations presenting multiple samples to a set of weight-sharing sub-networks, it is necessary to study new objective functions in order to define how semantic relations will be reflected in the output space.

2.2 Objective Function

The contrastive loss layer objective function aims at computing a similarity metric between the higher-level features extracted from multiple input patterns. Thus, this discriminative distance is trained to get smaller for similar patterns, and higher for dissimilar ones. It takes two forms, respectively bringing together and pushing away features from similar and dissimilar pair of patterns. Given two samples $\mathbf{X_1}$ and $\mathbf{X_2}$, two main similarity measures are used: the cosine similarity, based on the cosine value between these two samples $\cos(\mathbf{X_1}, \mathbf{X_2}) = \frac{\mathbf{X_1}.\mathbf{X_2}}{\|\mathbf{X_1}\|.\|\mathbf{X_2}\|}$; and the Euclidean similarity $d(\mathbf{X_1}, \mathbf{X_2}) = \|\mathbf{O_{X_1}} - \mathbf{O_{X_2}}\|_2$. In this study, we focus on cosine-based objective functions. A cosine objective function aims at learning a non-linear cosine similarity metric, whether it is expressed specifically, in the form of multiple targets, or relatively, by pair scores ranking. The cosine similarity metric is defined as:

$$cos_{sim}(\mathbf{X_1}, \mathbf{X_2}) = 1 - \cos(\mathbf{X_1}, \mathbf{X_2}) \tag{1}$$

Square Error Objective One approach comes from the original use of the square error objective function for the SFNN. Given a network with weights W and two samples $\mathbf{X_1}$ and $\mathbf{X_2}$, a target $t_{\mathbf{X_1X_2}}$ is defined for the cosine value between the two respective output vectors $\mathbf{O_{X_1}}$ and $\mathbf{O_{X_2}}$. In [3], Bromley *et al.* set this target to 1 if for a similar pair, and -1 otherwise. Given the similarity label Y and the weights W of the network, the error E_W for any pair defines:

$$E_W(X_1, X_2, Y) = (t_{\mathbf{X_1X_2}}(Y) - \cos(\mathbf{O_{X_1}}, \mathbf{O_{X_2}}))^2 \tag{2}$$

Triangular Similarity Metric Zheng *et al.* [14] imply these same targets. Given Y the numerical label for the $(\mathbf{X_1}, \mathbf{X_2})$ pair, acting as the target $t_{\mathbf{O_{X_1}O_{X_2}}}$ and respectively equal to 1 and -1 for similar and dissimilar pairs; the triangular inequality imposes:

$$\|\mathbf{O_{X_1}}\| + \|\mathbf{O_{X_2}}\| - \|\mathbf{C}\| \geq 0, \quad with \quad \mathbf{C}(\mathbf{X_1}, \mathbf{X_2}, \mathbf{Y}) = \mathbf{O_{X_1}} + Y.\mathbf{O_{X_2}} \tag{3}$$

After adding norm constraints to prevent a degeneration towards a null projection, the final objective function becomes:

$$\begin{aligned} E_W(X_1, X_2, Y) = \|\mathbf{O_{X_1}}\| + \|\mathbf{O_{X_2}}\| - \|\mathbf{C}(\mathbf{X_1}, \mathbf{X_2}, \mathbf{Y})\| + 0.5(1 - \|\mathbf{X_1}\|)^2 \\ +0.5(1 - \|\mathbf{X_2}\|)^2 = 0.5\,\|\mathbf{O_{X_1}}\|^2 + 0.5\,\|\mathbf{O_{X_2}}\|^2 - \|\mathbf{C}(\mathbf{X_1}, \mathbf{X_2}, \mathbf{Y})\| + 1 \end{aligned} \tag{4}$$

Deviance Cost Function Inspired by the common loss functions such as square or exponential losses, Yi *et al.* [12] opt for the binomial deviance. Since their Siamese architecture does not necessarily share weights between subnetworks, let B_1 and B_2 be the respective functions associated to both subnetworks, and $B_1(\mathbf{X_1})$ and $B_2(\mathbf{X_2})$ be the projections of the samples of a pair, we get:

$$E_W(X_1, X_2, Y) = \ln(\exp^{-2Y.cos(B_1(\mathbf{X_1}), B_2(\mathbf{X_2}))} + 1) \tag{5}$$

Triplet Similarity Objective Lefebvre *et al.* [8] generalize the Square Error Objective by using simultaneously targets for genuine and impostor pairs. Samples outputs from similar classes are collinear while outputs from different classes tend to be orthogonal, which translates as a target equal to 1 for similar pairs and 0 for dissimilar ones. Let $(\mathbf{R}, \mathbf{P}, \mathbf{N})$ be a triplet, with a reference sample $\mathbf{R}$, a positive sample $\mathbf{P}$ forming a similar pair with $\mathbf{R}$, and a negative sample $\mathbf{N}$, forming a dissimilar pair with $\mathbf{R}$, we get:

$$E_W(\mathbf{R}, \mathbf{P}, \mathbf{N}) = (1 - cos(\mathbf{O_R}, \mathbf{O_P}))^2 + (0 - cos(\mathbf{O_R}, \mathbf{O_N}))^2. \tag{6}$$

3 Our contributions - SNN-psine

3.1 Training Set Selection Strategy

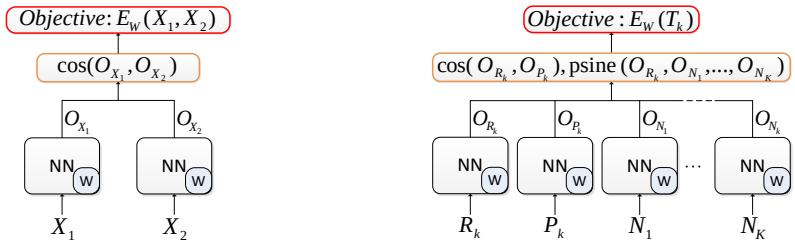
Every training set selection strategy for a Siamese network consists in defining a certain number of similar and dissimilar pairs, deemed representative of the global relationships within the data. This generally induces a bias, since it is not possible to ensure a perfect coverage for every relationship. For this reason, we first propose a unified approach for multi-class problems. Let $C = \{C_1, .., C_K\}$ be the set of classes represented in the training data, $\mathbf{O_{R_k}}$ the output vector of the reference sample $\mathbf{R_k}$ from the class C_k presented to the model for update, $\mathbf{O_{P_k}}$ the output of a different sample $\mathbf{P_k}$ from the same class, and $\mathbf{O_{N_l}}$ the output of a sample $\mathbf{N_l}$ from another class C_l. In order to keep symmetric roles for every class and optimize the efficiency of every update, we propose here to minimize an error criterion for training tuples $T_k = \{\mathbf{R_k}, \mathbf{P_k}, \{\mathbf{N_l}, l = 1..K, l \neq k\}\}$ involving one reference sample from the class C_k, one positive sample and one negative sample from every other class. This leads us to the definition of the SNN-cos, relying on the following cost function. The total error estimation for a training set T_k, $E_W(T_k)$, becomes:

$$E_W(T_k) = (1 - \cos(\mathbf{O_{R_k}}, \mathbf{O_{P_k}}))^2 + \sum_{l=1, l\neq k}^{K} (0 - cos(O_{R_k}, O_{N_l}))^2. \tag{7}$$

3.2 Objective Function Reformulation

While the cosine allows for a correlation estimation between two vectors in any Euclidean space of finite dimension, it is sensible to consider another function which would measure dissimilarities, like the sine in 2D. In the following, we propose a reformulation of the objective function based on a higher-dimensional dissimilarity measure, the polar sine. Lerman *et al.* [9] define the *polar sine (PolarSine)* for a set $V = \{v_1, \ldots, v_n\}$ of m-dimensional $(m > n)$ linearly independent vectors, forming the columns of the matrix $\mathbf{A} = \begin{bmatrix} \mathbf{v}_1 \ \mathbf{v}_2 \cdots \mathbf{v}_n \end{bmatrix}$ and its transpose $A^\top$:

$$PolarSine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \frac{\sqrt{\det(A^T.A)}}{\prod_{i=1}^{n} \|\mathbf{v_i}\|} \tag{8}$$



(a) Original SNN architecture. (b) Our SNN proposal.

Fig. 1: Comparison between the original and the proposed architectures. The original SNN processes pair similarity with two weight-sharing NNs and a cosine based objective, while our proposal handles comprehensive class relationships with a combination of cosine and psine metrics.

As a measure of a regularized hyper-volume, the polar sine acts as another similarity metric, more precisely as a dissimilarity metric. However, in order to prevent numerical instabilities during the training process and make the metric value independent from the size of the set of vectors, we propose a redefinition of the Polar Sine for learning angles. In the following, we call this adaptation the *Polar Sine Metric (psine)*. Given $\mathbf{A_{norm}} = \left[\frac{\mathbf{v_1}}{\|\mathbf{v_1}\|} \frac{\mathbf{v_2}}{\|\mathbf{v_2}\|} \cdots \frac{\mathbf{v}_n}{\|\mathbf{v_n}\|}\right]$ and $\mathbf{S} = \mathbf{A_{norm}}^\top.\mathbf{A_{norm}}$, i.e. $\mathbf{S}(i,j) = \cos(\mathbf{v_i}, \mathbf{v_j})$, the polar sine metric equals to:

$$psine(\mathbf{A}) = \sqrt[n]{\det(\mathbf{S})}. \tag{9}$$

Thus, optimizing the polar sine metric corresponds to assigning a target equal to 0 to the cosine between every available pair of different vectors drawn in $T_k \setminus \{O_{P_k}\}$. This comprehensive representation actually holds more information than our original objective function which aimed at assigning zero-cosine-values only for pairs between the reference and negative outputs. Furthermore, this approach is easily scalable to any number of classes. With two comparable similarity estimators, whose values are comprised between 0 and 1, it is now possible to redefine the objective function for our training sets T_k (see Figure 1b):

$$E_W(T_k) = (1 - \cos(\mathbf{O_{R_k}}, \mathbf{O_{P_k}}))^2 + (1 - psine(\mathbf{O_{R_k}}, \mathbf{O_{N_1}}, \ldots, \mathbf{O_{N_K}}))^2. \tag{10}$$

4 Experiments

4.1 Database

Using the same data and process as [7] and [5], both proving that neural approaches are suited to the gesture recognition problematic, two datasets were formed, based on the accelerometer and gyrometer data from the Android Samsung Nexus S device, sampled at 40Hz. The first dataset, named DB1, contains

40 repetitions of 14 different classes performed by a single individual, for a total of 720 records. Conversely, DB2 contains 5 repetitions of these 14 gesture classes performed by 22 individuals, for a total of 1540 records. DB2 corresponds to an open world testing with multiple users. The 14 classes in DB2 encompass gestures with different complexities. They are composed of linear gestures, with horizontal (flick North, South, East, West) and vertical (flick Up, Down) translations; curvilinear gestures (clockwise and counter-clockwise circles, alpha, heart, N and Z letters, a pick gesture towards, and a throw gesture away from the user).

4.2 Protocols

The classification results rely on 4 protocols, named C1 to C4, covering different real application settings: C1, based on DB1, covers the closed-world application with a single user in a context of a personalization paradigm, with 5 randomly selected samples per class for training, and 16 samples for testing; C2, based on DB2, corresponds to a multi-user, closed-world application. Every user is represented in the training data, with 2 samples per class and per user used for training, and the 3 remaining samples for testing; C3, based on DB2, consists in open-world problem, where a comprehensive user representation is not possible: training is performed on every sample from 17 users, while testing is carried out on the samples of the 5 remaining users; C4, based on DB2, is the most challenging scenario, testing the generalization capabilities of each model, with one user used as a training reference and the samples from the 21 remaining users used for tests. Each protocol is repeated 10 times so as to minimize the influence of the training and testing data selection.

The performance of our SNN-psine is compared to the following methods : our SFNN classifies the 270-feature vectors from 45-neuron hidden layer with a hyperbolic tangent activation function, and a 14-neuron "softmax" output layer; our SNN-cos and SNN-psine share the same architecture, and classify with a KNN (K=1) the outputs of a SNN from 270-feature vectors, with 45-neuron hidden layer with a hyperbolic tangent activation function, and a 80-neuron "linear" output.

4.3 Results

Protocol C1 : The general performance comparisons between the main models for gesture recognition are presented in Table 1. Every version of the SNN show a comparable result (i.e. 98.8% for SNN-cos and 98.7% for SNN-psine). These are the highest scores for neural-based methods, which proves the coherence of the learnt projections. Indeed, both SNN results overcome the SFNN average classification rate of 97.8%.

Protocol C2: The SNN-cos shows the best accuracy for protocol C2 of 96.9%, closely followed by SNN-psine with 96.8%, proving that the SNN performs well even when multiple, different gesture dynamics are involved. Once again, the SFNN obtains a lower score of 94.5%. A closer study of one confusion matrix for the SNN-psine shows small confusions between "N" and "Up", and "Alpha"

and "Heart", which are indeed similar gestures. Moreover, an understandable confusion between the vertical, upwards, gestures "Up" and "Pick" appears. An analysis of the source of these errors shows that all of these samples belong to a unique user. Thus, this phenomenon underlines the fact that some users may have a really specific way of performing gestures, which, combined with the imprecision of the sensors, may result in a great difficulty to manage them with a single, general model not specifically trained for these singletons.

Protocol C3: This protocol amplifies the difficulties encountered with C2. The SNN-psine and SNN-cos take advantage of the bigger training dataset with an accuracy of 93.4%. Once again, the SFNN performance is lower, with 90.5%. In that case, the SNN-psine shows a high symmetric confusion between "Pick" and "Up". It also handles badly the gesture "Throw". Indeed, this gesture, which consists in an arc away from the user, brought about fears of actually throwing the device, resulting in the highest disparities between users.

Protocol C4: Finally, this protocol presents the highest challenge for these methods, with a single user data for training. As a consequence, the SNN-psine and SNN-cos overtake the SFNN, with respective accuracies of 77.6% and 77.5% against 74.4%. The flaws identified above are amplified. The "Alpha" and "Clockwise" gestures are still confused. Moreover, the "Throw" gesture still shows the highest variability among users, representing 25% of the total number of errors, with heavy confusions with the "Tap" and "FlickN" gestures.

Consequently, our SNN-psine contribution is a very challenging solution on the 4 protocols, and even better for C3 and C4 protocols. Nevertheless, some limitations are identified, with confusions between gestures where one can be identified as a part of the other. Moreover, the complexity for the SNN-psine error computation, compared to the complexity for the SNN-cos in Table 2[1], implies a trade-off between class relationships which has to be taken into account. However, parallelizable matrix computations allow for a limited repercussion on training times for SNN-psine, with an effective 23% update time increase for the protocol C4 compared to the SNN-cos.

5 Conclusion and Perspectives

In this study, we first propose an adaptation of the Siamese strategy to a multi-class classification context for a stochastic training. We propose a unified similarity function, the Polar Sine Metric, which offers a comprehensive representation of dissimilarity relationships within the training set. The Polar Sine Metric proposes a matrix approach to describe relationships, and relies on a determinant to compute the final dissimilarity for a set of samples. The complexity evaluation implies $0.5N_c(N_c - 1) + 1$ relationships in the cost function per update given a reference sample, with N_c the number of classes. Thus, the training set sizes should be taken into account for future research, so as to study the trade-off between accuracy and complexity when the number of classes increases.

[1] Computations are performed on an Intel© Core™i7-4800MQ processor at 2.70GHz.

Table 1: Recognition rates on our 4 protocols.

	C1	C2	C3	C4
SFNN	0.978 ± 0.010	0.954 ± 0.006	0.905 ± 0.010	0.744 ± 0.040
SNN-cos	$\mathbf{0.988} \pm 0.005$	$\mathbf{0.969} \pm 0.007$	0.934 ± 0.013	0.775 ± 0.032
SNN-psine	0.987 ± 0.011	0.968 ± 0.006	$\mathbf{0.934} \pm 0.011$	$\mathbf{0.776} \pm 0.025$

Table 2: Complexities and times for one update (in ms) on protocol C4.

	Complexity	Number of relationships	Training time for C4 ($N_c = 14$)
cos	$\mathcal{O}(N_c)$	N_c	$2.61779 \pm 1.03648.10^{-1}$
psine	$\mathcal{O}(N_c^{\log_2 7})$	$N_c(N_c - 1)/2 + 1$	$3.21632 \pm 1.79093.10^{-1}$

References

1. A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In *ICASSP*, 2010.
2. S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia. Siamese neural network based similarity metric for inertial gesture classification and rejection. In *AFGR*, 2015.
3. J. Bromley, I. Guyon, Y. Lecun, E. Sackinger, and R. Shah. Signature Verification using a "Siamese" Time Delay Neural Network. In *NIPS*, 1994.
4. Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pages 539–546. IEEE, 2005.
5. S. Duffner, S. Berlemont, G. Lefebvre, and C. Garcia. 3d gesture classification with convolutional neural networks. In *ICASSP*, pages 5432–5436. IEEE, 2014.
6. R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.
7. G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia. BLSTM-RNN based 3d gesture classification. In *ICANN*, pages 381–388. Springer Berlin Heidelberg, 2013.
8. G. Lefebvre and C. Garcia. Learning a bag of features based nonlinear metric for facial similarity. In *AVSS*, pages 238–243. IEEE, 2013.
9. G. Lerman and J. T. Whitehouse. On d-dimensional d-semimetrics and simplex-type inequalities for high-dimensional sine functions. *J. Approx. Theory*, 156(1):52–81, January 2009.
10. T. Pylvänäinen. Accelerometer Based Gesture Recognition Using Continuous HMMs Pattern Recognition and Image Analysis. volume 3522 of *Lecture Notes in Computer Science*, chapter 77, pages 413–430. Berlin, Heidelberg, 2005.
11. J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li. Gesture recognition with a 3-d accelerometer. volume 5585 of *LNCS*, pages 25–38, 2009.
12. D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *ICPR*, pages 34–39. IEEE, 2014.
13. W.-T. Yih, K. Toutanova, J. C. Platt, and C. Meek. Learning discriminative projections for text similarity measures. In *CoNLL*, pages 247–256. Association for Computational Linguistics, 2011.
14. L. Zheng, K. Idrissi, C. Garcia, S. Duffner, and A. Baskurt. Triangular similarity metric learning for face verification. In *AFGR*, 2015.