Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks

Zhiguang Wang and **Tim Oates**
Computer Science and Electrical Engineering Department
University of Maryland Baltimore County
{stephen.wang, oates}@umbc.edu

Abstract

Inspired by recent successes of deep learning in computer vision and speech recognition, we propose a novel framework to encode time series data as different types of images, namely, Gramian Angular Fields (GAF) and Markov Transition Fields (MTF). This enables the use of techniques from computer vision for classification. Using a polar coordinate system, GAF images are represented as a Gramian matrix where each element is the trigonometric sum (i.e., superposition of directions) between different time intervals. MTF images represent the first order Markov transition probability along one dimension and temporal dependency along the other. We used Tiled Convolutional Neural Networks (tiled CNNs) on 12 standard datasets to learn high-level features from individual GAF, MTF, and GAF-MTF images that resulted from combining GAF and MTF representations into a single image. The classification results of our approach are competitive with five state-of-the-art approaches. An analysis of the features and weights learned via tiled CNNs explains why the approach works.

Introduction

We consider the problem of encoding time series data as images to allow machines to "visually" recognize and classify the time series. One type of time series recognition in speech and audio has been well studied. Researchers have achieved success using combinations of HMMs with acoustic models based on Gaussian Mixture models (GMMs) (Reynolds and Rose 1995; Leggetter and Woodland 1995). An alternative approach is to use a deep neural networks to produce the posterior probabilities over HMM states. Deep learning has become increasingly popular since the introduction of effective ways to train multiple hidden layers (Hinton, Osindero, and Teh 2006) and has been proposed as a replacement for GMMs to model acoustic data in speech recognition tasks (Mohamed, Dahl, and Hinton 2012). These Deep Neural Network - Hidden Markov Model hybrid systems (DNN-HMM) achieved remarkable performance in a variety of speech recognition tasks (Hinton et al. 2012; Deng, Hinton, and Kingsbury 2013; Deng et al. 2013). Such success stems from learning distributed representations via deeply layered structure and unsupervised pretraining by stacking single layer Restricted Boltzmann Machines (RBM).

Another deep learning architecture used in computer vision is convolutional neural networks (CNN) (LeCun et al. 1998). CNNs exploit translational invariance within their structures by extracting features through receptive fields (Hubel and Wiesel 1962) and learn with weight sharing, becoming the state-of-the-art approach in various image recognition and computer vision tasks (Lawrence et al. 1997; Krizhevsky, Sutskever, and Hinton 2012; LeCun, Kavukcuoglu, and Farabet 2010). Since unsupervised pretraining has been shown to improve performance (Erhan et al. 2010), sparse coding and Topographic Independent Component Analysis (TICA) are integrated as unsupervised pretraining approaches to learn more diverse features with complex invariances (Kavukcuoglu et al. 2010; Ngiam et al. 2010).

CNNs were proposed for speech processing to be invariant to shifts in time and frequency by LeCun and Bengio. Recently, CNNs have been shown to further improve hybrid model performance by applying convolution and max-pooling in the frequency domain on the TIMIT phone recognition task (Abdel-Hamid et al. 2012). A heterogeneous pooling approach proved to be beneficial for training acoustic invariance in (Deng, Abdel-Hamid, and Yu 2013). Further exploration with limited weight sharing and a weighted softmax pooling layer has been proposed to optimize CNN structures for speech recognition tasks (Abdel-Hamid, Deng, and Yu 2013).

Except for audio and speech data, relatively little work has explored feature learning in the context of typical time series analysis tasks with current deep learning architectures. (Zheng et al. 2014) explores supervised feature learning with CNNs to classify multi-channel time series with two datasets. They extracted subsequences with sliding windows and compared their results to Dynamic Time Warping (DTW) with a 1-Nearest-Neighbor classifier (1NN-DTW). Our motivation is to explore a novel framework to encode time series as images and thus to take advantage of the success of deep learning architectures in computer vision to learn features and identify structure in time series. Unlike speech recognition systems in which acoustic/speech

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

data input is typically represented by concatenating Mel-frequency cepstral coefficients (MFCCs) or perceptual linear predictive coefficient (PLPs) (Hermansky 1990), typical time series data are not likely to benefit from transformations typically applied to speech or acoustic data.

In this paper, we present two new representations for encoding time series as images that we call them Gramian Angular Field (GAF) and the Markov Transition Field (MTF). We select the same twelve "hard" time series dataset used by Oates et al., and applied deep Tiled Convolutional Neural Networks (Tiled CNN) with a pretraining stage that exploits local orthogonality by Topographic ICA (Ngiam et al. 2010) to "visually" represent the time series. We report our classification performance both on GAF and MTF separately, and GAF-MTF which resulted from combining GAF and MTF representations into a single image. By comparing our results with five previous and current state-of-the-art handcrafted representation and classification methods, we show that our approach in practice achieves competitive performance with the state of the art while exploring a relatively small parameter space. We also find that our Tiled CNN based deep learning method works well with small time series datasets, while the traditional CNN may not work well on such small datasets (Zheng et al. 2014). In addition to exploring the high level features learned by Tiled CNNs, we provide an in-depth analysis in terms of the duality between time series and images within our frameworks that more precisely identifies the reasons why our approaches work.

Encoding Time Series to Images

We first introduce our two frameworks for encoding time series as images. The first type of image is a Gramian Angular field (GAF), in which we represent time series in a polar coordinate system instead of the typical Cartesian coordinates. In the Gramian matrix, each element is actually the cosine of the summation of angles. Inspired by previous work on the duality between time series and complex networks (Campanharo et al. 2011), the main idea of the second framework, the Markov Transition Field (MTF), is to build the Markov matrix of quantile bins after discretization and encode the dynamic transition probability in a quasi-Gramian matrix.

Gramian Angular Field

Given a time series $X = \{x_1, x_2, ..., x_n\}$ of n real-valued observations, we rescale X so that all values fall in the interval $[-1, 1]$:

$$\tilde{x_i} = \frac{(x_i - max(X) + (x_i - min(X))}{max(X) - min(X)} \quad (1)$$

Thus we can represent the rescaled time series $\tilde{X}$ in polar coordinates by encoding the value as the angular cosine and time stamp as the radius with the equation below:

$$\begin{cases} \phi = \arccos(\tilde{x_i}), -1 \leq \tilde{x_i} \leq 1, \tilde{x_i} \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \quad (2)$$

In the equation above, t_i is the time stamp and N is a constant factor to regularize the span of the polar coordinate system. This polar coordinate based representation is a

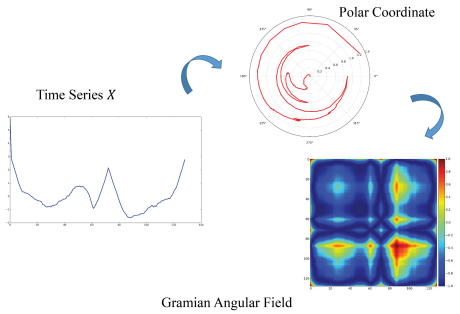

Figure 1: Illustration of the proposed encoding map of Gramian Angular Field. X is a sequence of typical time series in dataset 'SwedishLeaf'. After X is rescaled by eq. (1) and smoothed by PAA optionally, we transform it into polar coordinate system by eq. (2) and finally calculate its GAF image with eq. (4). In this example, we build GAF without PAA smoothing, so the GAF has a high resolution of 128×128.

novel way to understand time series. As time increases, corresponding values warp among different angular points on the spanning circles, like water rippling. The encoding map of equation 2 has two important properties. First, it is bijective as $\cos(\phi)$ is monotonic when $\phi \in [0, \pi]$. Given a time series, the proposed map produces one and only one result in the polar coordinate system with a unique inverse function. Second, as opposed to Cartesian coordinates, polar coordinates preserve absolute temporal relations. In Cartesian coordinates, the area is defined by $S_{i,j} = \int_{x(i)}^{x(j)} f(x(t))dx(t)$, we have $S_{i,i+k} = S_{j,j+k}$ if $f(x(t))$ has the same values on $[i, i+k]$ and $[j, j+k]$. However, in polar coordinates, if the area is defined as $S'_{i,j} = \int_{\phi(i)}^{\phi(j)} r[\phi(t)]^2 d(\phi(t))$, then $S'_{i,i+k} \neq S'_{j,j+k}$. That is, the corresponding area from time stamp i to time stamp j is not only dependent on the time interval $|i - j|$, but also determined by the absolute value of i and j. We will discuss this in more detail in another work.

After transforming the rescaled time series into the polar coordinate system, we can easily exploit the angular perspective by considering the trigonometric sum between each point to identify the temporal correlation within different time intervals. The GAF is defined as follows:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \quad (3)$$

$$= \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2} \quad (4)$$

I is the unit row vector $[1, 1, ..., 1]$. After transforming to the polar coordinate system, we take time series at each time step as a 1-D metric space. By defining the inner product $< x, y >= x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$, G is a Gramian matrix:

$$\begin{bmatrix} < \tilde{x_1}, \tilde{x_1} > & \cdots & < \tilde{x_1}, \tilde{x_n} > \\ < \tilde{x_2}, \tilde{x_1} > & \cdots & < \tilde{x_2}, \tilde{x_n} > \\ \vdots & \ddots & \vdots \\ < \tilde{x_n}, \tilde{x_1} > & \cdots & < \tilde{x_n}, \tilde{x_n} > \end{bmatrix} \tag{5}$$

The GAF has several advantages. First, it provides a way to preserve the temporal dependency, since time increases as the position moves from top-left to bottom-right. The GAF contains temporal correlations because $G_{(i,j||i-j|=k)}$ represents the relative correlation by superposition of directions with respect to time interval k. The main diagonal $G_{i,i}$ is the special case when $k = 0$, which contains the original value/angular information. With the main diagonal, we will approximately reconstruct the time series from the high level features learned by the deep neural network. However, the GAF is large because the size of Gramian matrix is $n \times n$ when the length of the raw time series is n. To reduce the size of the GAF, we apply Piecewise Aggregation Approximation (Keogh and Pazzani 2000) to smooth the time series and while keeping trends. The full procedure for generating the GAF is illustrated in Figure 1.

Markov Transition Field

We propose a framework similar to (Campanharo et al. 2011) for encoding dynamical transition statistics, but we extend that idea by representing the Markov transition probabilities sequentially to preserve information in the time domain.

Given a time series X, we identify its Q quantile bins and assign each x_i to the corresponding bins q_j ($j \in [1, Q]$). Thus we construct a $Q \times Q$ weighted adjacency matrix W by counting transitions among quantile bins in the manner of a first-order Markov chain along the time axis. $w_{i,j}$ is given by the frequency with which a point in the quantile q_j is followed by a point in the quantile q_i. After normalization by $\sum_j w_{ij} = 1$, W is the Markov transition matrix. It is insensitive to the distribution of X and temporal dependency on time steps t_i. However, getting rid of the temporal dependency results in too much information loss in matrix W. To overcome this drawback, we define the Markov Transition Field (MTF) as follows:

$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix} \tag{6}$$

We build a $Q \times Q$ Markov transition matrix (say, W) by dividing the data (magnitude) into Q quantile bins. The quantile bins that contain the data at time stamp i and j (temporal axis) are q_i and q_j ($q \in [1, Q]$). M_{ij} in MTF denotes the transition probability of $q_i \rightarrow q_j$. That is, we spread out matrix W which contains the transition probability on the magnitude axis into the MTF matrix by considering the temporal positions.

By assigning the probability from the quantile at time step i to the quantile at time step j at each pixel M_{ij}, the MTF M actually encodes the multi-span transition probabilities

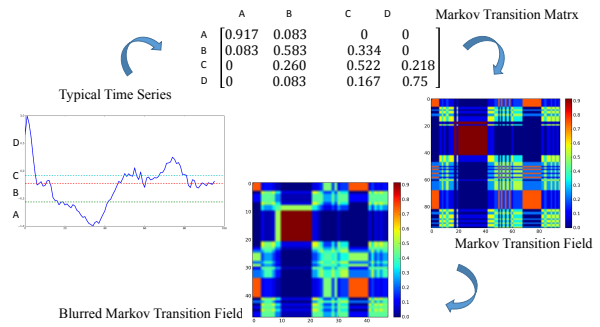

Figure 2: Illustration of the proposed encoding map of Markov Transition Field. X is a sequence of typical time series in dataset 'ECG'. X is firstly discretized into Q quantile bins. Then we calculate its Markov Transition Matrix W and finally build its MTF with eq. (6). In addition, we reduce the image size from 96×96 to 48×48 by averaging the pixels in each non-overlapping 2×2 patch.

of the time series. $M_{i,j||i-j|=k}$ denotes the transition probability between the points with time interval k. For example, $M_{ij|j-i=1}$ illustrates the transition process along the time axis with a skip step. The main diagonal M_{ii}, which is a special case when $k = 0$ captures the probability from each quantile to itself (the self-transition probability) at time step i. To make the image size manageable and computation more efficient, we reduce the MTF size by averaging the pixels in each non-overlapping $m \times m$ patch with the blurring kernel $\{\frac{1}{m^2}\}_{m \times m}$. That is, we aggregate the transition probabilities in each subsequence of length m together. Figure 2 shows the procedure to encode time series to MTF.

Tiled Convolutional Neural Networks

Tiled Convolutional Neural Networks (Ngiam et al. 2010) are a variation of Convolutional Neural Networks that use tiles and multiple feature maps to learn invariant features. Tiles are parameterized by a tile size k to control the distance over which weights are shared. By producing multiple feature maps, Tiled CNNs learn overcomplete representations through unsupervised pretraining with Topographic ICA (TICA).

A typical TICA network is actually a double-stage optimization procedure with squares and square root nonlinearities in each stage, respectively. In the first stage, the weight matrix W is learned while the matrix V is hard-coded to represent the topographic structure of units. More precisely, given a sequence of inputs $\{x^h\}$, the activation of each unit in the second stage is $f_i(x^{(h)}; W, V) = \sqrt{\sum_{k=1}^{p} V_{ik} (\sum_{j=1}^{q} W_{kj} x_j^{(h)})^2}$. TICA learns the weight matrix W in the second stage by solving the following:

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \sum_{h=1}^{n} \sum_{i=1}^{p} f_i(x^{(h)}; W, V) \\ \text{subject to} \quad & WW^T = I \end{aligned} \tag{7}$$

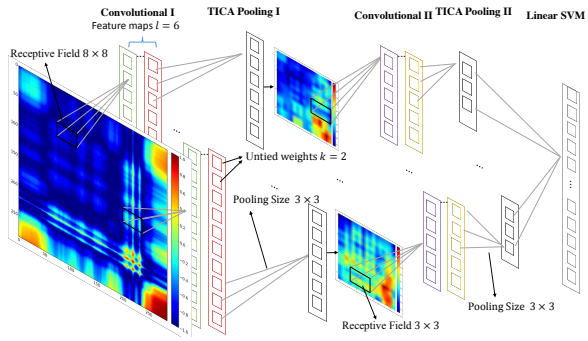

Figure 3: Structure of the tiled convolutional neural network. We fix the size of receptive field to 8×8 in the first convolutional layer and 3×3 in the second convolutional layer. Each TICA pooling layer pools over a block of 3×3 input units in the previous layer without wraparound the boarders to optimize for sparsity of the pooling units. The number of pooling units in each map is exactly the same as the number of input units. The last layer is a linear SVM for classification. We construct this network by stacking two Tiled CNNs, each with 6 maps ($l = 6$) and tiling size $k = 2$.

Above, $W \in \mathbb{R}^{p \times q}$ and $V \in \mathbb{R}^{p \times p}$ where p is the number of hidden units in a layer and q is the size of the input. V is a logical matrix ($V_{ij} = 1$ or 0) that encodes the topographic structure of the hidden units by a contiguous 3×3 block. The orthogonality constraint $WW^T = I$ provides diversity among learned features.

Neither GAF nor MTF images are natural images; they have no natural concepts such as "edges" and "angles". Thus, we propose to exploit the benefits of unsupervised pretraining with TICA to learn many diverse features with local orthogonality. In addition, Ngiam et al. empirically demonstrate that tiled CNNs perform well with limited labeled data because the partial weight tying requires fewer parameters and reduces the need for a large amount of labeled data. Our data from the UCR Time Series Repository (Keogh et al. 2011) tends to have few instances (e.g., the "yoga" dataset has 300 labeled instance in the training set and 3000 unlabeled instance in the test set), tiled CNNs are suitable for our learning task.

Typically, tiled CNNs are trained with two hyperparameters, the tiling size k and the number of feature maps l. In our experiments, we directly fixed the network structures without tuning these hyperparameters in loops for several reasons. First, our goal is to explore the expressive power of the high level features learned from GAF and MTF images. We have already achieved competitive results with the default deep network structures that Ngiam et al. used for image classification on the NORB image classification benchmark. Although tuning the parameters will surely enhance performance, doing so may cloud our understanding of the power of the representation. Another consideration is computational efficiency. All of the experiments on the 12 "hard" datasets could be done in one day on a laptop with an Intel i7-3630QM CPU and 8GB of memory (our experimental platform). Thus, the results in this paper are a preliminary lower bound on the potential best performance. Thoroughly exploring the deep network structures and parameters will be addressed in future work. The structure and parameters of the tiled CNN used in this paper are illustrated in Figure 3.

Table 1: Tiled CNN error rate on training set and test set

DATASET	GAF		MTF	
	TRAIN	TEST	TRAIN	TEST
50words	0.338	0.310	0.442	0.426
adiac	0.321	0.284	0.638	0.665
beef	0.633	0.4	0.533	0.233
coffee	0	0	0	0
ECG200	0.16	0.11	0.15	0.21
faceall	0.121	0.244	0.102	0.259
lighting2	0.2	0.18	0.167	0.361
lighting7	0.329	0.397	0.386	0.411
oliveoil	0.2	0.2	0.033	0.3
OSULeaf	0.415	0.463	0.43	0.483
SwedishLeaf	0.134	0.104	0.206	0.176
yoga	0.183	0.177	0.193	0.243

Classifying Time Series Using GAF/MTF

We apply Tiled CNNs to classify using GAF and MTF representation on twelve tough datasets, on which the classification error rate is above 0.1 with the state-of-the-art SAX-BoP approach (Lin, Khade, and Li 2012; Oates et al. 2012). More detailed statistics are summarized in Table 2. The datasets are pre-split into training and testing sets for experimental comparisons. For each dataset, the table gives its name, the number of classes, the number of training and test instances, and the length of the individual time series.

Experimental Setting

In our experiments, the size of the GAF image is regulated by the the number of PAA bins S_{GAF}. Given a time series X of size n, we divide the time series into S_{GAF} adjacent, non-overlapping windows along the time axis and extract the means of each bin. This enables us to construct the smaller GAF matrix $G_{S_{GAF} \times S_{GAF}}$. MTF requires the time series to be discretized into Q quantile bins to calculate the $Q \times Q$ Markov transition matrix, from which we construct the raw MTF image $M_{n \times n}$ afterwards. Before classification, we shrink the MTF image size to $S_{MTF} \times S_{MTF}$ by the blurring kernel $\{\frac{1}{m^2}\}_{m \times m}$ where $m = \lceil \frac{n}{S_{MTF}} \rceil$. The Tiled CNN is trained with image size $\{S_{GAF}, S_{MTF}\} \in \{16, 24, 32, 40, 48\}$ and quantile size $Q \in \{8, 16, 32, 64\}$. At the last layer of the Tiled CNN, we use a linear soft margin SVM (Fan et al. 2008) and select C by 5-fold cross validation over $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ on the training set.

For each input of image size S_{GAF} or S_{MTF} and quantile size Q, we pretrain the Tiled CNN with the full unlabeled dataset (both training and test set) to learn the initial weights W through TICA. Then we train the SVM at the last layer by selecting the penalty factor C with cross validation.

Table 2: Summary statistics of standard dataset and comparative results

DATASET	CLASS	TRAIN	TEST	LENGTH	1NN-EUCLIDEAN	1NN-DTW	FAST SHAPELET	BOP	SAX-VSM	GAF-MTF
50words	50	450	455	270	0.369	**0.242**	0.4429	0.466	N/A	0.284
Adiac	37	390	391	176	0.389	0.391	0.514	0.432	0.381	**0.307**
Beef	5	30	30	470	0.467	0.467	0.447	0.433	**0.033**	0.3
Coffee	2	28	28	286	0.25	0.18	0.067	0.036	**0**	**0**
ECG200	2	100	100	96	0.12	0.23	0.227	0.14	0.14	**0.08**
FaceAll	14	560	1,690	131	0.286	**0.192**	0.402	0.219	0.207	0.223
Lightning2	2	60	61	637	0.246	**0.131**	0.295	0.164	0.196	0.18
Lightning7	7	70	73	319	0.425	**0.274**	0.403	0.466	0.301	0.397
OliveOil	4	30	30	570	0.133	0.133	0.213	0.133	**0.1**	0.167
OSULeaf	6	200	242	427	0.483	0.409	0.359	0.236	**0.107**	0.446
SwedishLeaf	15	500	625	128	0.213	0.21	0.27	0.198	0.251	**0.093**
Yoga	2	300	3,000	426	0.17	0.164	0.249	0.17	0.164	**0.16**

Finally, we classify the test set using the optimal hyperparameters $\{S, Q, C\}$ with the lowest error rate on the training set. If two or more models tie, we prefer the larger S and Q because larger S helps preserve more information through the PAA procedure and larger Q encodes the dynamic transition statistics with more detail. Our model selection approach provides generalization without being overly expensive computationally.

Results and Discussion

We use Tiled CNNs to classify GAF and MTF representations separately on the 12 datasets. The training and test error rates are shown in Table 1. Generally, our approach is not prone to overfitting as seen by the relatively small difference between training and test set errors. One exception is the Olive Oil dataset with the MTF approach where the test error is significantly higher.

In addition to the risk of potential overfitting, MTF has generally higher error rates than GAF. This is most likely because of uncertainty in the inverse image of MTF. Note that the encoding function from time series to GAF and MTF are both surjective. The map functions of GAF and MTF will each produce only one image with fixed S and Q for each given time series X . Because they are both surjective mapping functions, the inverse image of both mapping functions is not fixed. As shown in a later section, we can approximately reconstruct the raw time series from GAF, but it is very hard to even roughly recover the signal from MTF. GAF has smaller uncertainty in the inverse image of its mapping function because such randomness only comes from the ambiguity of $\cos(\phi)$ when $\phi \in [0, 2\pi]$. MTF, on the other hand, has a much larger inverse image space, which results in large variation when we try to recover the signal. Although MTF encodes the transition dynamics which are important features of time series, such features seem not to be sufficient for recognition/classification tasks.

Note that at each pixel, G_{ij}, denotes the superstition of the directions at t_i and t_j, M_{ij} is the transition probability from quantile at t_i to quantile at t_j. GAF encodes static information while MTF depicts information about dynamics. From this point of view, we consider them as two "orthogonal" channels, like different colors in the RGB image space. Thus, we can combine GAF and MTF images of the same size (i.e. $S_{GAF} = S_{MTF}$) to construct a double-channel image (GAF-MTF). Since GAF-MTF combines both the static and dynamic statistics embedded in raw time series, we posit that it will be able to enhance classification performance. In the next experiment, we pretrain and train the Tiled CNN on the compound GAF-MTF images. Then, we report the classification error rate on test sets.

Table 2 compares the classification error rate of our approach with previously published performance results of five competing methods: two state-of-the-art 1NN classifiers based on Euclidean distance and DTW, the recently proposed Fast-Shapelets based classifier (Rakthanmanon and Keogh 2013), the classifier based on Bag-of-Patterns (BoP) (Lin, Khade, and Li 2012; Oates et al. 2012) and the most recent SAX-VSM approach (Senin and Malinchik 2013). Our approach outperforms 1NN-Euclidean, fast-shapelets, and BoP, and is competitive with 1NN-DTW and SAX-VSM.

In addition, by comparing the results between Table 2 and Table 1, we verified our assumption that combined GAF-MTF images have better expressive power than GAF or MTF alone for classification. GAF-MTF achieves the lower test error rate on ten datasets out of twelve (except for the dataset Adiac and Beef). On the Olive Oil dataset, the training error rate is 6.67% and the test error rate is 16.67%. This demonstrates that the integration of both types of images into one compound image decreases the risk of overfitting as well as enhancing the overall classification accuracy.

Analysis on Features and Weights Learned through Tiled CNNs

In contrast to the cases in which the CNN is applied in natural image recognition tasks, neither GAF nor MTF has natural interpretations of visual concepts like "edges" or "angles". In this section we analyze the features and weights learned through Tiled CNNs to explain why our approach works.

As mentioned earlier, the mapping function from time series to GAF is surjective and the uncertainty in its inverse image comes from the ambiguity of the $\cos(\phi)$ when $\phi \in$

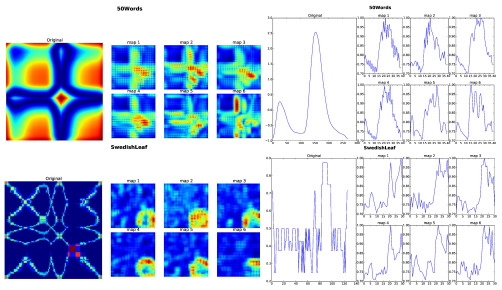

Figure 4: (a) Original GAF and its six learned feature maps before the SVM layer in Tiled CNN (top left), and (b) raw time series and approximate reconstructions based on the main diagonal of six feature maps (top right) on '50Words' dataset; (c) Original MTF and its six learned feature maps before the SVM layer in Tiled CNN (bottom left), and (d) curve of self-transition probability along time axis (main diagonal of MTF) and approximate reconstructions based on the main diagonal of six feature maps (bottom right) on "SwedishLeaf" dataset.

$[0, 2\pi]$. The main diagonal of GAF, i.e. $\{G_{ii}\} = \{\cos(2\phi_i)\}$ allows us to approximately reconstruct the original time series, ignoring the signs by

$$\cos(\phi) = \sqrt{\frac{\cos(2\phi) + 1}{2}} \tag{8}$$

MTF has much larger uncertainty in its inverse image, making it hard to reconstruct the raw data from MTF alone. However, the diagonal $\{M_{ij||i-j|=k}\}$ represents the transition probability among the quantiles in temporal order considering the time interval k. We construct the self-transition probability along the time axis from the main diagonal of MTF like we do for GAF. Although such reconstructions less accurately capture the morphology of the raw time series, they provide another perspective of how Tiled CNNs capture the transition dynamics embedded in MTF.

Figure 4 illustrates the reconstruction results from six feature maps learned before the last SVM layer on GAF and MTF. The Tiled CNN extracts the color patch, which is essentially a moving average that enhances several receptive fields within the nonlinear units by different trained weights. It is not a simple moving average but the synthetic integration by considering the 2D temporal dependencies among different time intervals, which is a benefit from the Gramian matrix structure that helps preserve the temporal information. By observing the rough orthogonal reconstruction from each layer of the feature maps, we can clearly observe that the tiled CNN can extract the multi-frequency dependencies through the convolution and pooling architecture on the GAF and MTF images to preserve the trend while addressing more details in different subphases. As shown in Figure 4(b) and 4(d), the high-leveled feature maps learned by the Tiled CNN are equivalent to a multi-frequency approximator of the original curve.

Figure 5 demonstrates the learned sparse weight matrix W with the constraint $WW^T = I$, which makes effective

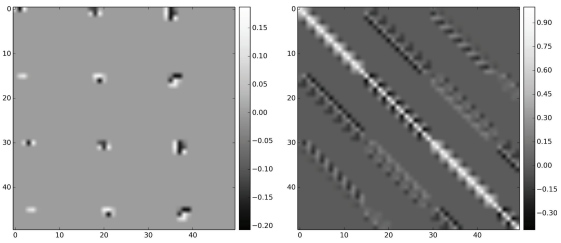

Figure 5: learned sparse weights W for the last SVM layer in Tiled CNN (left) and its orthogonality constraint by $WW^T = I$ (right).

use of local orthogonality. The TICA pretraining provides the built-in advantage that the function w.r.t the parameter space is not likely to be ill-conditioned as $WW^T = 1$. As shown in Figure 5 (right), the weight matrix W is quasi-orthogonal and approaching 0 without very large magnitude. This implies that the condition number of W approaches 1 helps the system to be well-conditioned.

Conclusions and Future Work

We created a pipeline for converting time series data into novel representations, GAF and MTF images, and extracted high-level features from these using Tiled CNN. The features were subsequently used for classification. We demonstrated that our approach yields competitive results when compared to state-of-the-art methods when searching a relatively small parameter space. We found that GAF-MTF multi-channel images are scalable to larger numbers of quasi-orthogonal features that yield more comprehensive images. Our analysis of high-level features learned from Tiled CNN suggested that Tiled CNN works like a multi-frequency moving average that benefits from the 2D temporal dependency that is preserved by Gramian matrix.

Important future work will involve applying our method to massive amounts of data and searching in a more complete parameter space to solve the real world problems. We are also quite interested in how different deep learning architectures perform on the GAF and MTF images. Another interesting future work is to model time series through GAF and MTF images. We aim to apply learned time series models in regression/imputation and anomaly detection tasks. To extend our methods to the streaming data, we suppose to design the online learning approach with recurrent network structures.

References

Abdel-Hamid, O.; Mohamed, A.-r.; Jiang, H.; and Penn, G. 2012. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 4277–4280. IEEE.

Abdel-Hamid, O.; Deng, L.; and Yu, D. 2013. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *INTERSPEECH*, 3366–3370.

Campanharo, A. S.; Sirer, M. I.; Malmgren, R. D.; Ramos, F. M.; and Amaral, L. A. N. 2011. Duality between time series and networks. *PloS one* 6(8):e23378.

Deng, L.; Abdel-Hamid, O.; and Yu, D. 2013. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 6669–6673. IEEE.

Deng, L.; Li, J.; Huang, J.-T.; Yao, K.; Yu, D.; Seide, F.; Seltzer, M.; Zweig, G.; He, X.; Williams, J.; et al. 2013. Recent advances in deep learning for speech research at microsoft. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 8604–8608. IEEE.

Deng, L.; Hinton, G.; and Kingsbury, B. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 8599–8603. IEEE.

Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.-A.; Vincent, P.; and Bengio, S. 2010. Why does unsupervised pretraining help deep learning? *The Journal of Machine Learning Research* 11:625–660.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* 9:1871–1874.

Hermansky, H. 1990. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America* 87(4):1738–1752.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE* 29(6):82–97.

Hinton, G.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7):1527–1554.

Hubel, D. H., and Wiesel, T. N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology* 160(1):106.

Kavukcuoglu, K.; Sermanet, P.; Boureau, Y.-L.; Gregor, K.; Mathieu, M.; and Cun, Y. L. 2010. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, 1090–1098.

Keogh, E. J., and Pazzani, M. J. 2000. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 285–289. ACM.

Keogh, E.; Xi, X.; Wei, L.; and Ratanamahatana, C. A. 2011. The ucr time series classification/clustering homepage. *URL= http://www. cs. ucr. edu/˜ eamonn/time_series_data*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Lawrence, S.; Giles, C. L.; Tsoi, A. C.; and Back, A. D. 1997. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE Transactions on* 8(1):98–113.

LeCun, Y., and Bengio, Y. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

LeCun, Y.; Kavukcuoglu, K.; and Farabet, C. 2010. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 253–256. IEEE.

Leggetter, C. J., and Woodland, P. C. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language* 9(2):171–185.

Lin, J.; Khade, R.; and Li, Y. 2012. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems* 39(2):287–315.

Mohamed, A.-r.; Dahl, G. E.; and Hinton, G. 2012. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on* 20(1):14–22.

Ngiam, J.; Chen, Z.; Chia, D.; Koh, P. W.; Le, Q. V.; and Ng, A. Y. 2010. Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1279–1287.

Oates, T.; Mackenzie, C. F.; Stein, D. M.; Stansbury, L. G.; Dubose, J.; Aarabi, B.; and Hu, P. F. 2012. Exploiting representational diversity for time series classification. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, 538–544. IEEE.

Rakthanmanon, T., and Keogh, E. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the thirteenth SIAM conference on data mining (SDM)*. SIAM.

Reynolds, D. A., and Rose, R. C. 1995. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on* 3(1):72–83.

Senin, P., and Malinchik, S. 2013. Sax-vsm: Interpretable time series classification using sax and vector space model. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, 1175–1180. IEEE.

Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; and Zhao, J. L. 2014. Time series classification using multi-channels deep convolutional neural networks. In *Web-Age Information Management*. Springer. 298–310.