ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

TimeNet: Pre-trained deep recurrent neural network for time series classification

Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, Gautam Shroff

TCS Research, New Delhi, India
{malhotra.pankaj, vishnu.tv, lovekesh.vig, puneet.a, gautam.shroff}@tcs.com

Abstract. Inspired by the tremendous success of deep Convolutional Neural Networks as generic feature extractors for images, we propose *TimeNet*: a deep recurrent neural network (RNN) trained on diverse time series in an unsupervised manner using sequence to sequence (seq2seq) models to extract features from time series. Rather than relying on data from the problem domain, TimeNet attempts to generalize time series representation across domains by ingesting time series from several domains simultaneously. Once trained, TimeNet can be used as a *generic off-the-shelf feature extractor for time series*. The representations or embeddings given by a pre-trained TimeNet are found to be useful for time series classification (TSC). For several publicly available datasets from UCR TSC Archive and an industrial telematics sensor data from vehicles, we observe that a classifier learned over the TimeNet embeddings yields significantly better performance compared to (i) a classifier learned over the embeddings given by a domain-specific RNN, as well as (ii) a nearest neighbor classifier based on Dynamic Time Warping.

1 Introduction

Recently, fixed-dimensional vector representations for sequences of words in the form of sentences, paragraphs, and documents have been successfully used for natural language processing tasks such as machine translation and sentiment analysis [1]. Noticeably, deep Convolutional Neural Networks (CNNs) trained on millions of images from 1000 object classes have been used as off-the-shelf feature extractors to yield powerful generic image descriptors for a diverse range of tasks such as image classification, scene recognition and image retrieval [2]. These features or representations have even been shown to outperform models heavily tuned for the specific tasks.

For time series, deep recurrent neural networks (RNNs) have been shown to perform hierarchical processing with different layers tackling different time scales [3, 4]. However, it is well known that deep learning models are data-intensive, and getting access to labeled training data is expensive. With the advent of "Industrial Internet", unlabeled time series data from sensors is available in abundance, and leveraging it for training deep RNNs in an unsupervised manner can be useful. Recently, sequence to sequence (*seq2seq*) models [5] based on RNNs have been used in an unsupervised manner for time series modeling tasks such as audio segment representation [6], anomaly detection [7], and determining machine-health from sensor data [8]. We leverage a seq2seq model trained simultaneously on a large number of diverse time series from multiple

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

domains in an unsupervised manner to obtain a multilayered RNN, which we call *TimeNet* (refer Section 2 for details). We show that TimeNet yields vector representations or embeddings that capture important characteristics from time series. For several time series classification (TSC) tasks, TimeNet based embeddings perform better at classification compared to: i) embeddings obtained from a deep RNN trained specifically for the task domain using seq2seq, ii) Dynamic Time Warping (DTW) based nearest neighbor classifier (DTW-C). Also, we observe that TimeNet based embeddings for time series belonging to different classes form well-separated clusters when visualized using t-SNE [9].

Related Work: It has been shown that it is possible to have a generic deep neural network for raw audio waveforms that could capture different characteristics of many different speakers with equal fidelty [10]. Audio Word2Vec model [6] demonstrated usefulness of learning fixed-dimensional representations from varying length audio signals using seq2seq models. Our work extends such approaches from speech domain and deep CNNs based approaches from image domain [2], and shows that it is possible to learn a generic model for time series across diverse domains. Data augmentation techniques (e.g. [11], [12]) have been proposed to handle data scarcity for training deep models for sequences or time series. ODEs as a generative model for time series have been shown to improve performance of RNNs for anomaly detection [11]. Data augmentation through window slicing and warping is used to handle scarcity of data for training CNNs for TSC in [12]. On the other hand, our approach uses time series data from several domains simultaneously to train a deep RNN based time series model that can then be used as a pre-trained model to obtain representations for time series, in effect overcoming the need for large amounts of training data for the problem domain. To the best of our knowledge, our work is the first to show that it is possible to leverage unlabeled varying length time series from diverse domains to obtain a multilayered RNN as a generic time series feature extractor.

The rest of the paper is organized as follows: Section 2 describes our approach to train TimeNet using seq2seq models. Section 3 provides details of experimental evaluation of TimeNet and comparison with domain-specific RNN encoders. Section 4 offers concluding remarks.

2 Learning TimeNet using sequence-to-sequence models

We briefly introduce multilayered RNNs with dropout based regularization, and then describe our approach to learn TimeNet using seq2seq models consisting of a pair of multilayered RNNs trained together: an encoder RNN and a decoder RNN (refer Fig. 1).

Multilayered RNN with Dropout: We consider a multilayered RNN with Gated Recurrent Units [5] in the hidden layers where dropout is used for regularization [13]. Dropout is applied to non-recurrent connections ensuring that the state of any hidden unit is not affected ensuring information flow across time-steps. For lth hidden layer of a multilayered RNN with L hidden layers, the hidden state $\mathbf{h}_t^l$ at time t is obtained from the previous hidden state $\mathbf{h}_{t-1}^l$ and

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

the hidden state $\mathbf{h}_t^{l-1}$. The time series goes through following transformations iteratively for $t = 1$ through T, where T is the time series length:

$$reset\,gate : \mathbf{r}_t^l = \sigma(\mathbf{W}_r^l.[\mathbf{D}(\mathbf{h}_t^{l-1}), \mathbf{h}_{t-1}^l]) \quad (1)$$

$$update\,gate : \mathbf{u}_t^l = \sigma(\mathbf{W}_u^l.[\mathbf{D}(\mathbf{h}_t^{l-1}), \mathbf{h}_{t-1}^l]) \quad (2)$$

$$proposed\,state : \tilde{\mathbf{h}}_t^l = \tanh(\mathbf{W}_p^l.[\mathbf{D}(\mathbf{h}_t^{l-1}), \mathbf{r}_t \odot \mathbf{h}_{t-1}^l]) \quad (3)$$

$$hidden\,state : \mathbf{h}_t^l = (1 - \mathbf{u}_t^l) \odot \mathbf{h}_{t-1}^l + \mathbf{u}_t^l \odot \tilde{\mathbf{h}}_t^l \quad (4)$$

where $\odot$ is Hadamard product, $[\mathbf{a}, \mathbf{b}]$ is concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{D}(.)$ is dropout operator that randomly sets the dimensions of its argument to zero with probability equal to dropout rate, $\mathbf{h}_t^0$ is the input z_t at time-step t. $\mathbf{W}_r$, $\mathbf{W}_u$, and $\mathbf{W}_p$ are weight matrices of appropriate dimensions s.t. $\mathbf{r}_t^l, \mathbf{u}_t^l, \tilde{\mathbf{h}}_t^l$, and $\mathbf{h}_t^l$ are vectors in $\mathbb{R}^{c^l}$, where c^l is the number of units in layer l. The sigmoid (σ) and *tanh* activation functions are applied element-wise.

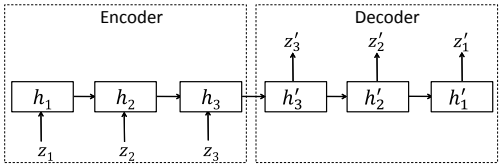


Fig. 1: Sequence-to-sequence auto-encoder for sample time series $\{z_1, z_2, z_3\}$.

We train a seq2seq model on time series of varying length from diverse domains, and once trained, freeze the encoder RNN to be used as TimeNet. The final hidden state of the encoder after processing a time series is used as the embedding for the time series. More specifically, given a time series $Z^{(i)} = \{z_1^{(i)}, z_2^{(i)}, ..., z_{T^{(i)}}^{(i)}\}$ of length $T^{(i)}$, $\mathbf{h}_t^{(i)}$ is the hidden state of the encoder at time t, where $\mathbf{h}_t^{(i)} \in \mathbb{R}^c$. The number of hidden units in the encoder is $c = \sum_{l=1}^{L} c^l$. The encoder captures information relevant to reconstruct the time series as it encodes the time series, and when it reaches the last point in the time series, the hidden state $\mathbf{h}_{T^{(i)}}^{(i)}$ is the vector representation or embedding for time series $Z^{(i)}$. The decoder has the same network structure as the encoder, with the final hidden state $\mathbf{h}_{T^{(i)}}^{(i)}$ of the encoder being used as the initial hidden state of decoder. The decoder additionally has a linear layer as the output layer. The decoder reconstructs the time series in reverse order, i.e., the target time series is $\{z_{T^{(i)}}^{(i)}, z_{T^{(i)}-1}^{(i)}, ..., z_1^{(i)}\}$. The encoder-decoder pair is trained in an unsupervised manner as a sequence auto-encoder (SAE) to reconstruct the input time series so as to minimize the objective $E = \sum_{i=1}^{N} \sum_{t=1}^{T^{(i)}} (z_t^{(i)} - z_t'^{(i)})^2$, where $z_t'^{(i)}$ is the reconstructed value corresponding to $z_t^{(i)}$, N is the number of time series.

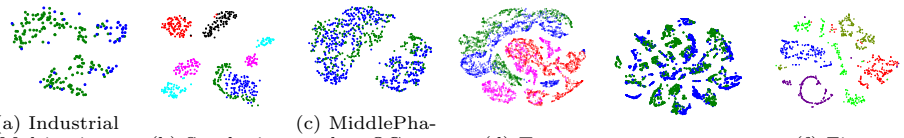
Unlike the conventional approach of feeding an input to the decoder at each time step during training and inference [5], the only inputs the decoder gets are the embedding for the time series (final hidden state of encoder), and the steps T for which the decoder has to be iterated in order to reconstruct the input. We observe that the embedding or the final hidden state of the encoder thus obtained carries all the relevant information necessary to represent a time series.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

3 Experimental Evaluation

We begin with the training details of TimeNet and domain-specific SAEs, and then present a qualitative analysis of the TimeNet embeddings using t-SNE. To test the robustness of TimeNet as a generic feature extractor, we compare pre-trained TimeNet based embeddings with domain-specific SAE based embeddings on several diverse TSC datasets which were not used for training the TimeNet.

Experimental setup and training details: We chose 18 datasets for training, 6 datasets for validation, and another 30 as test datasets for TimeNet evaluation from the UCR TSC Archive [14]. Each dataset comes with a pre-defined train-test split and a class label for each time series. The training dataset contains over 10K diverse univariate time series belonging to 151 different classes from the 18 datasets with time series length T varying from 24 to 512. We also evaluate TimeNet on an industrial telematics dataset consisting of hourly readings from six sensors installed on engines in vehicles, where the task is to classify normal and abnormal behavior of engines (referred as Industrial Multivariate in Fig. 2a and Table 1). We concatenate embeddings for time series of each sensor to obtain the final embedding for each multivariate time series. We use Adam optimizer for training with learning rate 0.006, batch size 32, and dropout rate 0.4. The architecture with minimum average reconstruction error on the time series in the validation set is used as final model. The best TimeNet model obtained has $c^l = 60$ units in each hidden layer with $L = 3$ such that the embedding dimension $c = 180$. The domain-specific SAE models are trained under same parameter settings using the training set of the respective dataset while tuning for c^l and L.



(a) Industrial Multivariate (C=2) (b) Synthetic Control (C=6) (c) MiddlePhalanxOC (C=2) (d) Two Patterns (C=4) (e) Yoga (C=2) (f) Five datasets

Fig. 2: Sample TimeNet embeddings visualized using t-SNE. C denotes number of classes of time series in the dataset. (Image best viewed magnified.)

Visualization of embeddings using t-SNE: Fig. 2a and Figs. 2b - 2e show t-SNE visualizations of TimeNet embeddings on the industrial multivariate engine dataset and sample UCR datasets, respectively. Each color represents a different class of time series. Fig. 2f shows t-SNE plot for randomly selected time series across datasets with each color representing a different dataset. We observe that embeddings for time series belonging to different classes within a dataset form well-separated clusters for most datasets. Also, the embeddings for time series from one dataset are well-separated from those of other datasets. These observations suggest that *TimeNet embeddings capture important characteristics of the time series.*

Embeddings for TSC: For each test dataset, we compare two SVM clas-

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

sifiers with radial basis function (RBF) kernel: i) TN-C: using TimeNet embeddings as features, ii) SAE-C: using embeddings obtained from the encoder of a domain-specific SAE model (seq2seq auto-encoder trained on the respective dataset) as features. The classification error rates for TN-C, SAE-C, and DTW-C are reported in Table 1. We observe that TN-C exceeds or matches

Dataset	**T**	**DTW-C [15]**	**SAE-C**	**TN-C**	**TN-$C_{2/3}$**	**TN-C_{L1}**	**TN-C_{L2}**	**TN-C_{L3}**
Industrial Multivariate	30	-	0.221	**0.173**	0.176	0.135	0.154	0.154
Synthetic Control	60	0.017	0.017	**0.013**	0.016	0.010	0.013	0.027
PhalangesOC	80	0.239	0.228	**0.207**	0.225	0.213	0.221	0.217
DistalPhalanxOAG	80	0.228	**0.160**	0.223	0.211	0.178	0.200	0.165
DistalPhalanxOC	80	0.232	**0.187**	0.188	0.201	0.188	0.178	0.185
DistalPhalanxTW	80	0.272	0.243	**0.208**	0.220	0.203	0.213	0.223
MiddlePhalanxOAG	80	0.253	0.348	**0.210**	0.229	0.215	0.280	0.205
MiddlePhalanxOC	80	0.318	0.307	**0.270**	0.344	0.475	0.472	0.295
MiddlePhalanxTW	80	0.419	0.381	**0.363**	0.392	0.361	0.371	0.366
ProximalPhalanxOAG	80	0.215	**0.137**	0.146	0.154	0.141	0.151	0.156
ProximalPhalanxOC	80	0.210	0.179	**0.175**	0.199	0.175	0.175	0.175
ProximalPhalanxTW	80	0.263	**0.188**	0.195	0.194	0.200	0.200	0.188
ElectricDevices	96	0.376	0.335	**0.267**	0.288	0.265	0.280	0.309
MedicalImages	99	0.253	**0.247**	0.250	0.271	0.238	0.246	0.232
Swedish Leaf	128	0.157	**0.099**	0.102	0.139	0.123	0.126	0.115
Two Patterns	128	0.002	0.001	**0.000**	0.002	0.000	0.002	0.007
ECG5000	140	0.075	**0.066**	0.069	0.069	0.063	0.069	0.066
ECGFiveDays	136	0.203	**0.063**	0.074	0.150	0.129	0.127	0.096
Wafer	152	**0.005**	0.006	**0.005**	0.007	0.008	0.006	0.009
ChlorineConcentration	166	0.35	0.277	**0.269**	0.344	0.227	0.250	0.314
Adiac	176	0.391	0.435	**0.322**	0.372	0.366	0.304	0.294
Strawberry	235	**0.062**	0.070	**0.062**	0.075	0.090	0.072	0.077
Cricket_X	300	**0.236**	0.341	0.300	0.321	0.346	0.326	0.364
Cricket_Y	300	**0.197**	0.397	0.338	0.363	0.379	0.351	0.400
Cricket_Z	300	**0.180**	0.305	0.308	0.336	0.328	0.338	0.359
uWaveGestureLib_X	315	0.227	**0.211**	0.214	0.228	0.219	0.216	0.220
uWaveGestureLib_Y	315	0.301	**0.291**	0.311	0.326	0.304	0.307	0.335
uWaveGestureLib_Z	315	0.322	**0.280**	0.281	0.295	0.298	0.289	0.286
Yoga	426	**0.155**	0.174	0.160	0.200	0.176	0.152	0.173
FordA	500	0.341	0.284	**0.219**	0.229	0.234	0.242	0.261
FordB	500	0.414	0.405	**0.263**	0.285	0.263	0.299	0.298
Win / Tie w.r.t DTW-C	-	-	22/30	**25/30**	20/30	22/30	22/30	21/30

Table 1: Classification error rates. TN-C_{Li} is classifier using ith layer.

the performance of DTW-C on 83% (25/30) datasets. This suggests that *a pre-trained TimeNet without domain-specific tuning for feature extraction gives embeddings which serve as relevant features for TSC.* Also, TN-C exceeds the performance of SAE-C on 61% (19/31) datasets suggesting that TimeNet can even be used for domains it is not trained on. TN-C is better than the state-of-art PROP on 4 out of 15 test datasets for which the results have been reported in [15]. We further test the robustness of TimeNet embeddings by reducing the amount of labeled data to two-thirds and learn a classifier TN-$C_{2/3}$, and observe that TN-$C_{2/3}$ performs better than DTW-C on 66% (20/30) datasets.

We evaluate the relevance of different layers of TimeNet by training SVM classifier TN-C_{Li} using embeddings from only the ith hidden layer. We observe that for datasets with small T, one layer of TimeNet gives classification performance comparable to TN-C (refer Table 1). For datasets with large T, TN-C is

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

better than any TN-C_{Li}. This suggests that *for shorter time series, one of the three layers extracts relevant features from time series, whereas for longer time series all layers carry relevant information.*

4 Discussion

We exploit seq2seq models trained in an unsupervised manner on diverse time series to obtain a deep RNN, namely TimeNet, that transforms time series to fixed-dimensional representations or embeddings. TimeNet could produce effective embeddings for time series classification on a wide variety of time series data not seen during training. TimeNet embeddings perform better than i) embeddings from RNN trained specifically for the problem domain, and ii) DTW based classifier. Our results suggest that TimeNet can be used to extract deep features for time series analysis when obtaining labeled training data is expensive.

References

[1] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[2] Ali Sharif Razavian, Hossein Azizpour, et al. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE CVPR Workshops*, pages 806–813, 2014.

[3] Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in NIPS*, pages 190–198, 2013.

[4] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, et al. Long short term memory networks for anomaly detection in time series. In *Proceedings of 23rd ESANN*, pages 89–94, 2015.

[5] Kyunghyun Cho, Bart Van Merriënboer, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

[6] Yu-An Chung et al. Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks. *arXiv:1603.00982*, 2016.

[7] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *In Anomaly Detection Workshop at 33rd ICML. arxiv:1607.00148*, 2016.

[8] Pankaj Malhotra, Vishnu TV, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, P. Agarwal, and G. Shroff. Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. *1st ACM SIGKDD Workshop on ML for PHM*, 2016.

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[10] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[11] Mohit Yadav, Pankaj Malhotra, Lovekesh Vig, K Sriram, and Gautam Shroff. Ode-augmented training improves anomaly detection in sensor data from machines. In *NIPS Time Series Workshop. arXiv preprint arXiv:1605.01534*, 2015.

[12] Arthur Le Guennec et al. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD Workshops*, 2016.

[13] Vu Pham et al. Dropout improves recurrent neural networks for handwriting recognition. In *14th ICFHR*, pages 285–290. IEEE, 2014.

[14] Yanping Chen, Eamonn Keogh, et al. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/\~eamonn/time_series_data/`.

[15] Jason Lines and Anthony Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.