Spatio-temporal Stacked LSTM for Temperature Prediction in Weather Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

Long Short-Term Memory (LSTM) is a well-known method used widely on sequence learning and time series prediction. In this paper we deployed stacked LSTM model in an application of weather forecasting. We propose a 2-layer spatio-temporal stacked LSTM model which consists of independent LSTM models per location in the first LSTM layer. Subsequently, the input of the second LSTM layer is formed based on the combination of the hidden states of the first layer LSTM models. The experiments show that by utilizing the spatial information the prediction performance of the stacked LSTM model improves in most of the cases.

1 Introduction

The weather system is a challenging complex system and state-of-the-art methods utilize Numerical Weather Prediction (NWP) which is a computationally intense method [2]. Recently, data-driven approaches for weather forecasting have become a major interest of researchers as they are computationally simpler and more straightforward [3, 9, 10, 11].

Long Short-Term Memory (LSTM), proposed by Hochreiter & Schmidhuber [8], is a popular type of recurrent neural network which is able to capture long-term dependencies. LSTMs have been widely used and shown significant performance on different sequence learning problems and time series prediction [14, 17, 7, 4, 12, 18]. LSTMs have been also used in analyzing spatio-temporal datasets [15, 13]. Stacked LSTM is a deep architecture which consists of more than one layer of LSTM and the input of each LSTM layer is the hidden states of the previous LSTM layer [6, 17].

In this paper a spatio-temporal stacked LSTM model is proposed and its performance is evaluated on the application of temperature prediction. In the proposed model independent LSTM models per location are trained and afterward, the input of the second layer LSTM is formed based on the combination of the hidden states of the LSTM models in the first layer. It is worth mentioning that the general structure of the spatio-temporal stacked LSTM is similar to the approach proposed by Su et al. in the framework of Convolutional Neural Networks for 3D shape recognition [16]. Note that both stacked LSTM and spatio-temporal stacked LSTM methodologies can be explained as a multi-view approach as, similar to [9], they are fusing the information of different cities. Stacked LSTM and spatio-temporal stacked LSTM benefit from early fusion and intermediate fusion of the information from different views respectively.

2 Spatio-temporal Stacked LSTM

Assuming $i_t^{[l]}$, $f_t^{[l]}$, $o_t^{[l]}$, $c_t^{[l]}$ and $h_t^{[l]}$ to be the values of the input gate, forget gate, output gate, memory cell and hidden state at time t in the sequence and layer l respectively, and $x_{t,k}$ be the input of the system at time t at location k, the stacked LSTM model based on the architecture of the LSTM cell

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

defined in [5] is shown in Table 1. Note that x_t as an input of the model is a concatenation of the variables from all locations; i.e. $x_t = [x_{t,1}, x_{t,2}, \cdots, x_{t,c}]^T$. In this study we focus on a 2-layer stacked LSTM; however, the methodology can be extended to a larger number of layers. The full weight matrices W_{xj} for $j \in \{i, f, o, c\}$ are the weights that connect the input to the corresponding gates and the memory cell. The weight matrices W_{cj} for $j \in \{i, f, o\}$ are diagonal matrices that connect the cell memory to different gates. Note that the number of neurons for all gates is a predefined parameter and the equations are applied for each neuron. For simplicity, we use column vectors $w^{[l]}_{\text{lstm}}$ and $b^{[l]}_{\text{lstm}}$ to indicate all the elements in $\{W^{[l]}_{xi}, W^{[l]}_{ci}, W^{[l]}_{xf}, W^{[l]}_{hf}, W^{[l]}_{cf}, W^{[l]}_{xc}, W^{[l]}_{hc}, W^{[l]}_{xo}, W^{[l]}_{ho}, W^{[l]}_{co}\}$ and $\{b^{[l]}_i, b^{[l]}_f, b^{[l]}_c, b^{[l]}_o\}$ respectively.

Table 1: Equations of the stacked LSTM

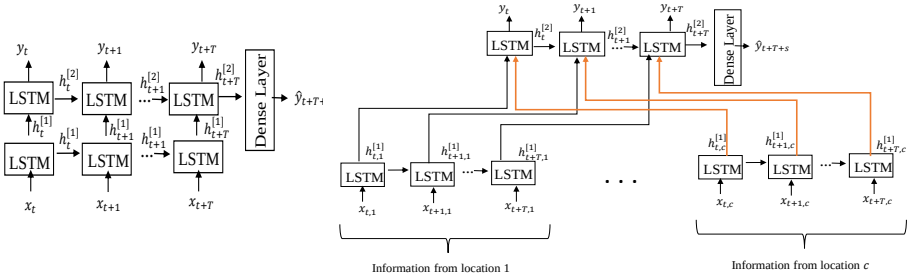
	Layer 1 LSTM	Layer 2 LSTM
Input	$x_t = [x_{t,1}, x_{t,2}, \cdots, x_{t,c}]$	$h^{[1]}_t$
Equations	$i^{[1]}_t = \sigma(W^{[1]}_{xi} x_t + W^{[1]}_{hi} h^{[1]}_{t-1} + W^{[1]}_{ci} c^{[1]}_{t-1} + b^{[1]}_i)$ $f^{[1]}_t = \sigma(W^{[1]}_{xf} x_t + W^{[1]}_{hf} h^{[1]}_{t-1} + W^{[1]}_{cf} c^{[1]}_{t-1} + b^{[1]}_f)$ $c^{[1]}_t = f^{[1]}_t \odot c^{[1]}_{t-1} + i^{[1]}_t \odot \tanh(W^{[1]}_{xc} x_t + W^{[1]}_{hc} h^{[1]}_{t-1} + b^{[1]}_c)$ $o^{[1]}_t = \sigma(W^{[1]}_{xo} x_t + W^{[1]}_{ho} h^{[1]}_{t-1} + W^{[1]}_{co} c^{[1]}_t + b^{[1]}_o)$ $h^{[1]}_t = o^{[1]}_t \odot \tanh(c^{[1]}_t)$	$i^{[2]}_t = \sigma(W^{[2]}_{h^{[1]}_t i} h^{[1]}_t + W^{[2]}_{hi} h^{[2]}_{t-1} + W^{[2]}_{ci} c^{[2]}_{t-1} + b^{[2]}_i)$ $f^{[2]}_t = \sigma(W^{[2]}_{xf} h^{[1]}_t + W^{[2]}_{hf} h^{[2]}_{t-1} + W^{[2]}_{cf} c^{[2]}_{t-1} + b^{[2]}_f)$ $c^{[2]}_t = f^{[2]}_t \odot c^{[2]}_{t-1} + i^{[2]}_t \odot \tanh(W^{[2]}_{xc} h^{[1]}_t + W^{[2]}_{hc} h^{[2]}_{t-1} + b^{[2]}_c)$ $o^{[2]}_t = \sigma(W^{[2]}_{xo} h^{[2]}_t + W^{[2]}_{ho} h^{[2]}_{t-1} + W^{[2]}_{co} c^{[2]}_t + b^{[2]}_o)$ $h^{[2]}_t = o^{[2]}_t \odot \tanh(c^{[2]}_t)$
Summary	$c^{[1]}_t = f(c^{[1]}_{t-1}, h^{[1]}_{t-1}, x_t; w^{[1]}_{\text{lstm}}, b^{[1]}_{\text{lstm}})$ $h^{[1]}_t = g(h^{[1]}_{t-1}, c^{[1]}_{t-1}, x_t; w^{[1]}_{\text{lstm}}, b^{[1]}_{\text{lstm}})$	$c^{[2]}_t = f(c^{[2]}_{t-1}, h^{[2]}_{t-1}, h^{[1]}_t; w^{[2]}_{\text{lstm}}, b^{[2]}_{\text{lstm}})$ $h^{[2]}_t = g(h^{[2]}_{t-1}, c^{[2]}_{t-1}, h^{[1]}_t; w^{[2]}_{\text{lstm}}, b^{[2]}_{\text{lstm}})$

In Table 2 the equations for the proposed spatio-temporal stacked LSTM model are shown. As is explained, instead of one LSTM model in the first layer, there are independent LSTM models per location. Hence, having 5 locations, 5 LSTM models are created. The full weight matrices $W_{xj,k}$ for $j \in \{i, f, o, c\}$ refer to the connection of the data of location k to different gates in the corresponding LSTM model. Similarly, other weight matrices, biases, gate values, memory cell and hidden state used subscript k to indicate that they correspond to the LSTM related to the location k. For simplicity, we use column vectors $w^{[l]}_{\text{lstm},k}$ and $b^{[l]}_{\text{lstm},k}$ which include all the elements in $\{W^{[l]}_{xi,k}, W^{[l]}_{ci,k}, W^{[l]}_{xf,k}, W^{[l]}_{hf,k}, W^{[l]}_{cf,k}, W^{[l]}_{xc,k}, W^{[l]}_{hc,k}, W^{[l]}_{xo,k}, W^{[l]}_{ho,k}, W^{[l]}_{co,k}\}$ and $\{b^{[l]}_{i,k}, b^{[l]}_{f,k}, b^{[l]}_{c,k}, b^{[l]}_{o,k}\}$ to refer to the parameters for the LSTM part related to location k for $k \in \{1, 2, \cdots c\}$ in the first layer. The information from different locations are then combined by merging the hidden states of the first layer and passing it as input to the second layer. For the second LSTM layer the definition of $w^{[l]}_{\text{lstm}}$ and $b^{[l]}_{\text{lstm}}$ remains the same. Figure 1 depicts the stacked LSTM and spatio-temporal stacked model when the number of layers is equal to two. As is shown, for the stacked LSTM model, the hidden states of the first layer are used as the input of the second layer. For the second LSTM layer the definition of $w^{[l]}_{\text{lstm}}$ and $b^{[l]}_{\text{lstm}}$ remains the same. However, in case of spatio-temporal stacked LSTM, there are independent LSTM models per location, and afterward the input of the second LSTM layer is defined based on the combination of the hidden states of the LSTM models of the first layer. Note that if the number of layers is more that two, the merging of the hidden states is possible at any layer before the last LSTM layer; e.g. if we have a 3-layer spatio-temporal stacked LSTM model, the combination of the hidden states can happen after the first LSTM layer or the second one. Note that in both stacked LSTM and spatio-temporal stacked LSTM models, after the second LSTM layer, a dense layer is used. The final prediction can be done by using $\hat{y}_{t+T+q} = w^T_{\text{dense}} h^{[2]}_{t+T} + b^{[2]}_{\text{dense}}$ where q is the number of days ahead to predict, T is the input sequence length, and w_{dense} and b_{dense} are the weights and bias term in the dense layer. For the experiments, we use a quadratic loss function to train the network and utilize $L2$-norm regularization to avoid overfitting.

One of the advantages of the proposed spatio-temporal stacked LSTM method is a smaller number of parameters in comparison with stacked LSTM. Assume the total number of neurons in the first layer and second layer is similar in both cases; in other words, if the number of neurons in stacked LSTM model in the first layer is n_1, then in the spatio-temporal stacked LSTM, each LSTM model in the first layer has $\frac{n_1}{c}$ neurons where c is the number of locations. In this case, the spatio-temporal stacked LSTM is similar to the case that the full weight matrices in stacked LSTM are considered to be block diagonal at which point each block is related to a location. Hence, the number of parameters to be optimized is smaller in the proposed method. This makes the spatio-temporal stacked LSTM a

Table 2: Equations of the spatio-temporal stacked LSTM

	Layer 1 LSTM (for $k \in \{1, 2, \cdots, c\}$)	Layer 2 LSTM
Input	$x_{t,k}$	$h_t^{[1]} = [h_{t,1}^{[1]}, h_{t,2}^{[1]}, \cdots, h_{t,c}^{[1]}]$
Equations	$i_{t,k}^{[1]} = \sigma(W_{xi,k}^{[1]} x_{t,k} + W_{hi,k}^{[1]} h_{t-1,k}^{[1]} + W_{ci,k}^{[1]} c_{t-1,k}^{[1]} + b_{i,k}^{[1]})$ $f_{t,k}^{[1]} = \sigma(W_{xf,k}^{[1]} x_{t,k} + W_{hf,k}^{[1]} h_{t-1,k}^{[1]} + W_{cf,k}^{[1]} c_{t-1,k}^{[1]} + b_{f,k}^{[1]})$ $c_{t,k}^{[1]} = f_{t,k}^{[1]} \odot c_{t-1,k}^{[1]} + i_{t,k}^{[1]} \odot \tanh(W_{xc,k}^{[1]} x_{t,k} + W_{hc,k}^{[1]} h_{t-1,k}^{[1]} + b_{c,k}^{[1]})$ $o_{t,k}^{[1]} = \sigma(W_{xo,k}^{[1]} x_{t,k} + W_{ho,k}^{[1]} h_{t-1,k}^{[1]} + W_{co,k}^{[1]} c_{t,k}^{[1]} + b_{o,k}^{[1]})$ $h_{t,k}^{[1]} = o_{t,k}^{[1]} \odot \tanh(c_{t,k}^{[1]})$	$i_t^{[2]} = \sigma(W_{h_{t,i}^{[1]} i}^{[2]} h_t^{[1]} + W_{hi}^{[2]} h_{t-1}^{[2]} + W_{ci}^{[2]} c_{t-1}^{[2]} + b_i^{[2]})$ $f_t^{[2]} = \sigma(W_{xf}^{[2]} h_t^{[1]} + W_{hf}^{[2]} h_{t-1}^{[2]} + W_{cf}^{[2]} c_{t-1}^{[2]} + b_f^{[2]})$ $c_t^{[2]} = f_t^{[2]} \odot c_{t-1}^{[2]} + i_t^{[2]} \odot \tanh(W_{xc}^{[2]} h_t^{[1]} + W_{hc}^{[2]} h_{t-1}^{[2]} + b_c^{[2]})$ $o_t^{[2]} = \sigma(W_{xo}^{[2]} h_t^{[2]} + W_{ho}^{[2]} h_{t-1}^{[2]} + W_{co}^{[2]} c_t^{[2]} + b_o^{[2]})$ $h_t^{[2]} = o_t^{[2]} \odot \tanh(c_t^{[2]})$
Summary	$c_{t,k}^{[1]} = f(c_{t-1,k}^{[1]}, h_{t-1,k}^{[1]}, x_{t,k}; w_{\text{lstm},k}^{[1]}, b_{\text{lstm},k}^{[1]})$ $h_{t,k}^{[1]} = g(h_{t-1,k}^{[1]}, c_{t-1,k}^{[1]}, x_{t,k}; w_{\text{lstm},k}^{[1]}, b_{\text{lstm},k}^{[1]})$	$c_t^{[2]} = f(c_{t-1}^{[2]}, h_{t-1}^{[2]}, h_t^{[1]}; w_{\text{lstm}}^{[2]}, b_{\text{lstm}}^{[2]})$ $h_t^{[2]} = g(h_{t-1}^{[2]}, c_{t-1}^{[2]}, h_t^{[1]}; w_{\text{lstm}}^{[2]}, b_{\text{lstm}}^{[2]})$



(a) Two-layer stacked LSTM (b) Two-layer spatio-temporal stacked LSTM

Figure 1: The scheme of the stacked LSTM and the proposed spatio-temporal stacked LSTM models when the number of layers is equal to two.

better choice when the number of samples in the training set is relatively small. On the other hand, in spatio-temporal stacked LSTM, the relation between the locations are taken into account in the second LSTM layer by combining the hidden states from the first layer.

3 Experiments

In this paper, the data have been collected from the Weather Underground company website [1] and cover a time period from the beginning of 2007 to mid-2014 for 5 cities including Brussels, Antwerp, Liege, Amsterdam and Eindhoven. To evaluate the performance of the proposed methods in various weather conditions, two test sets are defined: (i) from mid-November 2013 to mid-December 2013 (Nov/Dec) and (ii) from mid-April 2014 to mid-May 2014 (Apr/May). The data contain 18 measured weather variables, such as temperature and humidity, for each day per city. In order to benefit from all available data, the training data that is used for each test set includes the data from the beginning of 2007 until the day before the corresponding test set.

Figure 2: Weather stations (picture: Google maps)

In this study, the LSTM cell architecture described by Zaremba & Vinyals [19] implemented in TensorFlow has been used for the experiments. The considered range for the tuning parameters were selected empirically. For the number of neurons, in the stacked LSTM we examined $\{20, 40, 80, 160, 320, 640\}$ in the first layer and $\{32, 64, 128, 256\}$ for the second layer. In case of the spatio-temporal stacked LSTM, the number of neurons in LSTM per location in the first layer

Table 3: MAE of the stacked LSTM and the Spatio-Temporal stacked LSTM (ST stacked LSTM) for min. and max. temperature prediction in Nov/Dec and Apr/May test sets.

			Activation function :tanh				Activation function : sigmoid			
			MAE		MSE		MAE		MSE	
Testset	Steps ahead	Temp.	stacked LSTM	ST stacked LSTM	stacked LSTM	ST stacked LSTM	stacked LSTM	ST stacked LSTM	stacked LSTM	ST stacked LSTM
Nov/Dec	1	Min	1.66	**1.43**	4.36	**3.64**	1.69	**1.56**	4.04	**3.71**
		Max	**1.15**	1.22	**2.48**	2.65	1.37	**1.23**	3.87	**2.96**
	2	Min	2.30	**1.72**	8.17	**4.36**	1.86	**1.76**	5.12	**5.00**
		Max	1.89	**1.71**	6.51	**4.57**	1.65	**1.61**	**4.24**	4.99
	3	Min	3.04	**1.72**	14.16	**4.22**	**1.94**	**1.94**	5.35	**5.23**
		Max	3.44	**1.86**	17.30	**4.83**	**1.73**	**1.73**	**4.99**	5.34
	4	Min	3.28	**1.98**	17.52	**6.39**	1.66	**1.57**	4.16	**3.74**
		Max	2.56	**2.14**	9.36	**5.87**	**1.61**	1.76	**3.85**	3.87
	5	Min	3.72	**1.71**	22.20	**4.39**	**1.58**	**1.58**	**4.06**	4.13
		Max	2.75	**1.89**	11.30	**4.92**	1.58	**1.55**	**3.51**	3.65
	6	Min	3.23	**1.90**	14.06	**5.42**	**1.76**	1.90	**4.68**	5.09
		Max	4.01	**1.80**	22.06	**5.40**	**1.63**	1.68	**4.70**	4.79
Apr/May	1	Min	1.64	**1.60**	**4.15**	4.63	1.58	**1.55**	**3.78**	3.92
		Max	**2.27**	2.45	**7.66**	8.51	**2.24**	2.27	8.00	**7.93**
	2	Min	2.52	**2.15**	12.36	**9.09**	2.01	**1.96**	7.86	**7.75**
		Max	**2.64**	**2.64**	11.87	**10.37**	2.77	**2.55**	10.03	**9.38**
	3	Min	2.83	**2.20**	14.58	**9.07**	2.09	**2.03**	8.86	**8.54**
		Max	3.61	**3.03**	18.63	**12.75**	**2.53**	2.58	**8.98**	9.29
	4	Min	2.63	**2.25**	12.09	**9.59**	**2.03**	2.07	**8.27**	8.36
		Max	3.08	**2.92**	13.02	**12.69**	2.72	**2.59**	10.18	**9.75**
	5	Min	2.63	**2.51**	**11.71**	12.36	2.36	**2.31**	10.17	**10.05**
		Max	3.62	**2.89**	19.82	**12.89**	**2.64**	2.99	**10.17**	13.46
	6	Min	3.04	**2.93**	15.58	**15.39**	**2.46**	2.53	**10.92**	11.99
		Max	**2.77**	2.99	**10.99**	13.51	3.04	**2.95**	12.58	**12.15**

is considered to be in the set of $\{4, 8, 16, 32, 64, 128\}$. Note that as there are 5 locations in the first layer, the total number of neurons in the first layer are similar in both models. For the inner state, we deployed both tanh and sigmoid as the activation function. In the experiments, the sequence length is considered to be 10.

The experiments were conducted for the prediction of the minimum and maximum temperature in Brussels for 1 to 6 days ahead. To avoid local minima problems in neural networks, the experiments are repeated 5 times and the median Mean Absolute Error (MAE) and the Mean Squared Error (MSE) on both test sets are presented in Table 3. As is shown, using sigmoid as the inner activation function can result in better performance. Moreover, it can be seen that in most of the cases taking the spatial information into account can improve the performance. This is more evident in case the activation function in the inner state is tanh.

In addition, the comparison between the performance of the stacked LSTM, the proposed method and Weather Underground over the two test sets together is depicted in Figure 3.

As is shown, although very few locations are taken into account, the LSTM models (as data-driven approaches) outperform the state-of-the-art method used for minimum temperature prediction. Also, for maximum temperature prediction the performance of the LSTM models are competitive with the performance of the state-of-the-art methods.

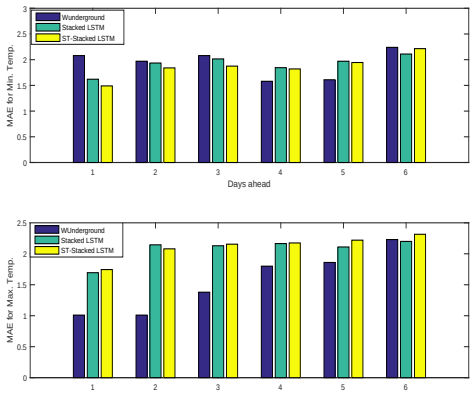


Figure 3: Comparing MAE of min. and max. temperature prediction for Weather Underground, stacked LSTM, and Spatio-Temporal stacked LSTM (ST stacked LSTM)

4 Conclusion

In this study, we proposed a spatio-temporal stacked LSTM model at which point in the first layer different LSTM models are considered per location, and then the corresponding hidden states are merged and given as input to the next layer. The proposed method was deployed in an application of weather forecasting.

The experimental results suggest that considering spatio-temporal property of the data in the LSTM model can improve the performance of the prediction. Moreover, it is shown that the proposed method is competitive with the state-of-the-art method in weather forecasting.

References

[1] Weather underground. www.wunderground.com.

[2] BAUER, P., THORPE, A., AND BRUNET, G. The quiet revolution of numerical weather prediction. *Nature 525*, 7567 (2015), 47–55.

[3] FENG, C., CUI, M., HODGE, B.-M., AND ZHANG, J. A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Applied Energy 190* (2017), 1245–1257.

[4] FREEMAN, B. S., TAYLOR, G., GHARABAGHI, B., AND THÉ, J. Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association*, just-accepted (2018).

[5] GRAVES, A. Generating sequences with recurrent neural networks. *preprint arXiv:1308.0850* (2013).

[6] GRAVES, A., JAITLY, N., AND MOHAMED, A.-R. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on* (2013), IEEE, pp. 273–278.

[7] GRAVES, A., MOHAMED, A.-R., AND HINTON, G. Speech recognition with deep recurrent neural networks. In *ICASSP* (2013), IEEE, pp. 6645–6649.

[8] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.

[9] HOUTHUYS, L., KAREVAN, Z., AND SUYKENS, J. A. K. Multi-view LS-SVM regression for black-box temperature prediction in weather forecasting. *IJCNN* (2017), 1102–1108.

[10] HU, Q., SU, P., YU, D., AND LIU, J. Pattern-based wind speed prediction based on generalized principal component analysis. *IEEE Transactions on Sustainable Energy 5*, 3 (2014), 866–874.

[11] KAREVAN, Z., FENG, Y., AND SUYKENS, J. A. K. Moving least squares support vector machines for weather temperature prediction. In *Proc. of the European Symposium on Artificial Neural Networks* (2016), pp. 611–616.

[12] LIPTON, Z. C., KALE, D. C., ELKAN, C., AND WETZEL, R. Learning to diagnose with LSTM recurrent neural networks. *ICLR* (2016).

[13] LIU, J., SHAHROUDY, A., XU, D., AND WANG, G. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *European Conference on Computer Vision* (2016), Springer, pp. 816–833.

[14] NG, J. Y.-H., HAUSKNECHT, M., VIJAYANARASIMHAN, S., VINYALS, O., MONGA, R., AND TODERICI, G. Beyond short snippets: Deep networks for video classification. In *CVPR, 2015* (2015), IEEE, pp. 4694–4702.

[15] PATRAUCEAN, V., HANDA, A., AND CIPOLLA, R. Spatio-temporal video autoencoder with differentiable memory. *preprint arXiv:1511.06309* (2015).

[16] SU, H., MAJI, S., KALOGERAKIS, E., AND LEARNED-MILLER, E. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 945–953.

[17] SUTSKEVER, I., VINYALS, O., AND LE, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (2014), pp. 3104–3112.

[18] TIAN, Y., AND PAN, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on* (2015), IEEE, pp. 153–158.

[19] ZAREMBA, W., SUTSKEVER, I., AND VINYALS, O. Recurrent neural network regularization. *preprint arXiv:1409.2329* (2014).