Sequence Metric Learning as Synchronization of Recurrent Neural Networks

Paul Compagnon*†, Grégoire Lefebvre *, Stefan Duffner† and Christophe Garcia†
*Orange Labs, Grenoble, France
Email: name.surname@orange.com
†Université de Lyon, INSA Lyon, LIRIS UMR 5205 CNRS, Lyon, France
Email: name.surname@liris.cnrs.fr

***Abstract*—Sequence metric learning is becoming a widely adopted approach for various applications dealing with sequential multi-variate data such as activity recognition or natural language processing. It is most of the time tackled with sequence alignment approaches or representation learning. In this paper, we propose to study this subject from the point of view of dynamical system theory by drawing the analogy between synchronized trajectories produced by dynamical systems and the distance between similar sequences processed by a siamese recurrent neural network. Indeed, a siamese recurrent network comprises two identical sub-networks, two identical dynamical systems which can theoretically achieve complete synchronization if a coupling is introduced between them. We therefore propose a new neural network model that implements this coupling with a new gate integrated into the classical Gated Recurrent Unit architecture. This model is thus able to simultaneously learn a similarity metric and the synchronization of unaligned multi-variate sequences in a weakly supervised way. Our experiments show that introducing such a coupling improves the performance of the siamese Gated Recurrent Unit architecture on two datasets: one dedicated to activity recognition and another to transportation recognition.**

***Index Terms*—Sequence Metric Learning, Recurrent Neural Networks, Dynamical Systems, Synchronization**

I. Introduction

Metric learning aims at learning an essential component for numerous machine learning algorithms used for classification or clustering: a similarity. It has the benefit to be usable in weakly supervised settings where only equivalence constraints between samples are known [1], which allows for a large number of applications on various data types: from person re-identification [2], object tracking [3] and gesture recognition [4] to sentence similarity computation [5]. Among those applications, less attention has been given to design specific sequence metric learning algorithms, specifically with neural networks despite the simplicity of the siamese architecture [6].

One easy way to adapt existing approaches to sequential data is to learn representations through Sequence-to-Sequence models [7] or Transformers [8]. However, these models would be difficult to learn in a weakly supervised way for providing a similarity metric and further lose temporal dependency information inside the sequence and alignment information between sequences. On the contrary, Dynamic Time Warping (DTW) [9] is a classical approach to measure distance between sequences and relies on aligning sequences. Its integration inside learning algorithms has been rendered difficult by its non-differentiability and its theoretical quadratic time complexity which badly suits the equivalence constraint framework and some associated more complex losses [2], [10], [11]. Recent works mitigate these drawbacks notably with virtual metric learning [12], [13] and soft versions of DTW [14], [15].

Therefore, we aim at designing a neural network architecture specifically adapted to sequence metric learning. Recurrent neural networks (RNN) have a temporal dynamic behavior which allows to study them as dynamical systems. We propose in this paper a new framework for sequence metric learning based on dynamical system synchronization theory. We propose to replace the concept of metric in a vector space by the concept of synchronization of trajectories in a state space. Instead of computing distances on input representations, we propose to measure how two dynamical systems, and precisely two RNNs, respond to input pairs in term of synchronization. The notion of coupling is crucial when trying to synchronize dynamical systems. We introduce a coupled version of the Gated Recurrent Unit (GRU) [16] to implement coupling inside a siamese architecture. Our experimental evaluation shows that this modification provides an improvement over a classical siamese GRU implementation.

The paper is organized as follows: Section II outlines the state-of-the-art approaches in sequence metric learning, Section III describes our framework and our new siamese architecture, Section IV shows our experimental results to assess the performances of our approach on two datasets, and Section V presents our conclusions and perspectives.

II. Related Work

a) Recurrent neural networks and dynamical system theory.: A main property of RNN is to exhibit a dynamic behavior which enables them to learn temporal sequence correlations. This behavior can therefore be studied using dynamical system theory: an important result being that RNN can approximate any finite-time trajectory of a dynamical system [17]. Other early works analyzed the RNN convergence stability [18] and helped to understand the problem of long term dependencies [19]. Laurent et al. [20] studied the dynamics of Long-Short Term Memory (LSTM) Neural Networks and GRU and observed that it is chaotic in the absence of input data. They designed a Chaos-Free RNN architecture having a more predictable behavior. In

another recent publication, Chang et al. [21] studied RNN trainability and established a connection with discretized Ordinary Differential Equations stability. They identified a criterion to guarantee that the system can preserve long-term dependencies and proposed a new version of RNN based on those observations. Both papers demonstrate that dynamical system theory is a fertile soil to study and conceive new RNN models. Finally, we would like to mention works on the definition of metrics to compare non-linear dynamical systems [22], [23] although our objective is not exactly the same, as we propose to use dynamical system synchronization theory to improve metric learning on any type of sequential data, whereas these methods have been conceived to work more specifically with structural data.

b) Sequence metric learning.: DTW is a classical approach for measuring distances between sequences [9]. Numerous improvements have been brought to the original formulation notably to improve the k-nearest neighbor performance [24]. Abid et al. [15] proposed to learn the DTW parameters that allow to reproduce the Euclidean distances between sequence representations learned with a Sequence-to-Sequence model [7]. In contrast, Su et al. [25] proposed an alternative to DTW, the Order-Preserving Wasserstein (OPW) distance, by viewing the problem of metric learning between sequences as an optimal transport problem regularized to preserve the temporal relationships between the samples, and they solved it with the matrix scaling algorithm. In a following paper [13], Su et al. reformulated the DTW and OPW distances as parameterized meta-metrics of a single ground metric and proposed an optimization process to learn the metric and the latent alignment with virtual metric learning [12], which reduces the number of constraints. Not only this approach speeds up training but it also outperforms several other metric learning approaches, notably approaches conceived for points generalized to sequences. In comparison, we propose a pure RNN approach similar to Müller et al. [5] who presented a siamese neural network approach to learn sentence similarities as a l_1-norm. In their method, the LSTM network combines the embeddings of the words of the sentence to learn a distance between representations of sentences. Finally, Varior et al. [26] proposed a siamese convolutional architecture for person re-identification from video data with gates linking parallel layers allowing to accentuate common patterns between both representations. This leads to representations that are more suited to distinguish some pairs of similar or dissimilar images.

In this paper, we introduce an alternative to DTW: a pure neural network approach to sequence metric learning based on the siamese RNN architecture. We propose to enhance the classical siamese RNN by studying this model from dynamical system point of view, as it has been already done for standard RNN. The disadvantage of DTW-based approaches compared to ours is that it can be computationally inefficient and non-differentiable, which prevents their possible combination with other gradient-based models.

III. Synchronizing GRU Siamese Networks

In this section, we first draw a parallel between the concept of synchronization for dynamical systems and the task of sequence metric learning with siamese RNN. We then justify from a theoretical point of view the introduction of coupling inside the siamese architecture. We finally introduce our main contribution in Section III-C: a modified siamese GRU model implementing this coupling.

A. Synchronization of Chaos and Metric Learning

The concept of synchronization is generally well-understood for time-periodic dynamical systems: this phenomenon is called phase synchronization. However, it is less-known that it can also occur for chaotic dynamical systems [27], that is, systems for which resulting trajectories exponentially diverge for infinitesimally close initial conditions. This practically means that the behavior of such systems can become rapidly unpredictable solely due to small variations of the initial conditions. Common examples of such systems are the double pendulum or the n-body problem. To formalize the concept of synchronization for chaotic systems, Brown et al. [28] proposed a general definition of it:

Definition 1: Let Z be a dynamical system composed of two subsystems X and Y such that:

$$\begin{aligned} X : \frac{\mathrm{d}x(t)}{\mathrm{d}t} &= f_1(x(t), y(t); t) \\ Y : \frac{\mathrm{d}y(t)}{\mathrm{d}t} &= f_2(y(t), x(t); t), \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^{d_1}$ and $y(t) \in \mathbb{R}^{d_2}$ with $d_1, d_2 \in \mathbb{N}$. Let $\phi(z_0)$ be a trajectory of Z with initial conditions $z_0 = [x(0), y(0)] \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$. Finally let $g : X$ (resp. Y) $\times \mathbb{R} \to \mathbb{R}^k$ with $k \in \mathbb{N}$, be a measurable property of the subsystems. They are synchronized on the trajectory $\phi(z_0)$ with respect to the property g if there is a time independent function $h : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^k$ such that:

$$||h(g(x), g(y))|| = 0, \tag{2}$$

where $||\cdot||$ is a norm.

From this definition, it is possible to derive several ways to measure synchronization between two trajectories. Brown et al. [28] report several slightly different formulations of the synchronization error with the following being the most used. According to them, for identical systems:

$$h(g(x), g(y)) = \lim_{t \to +\infty} (g(x(t)) - g(y(t))), \tag{3}$$

where h and g are the same as in Definition 1.

We rewrite this synchronization error for discrete systems in a continuous form by replacing the limit by a comparison of the last element of each trajectory x and y of length T:

$$h(g(x), g(y)) = x(T) - y(T)\ , \tag{4}$$

with g being here a function returning the coordinates of the points. We define d as a distance on discrete dynamical system trajectories derived from the synchronization error by

replacing the difference with the Euclidean norm (in definition 1, any norm can be used) to get only positive values:

$$d(x, y) = ||x(T) - y(T)||_2. \tag{5}$$

However, RNNs are dynamical systems, and the output sequences are trajectories. Thus, learning a Euclidean distance with a siamese RNN is equivalent to trying to synchronize the two output sequences of a siamese network for similar pairs.

While being intuitive and suitable for metric learning, the metric of Equation 5 measures synchronization only at one point in time, which forces the system to achieve synchronization at this precise point. This is called dead-beat synchronization, synchronization in a finite number of steps [29]. Even if at first sight, this seems not really different from computing distance on input sequence representations, synchronization could actually be assessed at several samples of the sequence and even continuously. We will now specify what type of synchronization siamese RNNs are able to achieve and under which conditions.

B. Complete Synchronization of Coupled Identical Systems

A special case of Definition 1 is when f_1 and f_2 are the same function, the dynamical systems share the same parameters, and they are said to be *identical*. Analogously, the two (or more in case of triplet inputs) sub-networks of a siamese network also share the same parameters; only input sequences differ [6]. To simplify, we will first study the case where the dynamics of the RNNs are solely driven by its initial condition (the initial hidden state) and where no input sequence is given (i.e. a sequence of 0s). We obtain what is called the *dynamical system induced* by the RNN. In this case, only the initial conditions differ and the sub-networks are identical dynamical systems. According to experiments conducted by Laurent et al. [20], dynamical systems induced by RNN exhibit a chaotic behavior. Under what conditions identical systems can synchronize? Consider now the following two identical systems:

$$\begin{aligned} X : \frac{\mathrm{d}x(t)}{\mathrm{d}t} &= f(x; t) + C(y(t) - x(t))^T \\ Y : \frac{\mathrm{d}y(t)}{\mathrm{d}t} &= f(y(t); t) + C(x(t) - y(t))^T, \end{aligned} \tag{6}$$

where $x, y \in \mathbb{R}^n$ and C is a coupling matrix in $\mathbb{R}^{n \times n}$. This type of coupling is called *diffusive* because it will dissipate the dynamics of each sub-system [30]. X and Y are bidirectionally coupled systems[1]. Fujisaka et al. [31] showed that the system described by Equation 6 can achieve complete synchronization if C is a multiple of the Identity matrix and a constant c which verifies the following condition:

$$c > \frac{1}{2}\lambda_L, \tag{7}$$

where λ_L is the largest Liapunov exponent of the system. The Liapunov exponents quantify the sensibility of a system to initial conditions: if it has one positive Liapunov exponent, predictability of its behavior becomes impossible beyond a certain time horizon; it is therefore chaotic [32].

But the trajectories of RNNs are most of the time also influenced by external inputs: the input sequence. In this case, the dynamics of the RNNs are mostly driven by these external inputs [20] and RNN starting from different initial conditions but given identical input sequence will see their trajectories synchronize, i.e. the hidden states become the same after a few steps. Coupling is in this case not absolutely necessary to achieve synchronization: regarding metric learning, siamese LSTM actually works without coupling [5]: the model achieves low distances for similar inputs and therefore synchronization. However, coupling could allow to enforce lower distances with sequences that have similar dynamics but are composed of quite different data, i.e. so-called hard positive samples, and even to force the synchronization regardless of the input pair. Hard positive samples are samples that are factually belonging to one class, by the label, but lie close to the decision boundary or even beyond in terms of distance (conversely for hard negative samples). They are particularly studied by the metric learning community to improve the convergence and performances of metric learning models [33].

We showed in this section the motivation and aim of implementing coupling into the siamese RNN architecture. Indeed, the induced dynamics of GRU and LSTM are chaotic and, in this case, coupling allows their complete synchronization as the sub-networks of a siamese network share the same weights and represent thus identical dynamical systems. When given input sequences, the dynamics of GRU and LSTM are mostly driven by these external inputs. In this case, while not being critical to achieve synchronization (and therefore low distances between similar elements), coupling could facilitate bringing similar inputs closer, particularly for hard positive pairs. Moreover, complete synchronization is a special case of more general synchronization types such as the so-called *generalized* synchronization [34] for which other errors and metrics are associated (for example, mutual interdependence [35]). The same derivation could thus be made for these synchronization errors, leading to different metrics. We will in the experimental section use Equation 5 as our metric to learn, but this is here the most straightforward case of a framework from which more complex sequence metrics to be learned with siamese RNN models, can be obtained.

C. Coupled GRU

We present a new neural network model that directly implements coupling within a siamese RNN architecture. From a machine learning perspective, this coupling needs to be trainable such that the network *learns* to achieve synchronization for similar inputs and stay desynchronized for different ones. We propose to apply the coupling by means of two new gates inside the GRU architecture which we call in the following CGRU (Coupled Gated Recurrent Unit). We chose to use GRU and not LSTM [36] because the operation of a

[1]Coupling can also be directional, only one system influences the other: they are called in this case drive-response systems.

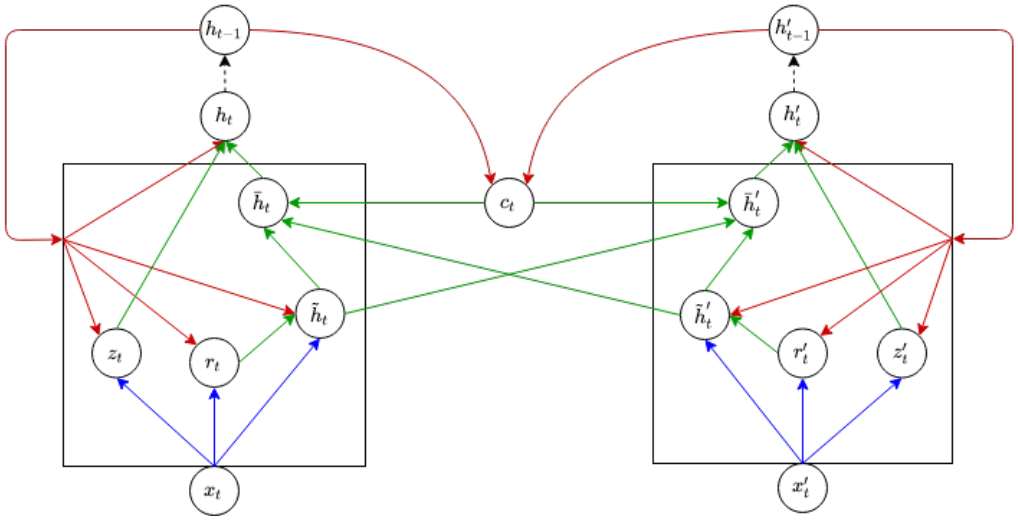


Fig. 1: Schema of the CGRU architecture. Blue arrows represent information coming from the input at time t, red ones are for the hidden states and green ones for transmissions between the gates.

GRU is defined by fewer equations while showing comparable performance in general [37]. The following equations describe the modifications brought to the architecture. Update, Reset and New gates are not modified. Let us notate h'_{t-1} and $\tilde{h}'_t$ the states coming from the second sub-network (see Figure 1):

$$\text{Hidden: } h_t = (1 - z_t)\bar{h}_t + z_t h_{t-1} \tag{8}$$

$$\text{Update: } z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \tag{9}$$

$$\text{Coupled New: } \bar{h}_t = (1 - c_t)\tilde{h}_t + c_t\tilde{h}'_t \tag{10}$$

$$\text{Coupling: } c_t = \sigma(W_{hc}(h_{t-1} + h'_{t-1}) + b_{hc}) \tag{11}$$

$$\text{New: } \tilde{h}_t = \tanh(W_{i\tilde{h}}x_t + b_{i\tilde{h}} + r_t(W_{h\tilde{h}}h_{t-1} + b_{h\tilde{h}})) \tag{12}$$

$$\text{Reset: } r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}). \tag{13}$$

The Coupling gate c_t (see Equation 11) serves the same purpose as z_t and r_t, controlling the information flow and is thus computed in a similar manner, but only from the hidden states. This forces the model to apply the coupling on the new content to be added at time t solely based on the previous inputs. Then, in Equation 10, $\tilde{h}_t$ and $\tilde{h}'_t$ are combined similarly as $\tilde{h}_t$ and h_{t-1} are combined in the original GRU architecture. This prevents $\bar{h}_t$ and subsequently h_t from exploding and saturate the gates. Finally, in Equation 8, $\bar{h}_t$ replaces $\tilde{h}_t$: the New state has been replaced by a coupled version of both New states of the siamese GRU. Several possibilities exist to implement this coupling. The idea behind this proposal is to alter as little as possible the GRU architecture and to stay close to the original purpose of each equation. Indeed, the addition of the coupling already greatly modifies the information flow inside the GRU and the gradient flow during training, and RNNs are known to be difficult to train. Therefore, by staying relatively close to the original model, a rigorous comparison should be more effective, and the impact of the actual coupling can be studied more reliably. In fact, if c_t is a vector of norm equal to zero, each sub-network is exactly a GRU. This suggests to initialize the coupling weights with very small values and to accentuate the decay. In this way, an increase of the norm of W_{hc} during training would signify that coupling is useful. Another interesting configuration of the coupling weights is when they are all equal to 0.5: in this configuration, the Coupled New States are the same, and the distance between the outputs will become null. That means, theoretically, this approach can make close any pair of input sequences, especially hard-positive samples.

IV. Experiments

A. Experimental setup

We experiment the CGRU architecture on two datasets. The first one is UCI HAR [38], a dataset of 6 activities[2] containing 9 features: total acceleration, body acceleration and angular velocity on 3 axes. The sequences have a length of 128. We chose this dataset because it has been extensively used by the activity recognition community and provides at the same time real data and a simple and well defined benchmark to study the behavior of CGRU and Siamese GRU (SGRU). Moreover, several activities should look very similar (e.g. three variants of walking or standing and sitting), and it should make the dataset harder to process for metric learning algorithms. Finally, *walking* or *running* exhibits dynamic components which could be differently processed by CGRU and SGRU. No further preprocessing has been applied. The features are globally centered and the standard deviations oscillate between 0.1 and 0.4. We kept the train-test split proposed by the authors of the dataset: there are 21 users in the training set and we therefore performed a 7-fold validation, leaving each time 3 different users out. Finally, the training set comprises 7352 sequences and the testing set 2947.

[2]walking, walking upstairs, walking downstairs, sitting, standing, laying

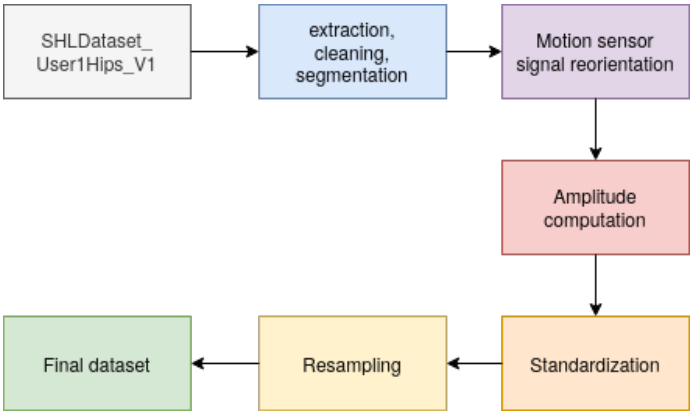


Fig. 2: Preprocessing steps applied to SHL dataset, we globally followed the process proposed by Janko et al. [41].

The second dataset was the object of a challenge in 2018 [39], the Sussex-Huawei Locomotion and Transportation (SHL) Dataset [40]. The data were recorded by a single individual during a 4 month period for a total of 82 days: 62 for training, 20 for testing. Up to 8 hours of data are recorded each day. This dataset proposes 8 locomotion transportation modes: car, bus, train, subway, walk, run, bike and standing still. It contains 20 features: accelerometer, gyroscope, magnetometer, gravity and linear acceleration on three axes, orientation on 4 axes and pressure. The raw dataset is sampled at a frequency of 100 Hz, that is to say, one hour sequences have a length of 360000. To use it for our experiments, we roughly followed the same preprocessing procedure as in [41], with some more steps (see Figure 2): all signals (except pressure) were reoriented to the North-East-Down axes convention and magnitudes were computed for the accelerometer, the gyroscope and the magnetometer. Orientations were converted to Euler angles. This process extends the number of features in the dataset from 20 to 36. Then, the dataset was standardized according to the training set to have, for a every feature, a mean of zero and a standard deviation of 1. Finally, to speed up computations, the 1 minute sequences we used of a length of 6000 were resampled to a length of 300. The challenge summary paper [39] reports that the best test accuracy result of 93.9% was achieved by Gjoreski et al. [42] with a deep learning approach.

We compared SGRU and CGRU on learning the metric describes in Equation 5 using the same hyperparameters for both models, those parameters for each datasets are displays on Table I along with some statistics on the training and testing set. The training is stopped based on the accuracy on the validation set (early stopping). We chose to use structural loss [2] to train the model since it works on distances and not embeddings. It combines a local term similarly to the n-pair loss [11] but emphasizes the weights on the hard positive samples, and a global term to improve the generalization. We used the same hyperparameters for the loss as in the original

Hyperparameters	UCI HAR	SHL
Training set size	7352	15660
Testing set size	2947	5472
Architecture	[20]	[100, 100]
Batch size	36	40
Initial learning rate	0.001	0.001

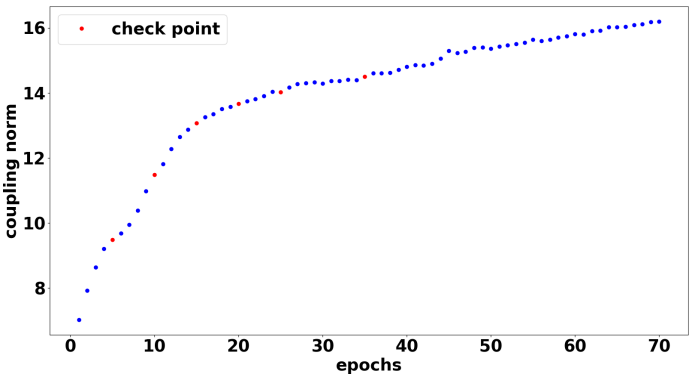
TABLE I: Main hyperparameters used to train our models on both datasets. These parameters are exactly the same for SGRU and CGRU.

paper. We applied a general weight decay with a factor of 10^{-4}. A stronger weight decay was applied on the coupling by adding 1% of the coupling weight norm to the loss, which seems to slightly improve the performances. The gradient was clipped according to [43] to a norm of 6. The coupling weights were initialized with a normal distribution having a mean of 0 and standard deviation of 0.1. The purpose of this initialization is to make the model start its training close to the behavior of an SGRU, with a very weak coupling and to let it increase during training. Our implementation was done in Python and CUDA with Pytorch [44].

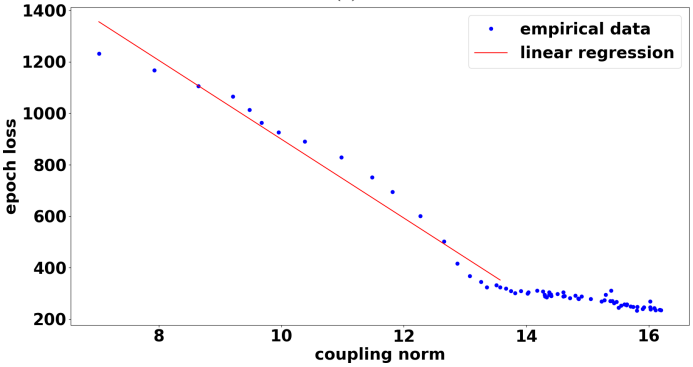
B. Results

1) Study of the coupling weight norm: We first analyze the evolution of the coupling weight norm during training. The coupling is initialized very low which theoretically makes it behave nearly as an SGRU at the beginning of training. We can observe in Figure 3a that the norm increases quickly during the first 20 epochs and more than doubles. Red points indicate the iterations where validation accuracy increased. This correlation thus indicates that the overall generalization is improving with the increase of coupling strength. In Figure 3b, we present the evolution of the loss on the train dataset in terms of the coupling weights on which we compute a linear regression of the first part. We can observe an almost linear relationship between the increase of the norm and the decrease of the loss suggesting again that the coupling is helpful to the model. Those figures also show that the convergence during training is rather smooth.

2) Classification performance on UCI HAR: We now present classification results on UCI HAR using 3 metrics: accuracy, F1 score Macro averaged and Mean Average Precision (MAP), similarly to [13]. The first two are computed by classifying the test samples using 1-nearest neighbor from training samples. The MAP is computed by querying the training set with a test sequence to retrieve all training samples of the same class. The value is averaged for all test sequences. This metric shows the ability of the algorithm to bring close every sequence of each class and not just few references to be used as nearest neighbors. The validation results are presented in Table IIa. We observe an improvement of CGRU over SGRU of about 8% points for accuracy and F1-score, and an improvement of 19% points for the MAP. On the test set, we compared our approach with Regressive Virtual Sequence Metric Learning (RVSML)



(a)



(b)

Fig. 3: In figure (a), evolution of the norm of the coupling gate weights during training on UCI HAR dataset, a red point indicates an increased accuracy for the validation set. On figure (b), Epoch loss in terms of coupling norm during training on UCI HAR dataset. The first part (rapid decrease) has been approximated with a linear regression. The correlation coefficient is -0.984.

[13], with OPW and DTW distances. We chose to compare with this approach because it recently outperformed several other metric learning approaches (e.g. Large Margin Nearest Neighbors [45], Regressive Virtual Metric Learning [12], etc.) although not all were specifically adapted for sequences, it is not based on neural networks and uses alignment-based distances. We keep the hyperparameter values recommended by the authors. The test results are presented in Table IIb. Here again we observe that CGRU outperformed SGRU with a notable improvement of 10% points of the accuracy and F1-score. Both neural network approaches clearly outperformed RVSML, especially the OPW variant. This can be, among other factors, attributed to a weak capacity to distinguish similar activities such as *standing* and *sitting*. On the other hand, *standing* was recognized perfectly by both RVSML variants. We also remark that they reach MAP values of the same order as in the original paper (around $0.4 \sim 0.45$) despite the fact that the data are of a completely different nature (signal instead of images). This could suggest that these approaches reached some kind of saturation whereas CGRU and SGRU are able to achieve much higher values.

Algorithms	Accuracy	F1 score Macro	MAP
Siamese GRU	0.835±0.068	0.827±0.074	0.711±0.016
Coupled GRU	**0.913±0.055***	**0.916±0.054***	**0.900±0.043***

(a) Validation results (21 fold average). An asterisk means a significant result with a threshold of 1%

Algorithms	Accuracy	F1 score Macro	MAP
RVSML (OPW)	0.597	0.568	0.438
RVSML (DTW)	0.698	0.687	0.437
Siamese GRU	0.782±0.041	0.781±0.044	0.633±0.152
Coupled GRU	**0.885±0.014***	**0.887±0.014***	**0.899±0.01***

(b) Test results, average of 5 runs for the neural networks approaches.

TABLE II: Results on UCI HAR

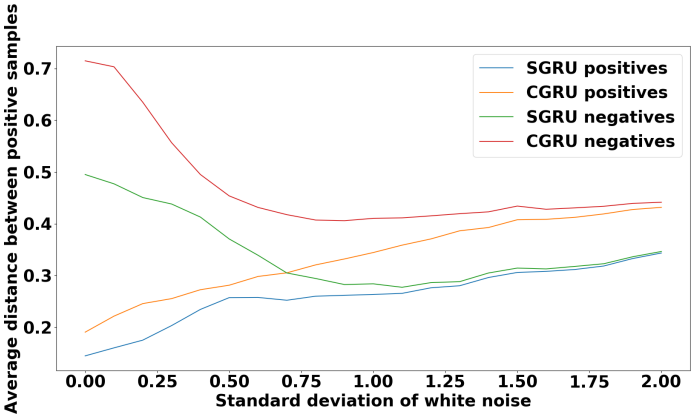


Fig. 4: Evolution of the average distance between the positive/negative samples for SGRU and CGRU on more and more noisy test sets.

3) Performances on hard positive samples: We made the hypothesis that CGRU could perform better on hard positive samples due to coupling theoretically being able to bring close any input pair of sequences: with enough coupling, both networks can output the same sequence whatever of the input pair. We propose to verify this by observing the evolution of the average distance between the positive samples and between the negative samples when white noise is gradually added to the feature sequences of the testing set. The standard deviation of the noise goes from 0 to 2. We also note that both models were trained up to comparable validation accuracy. The results are presented on Figure 4. We observe that the curves for both models evolve similarly with positive distances gradually increasing with the noise. The negative and positive curves join the moment too much noise is added and the sequences become indistinguishable. CGRU produces slightly higher distance average than SGRU but is able to maintain higher margins more longer: up to 1.75 units of standard deviation compared to about 1.25 for SGRU. This shows that, as theoretically possible with the coupling, CGRU better discriminate the hard samples.

4) Transportation recognition on SHL: We finally present a second experiment on a larger dataset with twice as much data

Algorithms	Accuracy	F1 score Macro	MAP
Siamese GRU	0.947±0.006	0.951±0.006	0.955±0.007
Coupled GRU	**0.965±0.05**	**0.967±0.006**	**0.97±0.004**

(a) Validation results (10 fold average).

Algorithms	Accuracy	F1 score Macro	MAP
Siamese GRU	0.727±0.019	0.746±0.016	0.784±0.019
Coupled GRU	**0.752±0.021**	**0.768±0.016**	**0.802±0.018**

(b) Test results (average of 5 runs).

TABLE III: Results on SHL dataset.

and a longer sequence size. The validation results are presented in the Table IIIa. Both architectures achieved results above 90% for each metric but we observe a slight improvement of CGRU over SGRU for the considered architecture. These results are further confirmed on the test set (see Table IIIb) with even larger improvements: 2,5% for accuracy, 2.2% for F1-score and 1.5% for MAP, in favor of CGRU. These results confirm the interest of our proposed approach CGRU compared to SGRU. However, compared to the overall results of the challenges [39], both approaches achieved similar validation results but clearly do not demonstrate the same generalization capacities on this dataset as the best ad-hoc approaches.

V. Conclusion and Perspectives

We presented a new framework for sequence metric learning based on dynamical system synchronization theory. We drew a parallel between synchronized trajectories and output sequences of siamese recurrent neural networks produced from similar input pairs. After characterizing Siamese GRU as identical chaotic systems, we showed the contribution of introducing coupling inside the siamese architecture to achieve synchronization more easily and to increase the capacity of the network to bring closer the embedding of some similar input pairs, especially hard positive samples. This coupling was implemented through a new gate inside the siamese GRU architecture which allows the network to mix the new content of both sides of the siamese network. Our experiments showed that the siamese GRU architecture benefits from the coupling, can be smoothly trained with it and fits well with recent complex metric learning losses such as structural loss. CGRU proved to be outperforming SGRU with the same architecture on an activity recognition benchmark and was able to maintain a higher margin for hard samples. On a larger dataset made for transportation recognition, CGRU also achieved higher performances than SGRU.

The study of sequence metric learning with synchronization opens several perspectives especially to design new forms of metrics by taking inspiration from the literature of synchronization criteria [35], [46] though non-differentiability, notably when those metrics use mutual neighbors which are computed using explicitly the time steps, has to be overcome in many cases. This can be done thanks to an attention mechanism, for example. The coupling itself could be tuned with theoretical contributions [47]. One drawback of the proposed architecture is that each pair has to be passed through the network instead of just computing once each representation and then the distance for each pair. This could be balanced by the use of virtual metric learning during training. Finally the coupling allows to bring any pair of inputs close to one another if sufficiently strong and could be use as an indicator in weakly supervised settings to invert the equivalence constraint of some pairs dynamically if the network is forcing the synchronization too much.

References

[1] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2003, pp. 521–528.

[2] X. Yang, P. Zhou, and M. Wang, "Person reidentification via structural deep metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2987–2998, 2018.

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[4] S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Class-balanced siamese neural networks," *Neurocomputing*, vol. 273, pp. 47–56, 2018.

[5] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity." in *AAAI*, 2016, pp. 2786–2792.

[6] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[10] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4004–4012.

[11] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016, pp. 1857–1865.

[12] M. Perrot and A. Habrard, "Regressive virtual metric learning," in *Advances in neural information processing systems*, 2015, pp. 1810–1818.

[13] B. Su and Y. Wu, "Learning distance for sequences by learning a ground metric," in *International Conference on Machine Learning*, 2019, pp. 6015–6025.

[14] X. Cai, T. Xu, J. Yi, J. Huang, and S. Rajasekaran, "Dtwnet: a dynamic time warping network," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 636–11 646.

[15] A. Abid and J. Zou, "Autowarp: learning a warping distance from unlabeled time series using sequence autoencoders," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 10 568–10 578.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[17] K.-i. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.

[18] M. W. Hirsch, "Convergent activation dynamics in continuous time networks," *Neural Networks*, vol. 2, no. 5, pp. 331–349, 1989.

[19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[20] T. Laurent and J. von Brecht, "A recurrent neural network without chaos," *arXiv preprint arXiv:1612.06212*, 2016.

[21] B. Chang, M. Chen, E. Haber, and E. H. Chi, "Antisymmetricrnn: A dynamical system view on recurrent neural networks," *arXiv preprint arXiv:1902.09689*, 2019.

[22] R. J. Martin, "A metric for arma processes," *IEEE transactions on Signal Processing*, vol. 48, no. 4, pp. 1164–1170, 2000.

[23] I. Ishikawa, K. Fujii, M. Ikeda, Y. Hashimoto, and Y. Kawahara, "Metric on nonlinear dynamical systems with perron-frobenius operators," in *Advances in Neural Information Processing Systems*, 2018, pp. 2856–2866.

[24] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 1033–1040.

[25] B. Su and G. Hua, "Order-preserving wasserstein distance for sequence matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1049–1057.

[26] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *European conference on computer vision*. Springer, 2016, pp. 791–808.

[27] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical review letters*, vol. 64, no. 8, p. 821, 1990.

[28] R. Brown and L. Kocarev, "A unifying definition of synchronization for dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 10, no. 2, pp. 344–349, 2000.

[29] A. De Angeli, R. Genesio, and A. Tesi, "Dead-beat chaos synchronization in discrete-time systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 1, pp. 54–56, 1995.

[30] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, "The synchronization of chaotic systems," *Physics reports*, vol. 366, no. 1-2, pp. 1–101, 2002.

[31] H. Fujisaka and T. Yamada, "Stability theory of synchronized motion in coupled-oscillator systems," *Progress of theoretical physics*, vol. 69, no. 1, pp. 32–47, 1983.

[32] S. H. Strogatz, "Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology," *Chemistry, and Engineering. CRC Press*, 2018.

[33] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[34] N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. Abarbanel, "Generalized synchronization of chaos in directionally coupled chaotic systems," *Physical Review E*, vol. 51, no. 2, p. 980, 1995.

[35] S. J. Schiff, P. So, T. Chang, R. E. Burke, and T. Sauer, "Detecting dynamical interdependence and generalized synchrony through mutual prediction in a neural ensemble," *Physical Review E*, vol. 54, no. 6, p. 6708, 1996.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[38] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.

[39] L. Wang, H. Gjoreskia, K. Murao, T. Okita, and D. Roggen, "Summary of the sussex-huawei locomotion-transportation recognition challenge," in *Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers*, 2018, pp. 1521–1530.

[40] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen, "The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[41] V. Janko, N. Rešçiç, M. Mlakar, V. Drobnič, M. Gams, G. Slapničar, M. Gjoreski, J. Bizjak, M. Marinko, and M. Luštrek, "A new frontier for activity recognition: The sussex-huawei locomotion challenge," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1511–1520.

[42] M. Gjoreski, V. Janko, N. Reščič, M. Mlakar, M. Luštrek, J. Bizjak, G. Slapničar, M. Marinko, V. Drobnič, and M. Gams, "Applying multiple knowledge to sussex-huawei locomotion challenge," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1488–1496.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[45] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[46] T. Kreuz, F. Mormann, R. G. Andrzejak, A. Kraskov, K. Lehnertz, and P. Grassberger, "Measuring synchronization in coupled model systems: A comparison of different approaches," *Physica D: Nonlinear Phenomena*, vol. 225, no. 1, pp. 29–42, 2007.

[47] R. Brown and N. F. Rulkov, "Designing a coupling that guarantees synchronization between identical chaotic systems," *Physical review letters*, vol. 78, no. 22, p. 4189, 1997.