A Robust Dissimilarity-based Neural Network for Temporal Pattern Recognition

Brian Kenji Iwana*†, Volkmar Frinken‡§, Seiichi Uchida*
*Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan
Email: brian@human.ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp
†Institute of Decision Science for a Sustainable Society, Kyushu University, Fukuoka, Japan
‡ONU Technology, San Jose, USA
§Electrical and Computer Engineering, University of California Davis, Davis, USA

***Abstract*—Temporal pattern recognition is challenging because temporal patterns require extra considerations over other data types, such as order, structure, and temporal distortions. Recently, there has been a trend in using large data and deep learning, however, many of the tools cannot be directly used with temporal patterns. Convolutional Neural Networks (CNN) for instance are traditionally used for visual and image pattern recognition. This paper proposes a method using a neural network to classify isolated temporal patterns directly. The proposed method uses dynamic time warping (DTW) as a kernel-like function to learn dissimilarity-based feature maps as the basis of the network. We show that using the proposed DTW-NN, efficient classification of on-line handwritten digits is possible with accuracies comparable to state-of-the-art methods.**

***Keywords*-Dynamic Time Warping; Neural Networks; Temporal Pattern Recognition; Deep Learning; Time Series**

I. Introduction

Temporal patterns are a vital part of life and come in many forms, such as sound, handwriting, gestures, signals, trends, and any other data that is dependent on time. Similar to other structural patterns, temporal patterns are complex and difficult due to the imporance of sequence order and data element connectivity. In addition, temporal patterns are subject to temporal distortions such as variable rate and inconsistent lengths. To overcome the difficulties and capture the relevant aspects of temporal patterns, methods specifically designed for temporal pattern recognition must be used.

Dynamic time warping (DTW) is a popular algorithm for determining the distance between two temporal patterns [1]. Unlike the direct approach to naively matching elements, DTW overcomes the temporal distortions by elastically matching sequence elements under constrained restrictions. Paired with a distance-based classifier such as k-Nearest Neighbors (k-NN), it has shown to be an effective distance measure between temporal patterns [2]. When used with large quantities of data, however, k-NN becomes prohibitively computationally intensive and not suitable for some applications due to its exhaustive search nature [3]. Some common solutions to overcome the computational requirement include reducing the size of the patterns, feature representation, and prototype selection.

In recent times, artificial neural networks have been a large focus of research due to their recent successes in pattern recognition and machine learning [4]. Inspired by biological neural networks, artificial neural networks function with layers of interconnected nodes, each with input stimuli and activated output signals. By stacking many layers of neurons, a network has the ability to learn complex functions, generally referred to as *deep learning*. Temporal pattern deep learning is also an active field of research, with successes in natural language processing [5], speech [6], and handwriting [7], [8]. Particularly the property to interpret nodes of a neural network as a feature extraction, and hence the ability to automatically derive and evaluate features from various input sources is promising.

To exploit the feature learning and classification for temporal patterns, we propose a novel method of using a DTW-layer within a neural network to create dissimilarity-based temporal feature maps. The objective of this is to preserve the nature of temporal patterns of variable sequence length while still using a feed-forward neural network. By considering DTW as a similarity between input patterns and filter patterns, the direct application of temporal patterns into the network is possible without rasterization or other transformations. Also, unlike other methods employing DTW, we use DTW as integral part of the neural network and hence also part of the training process.

The primary contribution of this paper is the presentation of a new neural network model called a Dynamic Time Warping Neural Network (DTW-NN). We demonstrate the possibility of learning isolated temporal pattern directly and unmodified. We also show the possibility of developing temporal filters with back-propagation through the DTW layer. Lastly, we show that it is an effective and efficient method of classifying on-line handwritten digits.

The remaining of this paper is organized as follows. Section II reviews the related literature in neural learning with temporal patterns. Section III details DTW and its application to kernels for machine learning. In Section IV, we propose a DTW-NN framework. Section V describes the architecture for using our network for on-line handwritten digit recognition. Finally, Section VII draws a conclusion and outlines future work.

II. Related Works

Neural network approaches to time series recognition is an active field of research. A common approach is to feed the model a *sliding window* view of the data giving each input a step in time. Recurrent Neural Networks (RNN) [9] use this method to read the patterns into layers of recurrent nodes, or nodes that connect back to themselves. A Long-Short Term Memory (LSTM) [10] network is a model built upon RNNs to learn data with long time lags. Alternatively, Time Delay Neural Networks (TDNN) [11], [12] are feed-forward networks that also use sliding window.

Convolutional Neural Networks (CNN), however, process entire inputs as pixel data and learns from the spacial structure. A common method of using CNNs for on-line handwriting recognition is to rasterize the data, or render the temporal pattern in pixel-space [13], [14]. Zheng, et al. [15] use channels of serialized 1D subsequences instead of 2D pixels to classify multivariate time series'. However, these methods of preprocessing are not relevant to many temporal patterns, such temporal patterns with very high dimensional elements or pattern with sporadic elements.

III. Dynamic Time Warping

A. DTW Definition

DTW determines the similarity between temporal patterns by calculating the summation of the distances between the optimally matched elements of sequences. Given a prototype sequence $\mathbf{p} = p_1, \ldots, p_i, \ldots, p_I$ and a sample sequence $\mathbf{s} = s_1, \ldots, s_j, \ldots, s_J$, the global difference between pattern is the collection of distances between the pairwise element matches. Where Q is the set of all sample elements matched to a prototype element, DTW is defined as

$$\mathrm{DTW}(\mathbf{p}, \mathbf{s}) = \sum_{s_q \in Q} ||p_i - s_q|| \,. \tag{1}$$

To determine the elements matched in Q, the experiment assumes the step size and slope constraints given by

$$D(i,j) = c(p_i, s_j) + \min_{j' \in \{j, j-1, j-2\}} D(i-1, j'), \tag{2}$$

where $D(i,j)$ is the cumulative cost at the i-th and j-th element. $c(p_i, s_j)$ is the cost matrix where c is the local distance measure between each element of the sequences $\mathbf{p}$ and $\mathbf{s}$ respectively. In other words, DTW is the minimization of the warping path on the cost matrix between $\mathbf{p}$ and $\mathbf{s}$.

The asymmetric slope constraint given by Eq. (2) provides DTW with a distance measure that is length and temporal fluctuation invariant. This slope constraint was selected because it assures that the number of element matches is always equal to the number of elements I in $\mathbf{p}$. With equal sized prototype sequences, DTW provides a robust method for the comparison of sample patterns unequal lengths.

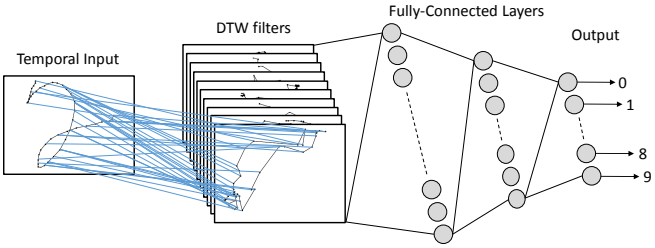


Figure 1. Illustration of the proposed DTW-NN architecture.

B. DTW and Kernels

In common artificial neural networks, a vectorial data representation is given to the input layer, with one node for each entry on the input vector. Then, the data is propagated forward, one layer at a time. Usually, in each layer, the activation of a node j with activation a_j^{n+1} in layer $n+1$ has the form

$$a_j^{n+1} = \phi\left(\sum w_{ij} a_i^n\right), \tag{3}$$

where w_{ij} is a weight, a_i^n the activation of node i in layer n, and $\phi(\cdot)$ an activation function, such as sigmoid, hyperbolic tangent function, or a rectified linear unit (ReLU) [16]. All these activation functions are monotonically increasing. Hence if the normalized weight vector $w_{i\cdot}$ and the normalized output of the of the previous layer a_i^n are similar, we can expect a larger activation a_j^{n+1}. In this sense, each weight-node combination can be seen as similarity function, not unlike a kernel, although the positive semidefiniteness is not given. From this point of view, each layer of a feed-forward neural network can be seen as dissimilarity space embedding [17] with the prototypes as the column vectors of the weight matrix between two layers.

We adopt this idea in this paper by considering the DTW distance of an input sequence to a given filter sequence as the node activation. In the past, attempts to adapt Support Vector Machines (SVM) to use temporal patterns were done by using DTW kernels [18], DTW for time alignment [19], and incorporating DTW as the distance measure for a RBF kernel [20].

IV. Dynamic Time Warping Neural Network (DTW-NN)

The architecture of the proposed method is detailed in this section. Training is accomplished using rounds of forward feeding the network and back propagating the errors to adjust the weights.

A. Forward Propagation of DTW

We propose using a DTW as a kernel-like method as a replacement for the typical node activation of a neural network. So, rather than extracting local features, we propose a model that uses the temporal features directly learned from temporal filters. To accomplish this, DTW is used to extract

similarity information from a temporal pattern input, as seen in Fig. 1.

During forward propagation, the DTW is calculated between the input and the temporal filter. Given the input treated as the sample $\mathbf{s}$ and the filter as the prototype $\mathbf{p}$, the DTW-distance is determined by the value of Eq. (2) at $D(I, J)$. DTW replaces the standard mechanism for combining the weights and outputs of the previous layer in Eq. (3), thus the activation function $\phi(\cdot)$ is passed the result of DTW. Due to the characteristic of DTW where a greater DTW-distance equates to a larger difference between temporal patterns, activation function is inversed. The result was further rescaled as suggested in [21] to values between -1 and 1.

B. Back Propagation through DTW

In the training of a neural network, back propagation is the step to update the weights of the network to minimize the error of the network. To adapt the temporal filters to minimize the error of the DTW-NN, we must consider back propagation through DTW. Therefore, the derivative of the error function with respect to DTW is computed, specifically, using the chain rule we find

$$\frac{\partial E}{\partial W_{\mathrm{DTW}}} = \frac{\partial E}{\partial z_{\mathrm{DTW}}} \frac{\partial z_{\mathrm{DTW}}}{\partial W_{\mathrm{DTW}}} \tag{4}$$

where E is the loss function of the output, W_{DTW} is the parameters of the DTW filter, and z_{DTW} is the result of the DTW filter. $\frac{\partial z_{\mathrm{DTW}}}{\partial W_{\mathrm{DTW}}}$ is the relationship between the elements of each sample and the result of DTW, namely, it is the derivative of the DTW function between the temporal filter $\mathbf{p}$ and the input $\mathbf{s}$, or $\mathrm{DTW}'(\mathbf{p}, \mathbf{s})$.

As detailed in Eq. (1), DTW is the sum of the distances between each pairwise element match of the sequences. In the example of two-dimensional coordinate sequences, each element $p_i = (x_{p_i}, y_{p_i})^T$ in the prototype sequence $\mathbf{p}$ has a matching element $s_q = (x_{s_q}, y_{s_q})^T$ in the sample sequence $\mathbf{s}$, and the DTW is the sum of the distance measures between them. The derivative of DTW with respect to $(x_{p_i}, y_{p_i})^T$ becomes

$$\mathrm{DTW}'(\mathbf{p}, \mathbf{s}) = \frac{\partial}{\partial (x_{p_i}, y_{p_i})^T} \mathrm{DTW}(\mathbf{p}, \mathbf{s}) = \sum_{(x_{s_q}, y_{s_q})^T \in Q} \frac{\partial}{\partial (x_{p_i}, y_{p_i})^T} \left|\left| \begin{array}{c} x_{p_i} - x_{s_q} \\ y_{p_i} - y_{s_q} \end{array} \right|\right|. \tag{5}$$

DTW can be calculated with various distance measures. The experiment assumes a Euclidean distance as the distance measure between elements. The derivative of the Euclidean distance with respect to each parameter gives the partial derivatives:

$$\frac{\partial}{\partial x_{p_i}} \left|\left| \begin{array}{c} x_{p_i} - x_{s_q} \\ y_{p_i} - y_{s_q} \end{array} \right|\right| = \frac{x_{p_i} - x_{s_q}}{\sqrt{(x_{p_i} - x_{s_q})^2 + (y_{p_i} - y_{s_q})^2}} \tag{6}$$

$$\frac{\partial}{\partial y_{p_i}} \left|\left| \begin{array}{c} x_{p_i} - x_{s_q} \\ y_{p_i} - y_{s_q} \end{array} \right|\right| = \frac{y_{p_i} - y_{s_q}}{\sqrt{(x_{p_i} - x_{s_q})^2 + (y_{p_i} - y_{s_q})^2}}. \tag{7}$$

This assumes a two-dimensional task but it can be solved in a similar manner for any number of dimensions.

Finally, using the derivative of the Euclidean distance with respect to every coordinate $(x_{p_i}, y_{p_i})^T$ in $\mathbf{p}$, we can calculate the derivative of DTW, Eq. (5), and substitute it into Eq. (4) to calculate the gradient of the error function with respect to DTW. This gives us the ability to adjust the DTW filter to reduce the value of the error function by moving each of the coordinates by a learning rate of η,

$$\begin{pmatrix} x_{p_i} \\ y_{p_i} \end{pmatrix} \leftarrow \begin{pmatrix} x_{p_i} \\ y_{p_i} \end{pmatrix} - \eta \frac{\partial E}{\partial W_{\mathrm{DTW}}}. \tag{8}$$

V. On-line Handwritten Digit Recognition with DTW-NN

Isolated on-line handwriting provides us with an ideal source of temporal patterns because while there is a high variation between patterns, they can still be classified into the ten distinct classes. This high variation can even exist within classes, due to stroke order, stroke direction, connectivity, serif, etc.

A. Data Set

The experiment was done using the Unipen on-line handwritten digit data set, UNIPEN multi-writer 1a. The data set consists of about 15,000 isolated handwritten digit patterns in ten digit classes. It was collected by the International Unipen Foundation (iUF), which is a popular source for benchmark data sets for writer identification and handwriting recognition [22].

The data sets were split into four subsets of randomly selected patterns. The training data set, validation data set, and test data set used 11,450 patterns, 200 patterns, and 1,300 patterns respectively. A final subset of 50 patterns was set aside as initial temporal filters for a digit initialization evaluation. Using this division in the data set, we trained the neural network for 10-class classification. The total number of patterns used is smaller than the full set to have an equal representation of each class. The patterns were chosen at random and no cleaning was done.

B. DTW-NN Architecture

To tackle the digit recognition task, we employed a network with four weighted layers consisting of one DTW layer, two fully-connected layers, and the fully-connected output layer. The DTW layer was designed as per Section IV with 50 temporal filters nodes of 50 sequence elements each. For the two fully-connected layers, we used a $\tanh$ activation function with an initial bias of 0. The two fully-connected layers were made of 400 and 100 nodes respectively. The final output layer used a 10-node softmax

function to normalize the output, representing the probability of each of the class labels.

The network was trained with batch gradient decent using a batch size of 200 digits over the course of 40,000 training rounds. A constant learning rate of 0.0001 was used for the fully-connected layers and a larger learning rate of 0.1 was used for the DTW layer. Different learning rates were required because of the difference in magnitude between the elements in the temporal patterns and the weights between the other layers.

VI. Experimental Results

A. Evaluation of the Proposed Method

For the evaluation of the proposed method on ten-class classification with on-line handwritten digits, we considered the following evaluations:

- *Proposed Method - DTW-NN (Gaussian Init.)*: The proposed DTW-NN architecture was used. The initial DTW filters were constructed from a series of randomly generated points. The points were selected from an isotropic Gaussian with a mean of $(65, 65)$ and a covariance of 195, which puts them in the general range of the patterns in the data set. This evaluation represents a random weight initialization.
- *Proposed Method - DTW-NN (Digit Init.)* This trial also uses the same DTW-NN architecture as the proposed method with Gaussian initialized filters. The difference is the initialization of the starting DTW filters. Instead of randomly generated temporal patterns, the filters were randomly pre-selected from the data set prior to the division of subsets. By pre-selecting patterns, we start this weight initialization with structure similar to the data set.
- *Raster Image CNN*: For this evaluation, the handwritten digits were rasterized into (56×56) pixel single channel images. A standard image-based CNN was built using the same activation functions and the same number of layers and nodes as the proposed method.
- *1-NN with DTW*: This is a baseline evaluation using the using k-NN where k is equal to 1. DTW is used as the distance measure between patterns. Unlike the other trials with trained models, testing was done between the entire test set and entire training set.
- *DAG-SVM-GDTW* [20]: Results reported by Bahlmann, et al. using an SVM with DTW embedded in an RBF kernel. The reported results use 40% of the UNIPEN data set for training and 40% for testing.
- *HMM* [23]: Results reported by Hu, et al. using an Hidden Markov Model (HMM). The data set was split 2/3 training and 1/3 testing. Note, the data set was cleaned and mislabeled data (about 4% of the total data) was removed prior to the experiment.

[1]This result reflects a data set with mislabels removed.

Table I
A table of comparisons of the evaluated methods in recognition rate and average per digit classification time.

Evaluation	Recognition Rate	Ave. Per Digit Time
DTW-NN (Gaussian Init.)	96.8%	0.3s
DTW-NN (Digit Init.)	96.8%	0.3s
Raster Image CNN	95.6%	–
1-NN with DTW	98.5%	77.6s
DAG-SVM-GDTW [20]	96.2%	–
HMM [23]	96.8%[1]	–

For DTW-NN (Gaussian Init.), DTW-NN (Digit Init.), and 1-NN evaluations, the average classification time of a single digit was recorded using a system with Intel Xeon E5 2.6 GHz processor.

The results of the evaluations are shown in Table I. The proposed method achieved a 10-class classification accuracy of 96.8% and 96.8% for Gaussian initialization and random digit initialization respectively. While 1-NN with DTW outperformed all other methods in accuracy, the slow single digit classification time of 77.6 seconds is unreasonable and not suitable for real time or large data applications. In comparison, the DTW-NN evaluations achieved a high rate of accuracy while taking a small fraction of the time, at 0.3 seconds each. The computational time improvement is due to only requiring DTW calculations between the inputs and the learned filters rather than the entire training set as required in 1-NN. The DTW-NN methods had the best recognition rate in comparison to the other trained models. This shows that the proposed method is both an efficient and effective model for temporal pattern classification.

In addition, when comparing the results of the Raster Image CNN to the DTW-NN, the results show that the proposed method was able to overcome the temporal distortions and stroke directions issues that are not present in the off-line rasterizations of the data. This is important because not all temporal patterns offer easy rasterizations to image-space like handwriting does. For instance, high-dimensional temporal patterns or temporal patterns with independent dimensions might lead to difficulties with conventional image-space CNNs.

A significant source of the error from the DTW-NN model comes from mislabels and noise found in the UNIPEN data set. Figure 2 shows examples of the misclassifications of the test set on the DTW-NN (Digit Init.) evaluation. Many of the errors made by the network are reasonable because the digits were either heavily distorted or ground truth mislabels. The highlighted patterns in Fig. 2 would be difficult to classify even for humans. It is interesting to note, even though the two evaluations had different starting filters and different initial layer weights, both networks had difficulty with many of the same patterns. 52.4% of the misclassifications were shared by the two initializations.

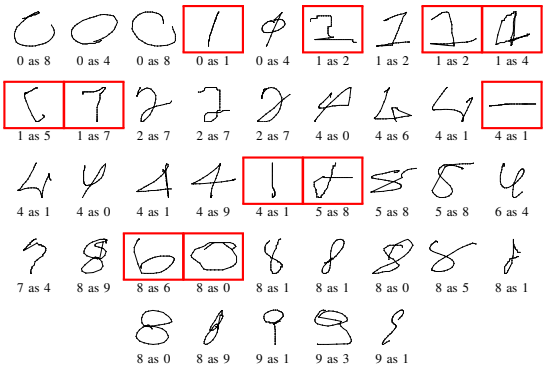


Figure 2. All of the misclassified using the DTW-NN (Digit Init.) method sorted by ground truth label. Each figure is labeled with the ground truth and the predicted class, where "x as y" means that the ground truth class is given as x and the system classified the input as y. The figures enclosed in red boxes are patterns that are mislabeled, ambiguous, or noise.

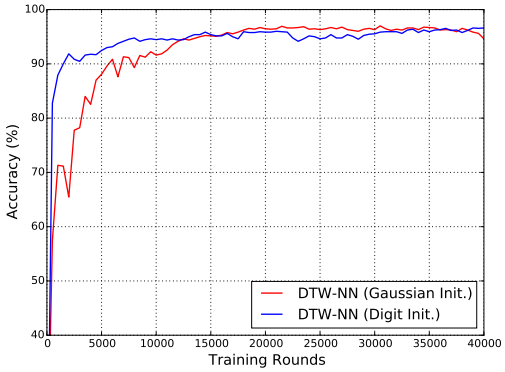


Figure 3. A graph portraying the accuracy of 10-class classification over number of batch rounds of training.

B. Observations on the Filters

The two filter preparations were done to observe the difference between utilizing randomly generated patterns and patterns with a preset structure. Figure 3 demonstrates that after enough time, there was little difference in the method of initializing the DTW filters. However, the accuracy of the Gaussian initialization took longer to stabilize and reach a similar accuracy rate as initializing the network with preset digits.

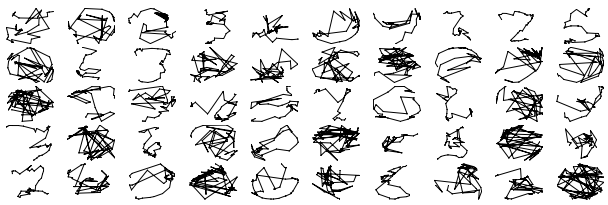
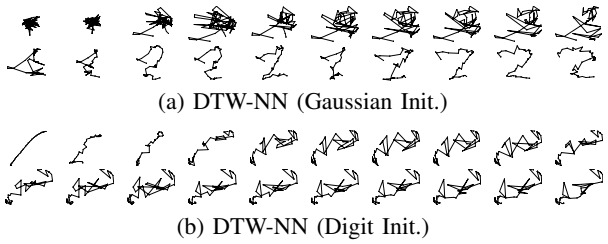


Figure 4. The state of the filters of DTW-NN (Gaussian Init.) after 35,000 rounds.



(a) DTW-NN (Gaussian Init.)

(b) DTW-NN (Digit Init.)

Figure 5. The progression of a sample filter from (a) the DTW-NN (Gaussian Init.) training and (b) the DTW-NN (Digit Init.) training. The top-left frame of each subfigure is the initial filter pattern and each subsequent frame from left to right, top to bottom is taken at 2000 round intervals.

Figure 6. The state of the filters of DTW-NN (Digit Init.) after 39,000 rounds.

The initial low accuracy of the Gaussian initialized filters is due to the randomized structures causing an equally large distance between the inputs and all the filters. Figure 4 shows that the through back propagation, eventually many of the Gaussian initialized filters gained structures resembling shapes and numbers. The delayed convergence is attributed to the need to *unravel* the patterns into useful structures with larger distinctions between the filters. Comparatively, when providing the filters with a starting structure, there is no need to unravel the temporal patterns. This can be seen by comparing Fig. 5 (a) and (b). It can be observed how quickly the DTW-NN (Digit Init.) converged compared to the DTW-NN (Gaussian Init.). The final filters of both evaluations are shown in Fig. 4 and Fig. 6.

C. Advantages and Disadvantages

DTW-NN excels at classification tasks, even with temporal patterns with low-density information element patterns and widely varying structures. It is able to generalize the temporal patterns based on the temporal features efficiently. Execution time is much faster than exhaustive searches. DTW-NN is suitable for accurate real-time on-line handwriting recognition.

The filters of the DTW-NN allow the network to overcome patterns of varying length. However, the size and shape of the initial filters must be considered in the design of the network. While there is no upper limit, the slope constraint given by Eq. (2) limits the input patterns from being less than half the size. However, alternate slope constraints can be used introducing different advantages and limitations. Also, when training, while the end accuracy is nearly

indistinguishable, it is suggested to use designed patterns than randomly generated patterns for faster training.

VII. Conclusion

This paper presented DTW-NN as a method of classifying isolated unnormalized temporal patterns. The proposed DTW-NN uses DTW as a kernel-like function to be used as a dissimilarity-based activation. This method offers the network the ability to learn temporal features by optimizing temporal filters through the back propagation through DTW. When classifying isolated on-line handwritten digits in multi-class classification with the UNIPEN data set, the experimental results show a high accuracy of 96.8% while having a very fast execution time. The results of the experiment show that DTW-NN can be an efficient method of implementing neural network models with variable size temporal patterns.

This work lays the foundation for using neural networks with temporal kernels. Future work can be done by expanding on this framework. For instance, by stacking multiple DTW-based temporal filter layers and creating a deeper network, it might be possible to learn higher level features. Also, additional improvements can be made on the DTW-NN such as adjusting the hyperparameters, changing the activations, or adding techniques such as dropout. The proposed DTW-NN is only the starting point for temporal kernel-based neural networks.

References

[1] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[2] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[3] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 12*. Association for Computing Machinery (ACM), 2012.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *INTERSPEECH*, 2010.

[6] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Institute of Electrical & Electronics Engineers (IEEE), may 2013.

[7] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, may 2009.

[8] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, 2009, pp. 545–552.

[9] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, 2002.

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

[11] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989.

[12] M. C. Mozer, "A focused back-propagation algorithm for temporal pattern recognition," *Complex systems*, vol. 3, no. 4, pp. 349–381, 1989.

[13] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[14] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten english character recognition based on convolutional neural network," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 125–129.

[15] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Web-Age Information Management*. Springer Science + Business Media, 2014, pp. 298–310.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[17] E. Pekalska and R. P. Duin, *The dissimilarity representation for pattern recognition: foundations and applications*. World Scientific, 2005.

[18] W. Chaovalitwongse and P. Pardalos, "On the time series support vector machine using dynamic time warping kernel for brain activity classification," *Cybernetics and Systems Analysis*, vol. 44, no. 1, pp. 125–138, 2008.

[19] H. S. K.-i. Noma, "Dynamic time-alignment kernel in support vector machine," 2002.

[20] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with support vector machines-a kernel approach," in *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*. IEEE, 2002, pp. 49–54.

[21] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[22] International Unipen Foundation, "Int. Unipen Foundation - iUF," http://www.unipen.org/home.html, accessed: 2015-10-19.

[23] J. Hu, S. G. Lim, and M. K. Brown, "Writer independent on-line handwriting recognition using an hmm approach," *Pattern Recognition*, vol. 33, no. 1, pp. 133–147, 2000.