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/.

Attention-based Information Fusion using Multi-Encoder-Decoder Recurrent Neural Networks

Stephan Baier[1], Sigurd Spieckermann[2] and Volker Tresp[1,2]

1- Ludwig Maximilian University
Oettingenstr. 67, Munich, Germany

2- Siemens AG, Corporate Technology
Otto-Hahn-Ring 6, Munich, Germany

Abstract. With the rising number of interconnected devices and sensors, modeling distributed sensor networks is of increasing interest. Recurrent neural networks (RNN) are considered particularly well suited for modeling sensory and streaming data. When predicting future behavior, incorporating information from neighboring sensor stations is often beneficial. We propose a new RNN based architecture for context specific information fusion across multiple spatially distributed sensor stations. Hereby, latent representations of multiple local models, each modeling one sensor station, are jointed and weighted, according to their importance for the prediction. The particular importance is assessed depending on the current context using a separate attention function. We demonstrate the effectiveness of our model on three different real-world sensor network datasets.

1 Introduction

In this paper we propose a recurrent neural network (RNN) architecture for combining information from multiple data streams in a distributed sensor network. With the rising number of connected devices and sensors, often referred to as the Internet of Things (IoT), modeling sensor networks and multi-agent systems is of increasing interest. We consider sensor networks consisting of multiple stations, where each station can measure multiple features at a single location. We address the task of sequence-to-sequence prediction, although our proposed architecture can easily be generalized to other tasks such as classification, recommendation, or anomaly detection. We build dedicated RNN models for all sensor stations, which are allowed to exchange information among each other to enable exploitation of cross-device correlations. The model, which we refer to as the "multi-encoder-decoder model", is an extension of the general encoder-decoder framework, which has become popular in various tasks such as machine translation, image caption generation and automatic speech recognition [1][2]. The idea of using multiple encoders and decoders has also recently been considered in natural language processing [4, 5, 6]. We propose an interconnection layer, which joins the latent representations of all encoders using an attention mechanism. Thereby, the attention mechanism, which was originally developed for neural machine translation (see [3][2]), is applied in a novel context which could also be useful for further sensor fusion tasks. We demonstrate the effectiveness of our proposed multi-sequence-to-sequence network on three datasets. The experimental results show that the proposed attention-based multi-encoder-decoder model outperforms competitive linear models and standard RNN architectures.

2 Representation Fusion Model

We propose a model which learns representations for multiple senor stations. The representations are fused using an attention mechanism. Finally, decoder models make predictions based on the fused representations. The whole system is completely differentiable and can thus be trained directly end-to-end. Figure 1 shows the model schematically.

2.1 Multi-Encoder-Decoder Model

We consider the task of predicting multiple multivariate output sequences given multiple multivariate input sequences. We apply the sequence-to-sequence model [1] to multiple data streams by creating multiple encoder and decoder functions. The multiple sequence-to-sequence models communicate through an interconnection layer, which acts like a soft-switching circuit between the single models. The input sequences are represented by a three-way tensor $\mathcal{X} \in \mathbb{R}^{E \times T_{\text{enc}} \times F_{\text{enc}}}$, where E denotes the number of encoder devices, T_{enc} denotes the encoder sequence length and F_{enc} is the number of encoder features. Similarly, the output sequences are represented by a three-way tensor $\mathcal{Y} \in \mathbb{R}^{D \times T_{\text{dec}} \times F_{\text{dec}}}$, where D denotes the number of decoder devices, T_{dec} denotes the decoder sequence length and F_{dec} is the number of decoder features. In the case of multivariate streaming data from a sensor network, the value $\mathcal{X}_{i,t,j}$ corresponds to the j-th feature measured at the i-th sensor station at time t. Similarly, the value $\widehat{\mathcal{Y}}_{i,t,j}$ corresponds to the prediction of the j-th feature at the i-th output node at time t. If we consider, for example, the task of predicting the features of the next T_{dec} values for all stations in a sensor network, then D is the number of stations, F_{dec} is the number of features and T_{dec} is the time period for which forecasts are performed.

Each input-sensing device is modeled by an encoder function

$$f_{\text{enc},i}(\mathcal{X}_{i,:,:}) = e_i, \quad \text{with } i \in \{1, 2, ..., E\}, \tag{1}$$

which takes the data measured at the i-th sensing device as input and outputs a latent representation $e_i \in \mathbb{R}^{\dim(e_i)}$. For each output device an interconnection function $f_{\text{con},j}$ combines the representations $\{e_i\}_{i=1}^{E}$ as

$$f_{\text{con},j}(\{e_i\}_{i=1}^{E}) = c_j, \quad \text{with } j \in \{1, 2, ..., D\}. \tag{2}$$

with $c_j \in \mathbb{R}^{\dim(e_i)}$. Finally, for each output device a decoder function $f_{\text{dec},j}$ models the prediction given the respective combined representation c_j as

$$f_{\text{dec},j}(c_j) = \widehat{\mathcal{Y}}_{j,:,:}, \quad \text{with } j \in \{1, 2, ..., D\}. \tag{3}$$

This way information between the different input and output sequences can be exchanged through the interconnection layer.

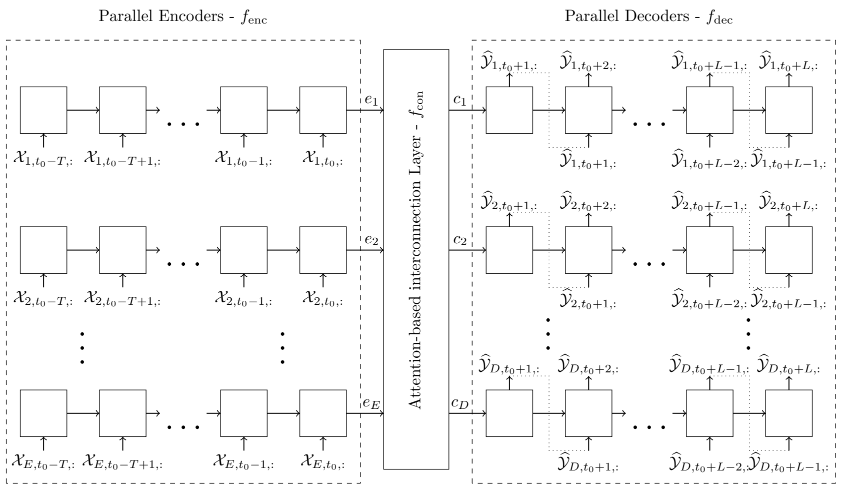
Parallel Encoders - f_{enc}
Parallel Decoders - f_{dec}
Attention-based interconnection Layer - f_{con}
e_1
e_2
e_E
c_1
c_2
c_D
$\mathcal{X}_{1,t_0-T,:}$
$\mathcal{X}_{1,t_0-T+1,:}$
$\mathcal{X}_{1,t_0-1,:}$
$\mathcal{X}_{1,t_0,:}$
$\widehat{\mathcal{Y}}_{1,t_0+1,:}$
$\widehat{\mathcal{Y}}_{1,t_0+2,:}$
$\widehat{\mathcal{Y}}_{1,t_0+L-1,:}$
$\widehat{\mathcal{Y}}_{1,t_0+L,:}$
$\widehat{\mathcal{Y}}_{1,t_0+L-2,:}$

Fig. 1: Unfolded multi-encoder-decoder recurrent neural network for multiple sequence-to-sequence prediction.

Figure 1 shows the architecture of a multi-encoder-decoder recurrent neural network model. For the sequence-to-sequence prediction, we model each encoder and each decoder function with an RNN. Each encoder RNN iterates over the sequence produced by the respective sensing node. Thus, the input of the i-th encoder RNN is $x_t = \mathcal{X}_{i,t,:}$. We define the last hidden state of the i-th encoder RNN to be the encoder output e_i. For each decoder RNN a combined representation is computed by the respective interconnection function, which is used as initial hidden representation. The decoder output $\widehat{\mathcal{Y}}_{i,t-1,:}$ is copied to the input of the i-th decoder RNN at time t.

2.2 Spatial Attention Mechanism

The interconnection layer is implemented using an attention mechanism, where the combination of latent representations is not fixed for every prediction but depends on the current context, which is encoded in the input representations. The attention mechanism assesses the importance of the representations of the encoding devices e_i and computes a weighted sum over the latent vectors

$$c_j = \frac{1}{E}\sum_{i=1}^{E} w_{ji} e_i, \tag{4}$$

where the weights $w_{ij} \in \mathbb{R}$ are derived from an additional attention function f_{att}. The attention function is modeled by an additional feed-forward neural network. The outputs of the attention function are normalized through a softmax function, such that

$$z_{ji} = f_{\text{att},j}(e_i) \tag{5a}$$

$$w_{ji} = \frac{\exp(z_{ji})}{\sum_{k=1}^{E} \exp(z_{jk})}. \tag{5b}$$

Whether attention is put on a representation e_i or not can vary for each prediction, depending on the encoded information in e_i. The approach draws inspiration from the attention-based machine translation model [3], however the attention is not used across time but spatially across sensing devices.

Note that this mechanism can deal with a variable amount of input devices, which is especially useful in settings where the number of input-devices is not constant over time, e.g. moving devices where devices appear and disappear over time, or where some input devices do not send any data, e.g. due to broken sensors.

2.3 Model Training

The complete model is trained end-to-end by minimizing the negative log-likelihood of a historical training set $\mathcal{D} = \{(\mathcal{X}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^{N}$ w.r.t. the model parameters such that

$$L = -\sum_{n=1}^{N} \log\, p(\mathcal{Y}^{(n)} | \mathcal{X}^{(n)}; \Phi), \tag{6}$$

where Φ includes the parameters of all encoders and decoders, as well as the parameters of the feedforward neural network for the attention function. The cost function is minimized using stochastic gradient descent with mini batches.

3 Experiments

We evaluate the performance of the multi-encoder-decoder network using sequence-to-sequence prediction in sensor networks on two climatological datasets and a smart grid dataset. We choose the task to be the prediction of future network behavior given a sequence of past measurements. Predictions are made for every sensor station and all features, thus, $E = D$ and $F_{\text{enc}} = F_{\text{dec}}$.

3.1 Datasets

We consider a sensor network of environmental sensing stations measuring climatological data on an hourly basis. The dataset consists of 18 stations distributed across Quebec, each measuring air temperature, dew point, relative humidity and wind speed. The second dataset is a sensor network of 15 environmental sensors spread across Alberta measuring the same features. We downloaded 5 years of data between 2010 and 2014 from ASOS[1] and selected stations and features with the least missing values. We extracted sequences of 72 hours as

[1]https://mesonet.agron.iastate.edu/request/download.phtml

Dataset	Quebec	Alberta	Smart Grid
Last observed values	65.15	72.95	51.69
Linear regression per station	42.89	41.89	33.82
Linear regression all stations	35.62	34.87	31.64
Regular RNN per station	38.17	34.92	31.50
Regular RNN all stations	34.77	34.68	29.56
Multi-enc-dec RNN attention	32.28	32.89	28.84

Table 1: Mean squared error results for the climatological and smart grid test sets in percent.

input to the encoders and made predictions for the next 24 hours. The data gathered between 2010 and 2013 was used for training and validation while the data gathered in 2014 was used for testing the models. In the second experiment we predict the load profiles of the next 3 days given the last 21 (3 weeks) load profiles from certain areas. We selected 18 zones with historical load profiles gathered between 2007 and 2014 from the smart grid dataset [7]. As there is only one measurement we chose the input and target features to be the hourly load and performed the forecasts on a daily basis.

3.2 Methods

We compare our model to multiple linear regression, which has shown state-of-the-art performance in the task of energy load forecasting [7]. Further, we compare against regular RNN models. Both, the linear and the RNN models are trained in two different settings: (i) a separate model for each station, i.e. no cross-correlations can be exploited and (ii) a joint model for all stations, i.e. cross-correlations between stations can be exploited. We evaluate on the normalized data to get a baseline mean squared error of 1.0 for predicting the historical mean. Further, we report as a baseline the constant prediction of the last observed value for each measured feature. For all models the optimal size of the hidden state was determined on the validation set. This resulted in a size of 130 hidden neurons for the RNNs modeling single stations and 300 hidden neurons for the RNNs which model all stations jointly. We also tried the extensions gated recurrent units (GRU) and long short-term memory (LSTM), however the prediction results did not significantly improve. In [8] it has also been found that LSTMs are not particularly well suited for time series forecasting. All experiments where implemented using Theano [9].

3.3 Results

Table 1 shows the results for both datasets. On the climatological dataset we can see that both the RNN and linear model perform significantly better when all stations are integrated into one model compared to one dedicated model for each station. This observation indicates strong cross-correlations between the stations. Using individual RNNs per station performs better than the linear regression model per station, and the joint RNN for all stations outperforms the linear model for all stations. Our proposed multi-encoder-decoder model with spatial attention achieves the best result. This indicates that the attention function helps exploiting the non-linear cross correlations in the overall system. For the smart grid dataset the prediction of the load profile of the last day (last observed values) is already a good baseline as the profiles do not change drastically within three days. Also here the linear model with all stations included slightly improves the prediction over the single models and also the RNN model including all stations outperforms the single per-station RNN models. Also on this dataset, the attention-based multi-encoder-decoder model yields better performance than the baseline models.

4 Conclusion

We proposed a neural network architecture for modeling distributed sensor networks, which extends the successful encoder-decoder framework. The fusion of hidden representations of multiple encoder networks using an attention mechanism, allows for exploiting cross-correlations across sensor stations. Using end-to-end training, the complete model consisting of the encoders, the interconnection layer with an attention mechanism, and the decoders is trained to predict a sequence of future behavior. In future work our architecture could also easily be extended to different prediction tasks such as classification or anomaly detection.

References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[2] KyungHyun Cho, Aaron C. Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *arXiv 1507.01053*, 2015.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[4] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[5] Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 2016.

[6] Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*, 2016.

[7] Tao Hong, Pierre Pinson, and Shu Fan. Global energy forecasting competition 2012. *International Journal of Forecasting*, 30(2):357–363, 2014.

[8] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Artificial Neural Networks ICANN 2001*, pages 669–676. Springer, 2001.

[9] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.