Partition-wise Recurrent Neural Networks for Point-based AIS Trajectory Classification

Xiang Jiang¹², Erico N de Souza¹², Xuan Liu¹², Behrouz Haji Soleimani¹², Xiaoguang Wang¹, Daniel L. Silver² and Stan Matwin¹²⁴

Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada
 Institute for Big Data Analytics, Dalhousie University, Halifax, NS, Canada
 Jodrey School of Computer Science, Acadia University, Wolfville, NS, Canada
 Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. We present Partition-wise Recurrent Neural Networks (pRNNs) for point-based trajectory classification to detect fishing activities in the ocean. This method partitions each feature and uses region-specific parameters for distinct partitions, which can greatly improve the expressive power of deep recurrent neural networks on low-dimensional yet heterogeneous trajectory data. We show that our approach outperforms the state-of-the-art systems.

1 Introduction

Modern computing tools and data analysis algorithms have spawned strong interest in the analysis of different kinds of mobility data. Most common representation of mobility is in the form of trajectories of moving objects. A trajectory is a sequence of points, representing positions of a moving object—a runner, a car, or a ship—over time. Automatic Identification System (AIS)—a vessel identification system that monitors the GPS trajectories of ships worldwide, broadcasting information such as longitude, latitude, and speed —is an extremely valuable data asset for applied mobility research. Understanding the AIS data will enable, for the first time, scientific understanding of the effects of human marine activities on the world's oceans. Different approaches have been proposed for fishing activity detection, including Mixtured Gaussian [1], Hidden Markov Models [2], Lavielle's algorithm [2], Autoencoders [3], and Conditional Random Field (CRF) [4], among which CRF performed the best.

As deep learning has produced significant results in a wide range of applications, applying deep learning to the analysis of sequential mobility data in general, and AIS data in particular, is a logical research direction. Deep Recurrent Neural Networks (RNN), especially Long Short-Term Memory (LSTM) [5], Gated Recurrent Units (GRU) [6], and bidirectional RNN [7], have been proven highly effective when modeling sequential data. However, AIS records are low-dimensional and heterogeneous. It is difficult to develop deep models with them. This paper aims to overcome this problem so as to improve pointbased trajectory classification systems for identifying fishing activities from AIS data.

To overcome the challenges posed by AIS data, we propose Partition-wise Recurrent Neural Networks (pRNNs) which partition the feature space and model them jointly in a hierarchical/multilayer structure. pRNNs partition the candidate values of each feature into different regions and use region-specific parameters to model complex nonlinear input-output mappings. Our experimental results demonstrate substantial improvements over the state-of-the-art systems.

2 Partition-wise Recurrent Neural Networks

Motivation. The ability to learn high-dimensional and homogeneous data with hierarchical feature representations is the primary contributing factor in the success of deep learning [8, 9]. For instance, images are high-dimensional and distributed in the sense that each image has many pixels and can be modeled by a set of local filters collectively; they are homogeneous in the sense that the atomic components, pixels, represent the same kind of feature. This representation is highly redundant because distorting or corrupting some pixels can still preserve the semantics of an image. However, point-based trajectory data, such as AIS, are low-dimensional and have a dense representation of heterogeneous sensory inputs. Bengio and Xavier pointed out that, "a dense representation is highly entangled because almost any change in the input modifies most of the entries in the representation vector" [10], resulting in difficulties to "disentangle factors of variations in the data" [11]. This low-dimensional feature space also leads to a limited number of feature compositions which makes it hard to develop deep models. To overcome these challenges, we adopt the idea of feature-level partitioning [12, 13] which partitions each feature and associates different parameters with distinct regions of each feature to learn meaningful representations of the input data. Similar to piece-wise linear activations [14, 10, 15], the proposed method uses piece-wise learnable parameters to represent the activation functions of the input features, which is more expressive than traditional activation functions—such as sigmoid and hyperbolic tangent—in learning complex hypotheses. This method not only learns the activations of hidden units but also learns feature compositions between the hidden units in a hierarchical structure. Partition-wise Activation. For notations, we denote X as a vector of input variables, \mathbf{x} as an observation of \mathbf{X} where \mathbf{x} takes outcomes from a set of candidate values V, X_i as the i-th feature of **X** and takes outcomes from a set of candidate values V_i . $\mathbb{1}_{\{x_i \in V'_i\}}$ denotes an indicator function of x_i which takes value 1 when $x_i \in V'_i$ and 0 otherwise. $V'_i \in P_i$ where P_i is a finite partition of V_i that has cardinality of C. More intuitively, the feature space of X_i is partitioned into C non-overlapping subsets and is further parameterized using different parameters with the use of feature functions. We define feature function that takes the form:

$$f_{i}(x_{i}) = \sum_{k=1, V_{ik}' \in P_{i}}^{C} \mathbb{1}_{\left\{x_{i} \in V_{ik}'\right\}} \lambda_{ik}$$
(1)

which transforms the feature x_i to another space using parameters λ_{ik} depending on which partition x_i belongs to. Here V'_{ik} denotes the k-th subset of partition P_i . We obtain the feature function $f_i(x_i)$ by summing over each disjoint subset 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/.

 V'_{ik} of the partition P_i parameterized by λ_{ik} . The feature function $f_i(x_i)$ can approximate a wide variety of functions—including sigmoid, hyperbolic tangent and ReLU—when given appropriate parameters. The partition-wise activation is given by:

$$activation = \sigma\left(\sum_{i} f_i(x_i)\right).$$
(2)

Partition-wise Recurrent Networks. Partition-wise LSTM (pLSTM) is defined by:

$$\begin{pmatrix} i_t \\ o_t \\ f_t \\ \widetilde{C}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ tanh \end{pmatrix} \left(\sum_g f_g(x_t) + Wh_{t-1} \right)$$

$$C_t = i_t \odot \widetilde{C}_t + f_t \odot C_{t-1}$$

$$h_t = o_t \odot tanh(C_t),$$

$$(3)$$

where independent feature functions f_g are defined for input gate i_t , output gate o_t , forget gate f_t and candidate state \widetilde{C}_t . pGRU can be defined in similar ways by substituting the input transformation with partition-wise activations.

Gradient Property. The network parameters can be optimized using backpropagation. As the feature functions are piece-wise constant, we calculate the local gradient of activation with respect to the parameters λ of each partition:

$$\frac{\partial \sum_{i} f_{i}(x_{i})}{\partial \lambda_{jh}} = \frac{\partial f_{j}(x_{j})}{\partial \lambda_{jh}} = \frac{\partial \sum_{k=1, V_{ik}' \in P_{j}}^{C} \mathbb{1}_{\left\{x_{j} \in V_{jk}'\right\}} \lambda_{jk}}{\partial \lambda_{jh}} = \mathbb{1}_{\left\{x_{j} \in V_{jh}'\right\}}, \tag{4}$$

where λ_{jh} represents the parameter for h-th partition of the j-th feature. Note that the terms *i* and *k* are summed over resulting in the activation of input *x*, while terms *j* and *h* represent the specific parameter within λ that we are optimizing. As a result, the local gradient of feature functions with respect to the parameter λ can only take binary values and the parameters are updated only when the input feature belongs to a certain partition.

Partition Strategy. If the partition strategy can be defined prior to training, the feature functions can be equivalently reformulated as a preprocessing step: discretize and encode. The encoding can be fed directly to a neural network without any architectural modifications.

3 Experiments

Data. The data consist of 14 longliner vessels with 481,887 data points in total. The data are labeled by an expert into two classes, fishing and non-fishing with a ratio of 78% and 22%, respectively. The two classes are our goal for classification. Point-based features—including difference of longitude and latitude, speed over ground, course over ground and acceleration—are used to develop the model.

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/.



Fig. 1: Network architecture for point-based trajectory classification.

Model		batch number											A		
Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Avg.
CRF	79	82	73	75	82	79	69	77	73	75	75	96	92	90	79.8
$LSTM^{a^*}$	75	71	74	85	67	66	72	66	68	70	36	55	85	93	70.2
pLSTM1 ^{b*}	80	92	81	97	77	86	80	82	82	69	88	96	92	95	85.5
pLSTM2 ^{c*}	85	93	88	99	82	87	85	85	84	69	86	97	96	98	88.1
$\mathrm{GRU}^{\mathrm{d}*}$	78	79	75	80	78	75	73	70	77	68	35	56	90	94	73.4
$pGRU1^{e^*}$	80	92	83	91	80	90	85	82	81	70	82	98	95	96	86.1
$pGRU2^{f^*}$	94	96	90	96	86	94	88	91	94	98	92	98	97	97	93.7

^a Conventional LSTM. ^b pLSTM. ^c Bidirectional pLSTM with peephole connections. ^d Conventional GRU. ^e pGRU. ^f Bidirectional pGRU.

*All recurrent neural networks have the same architecture of two layers with 30 nodes

Table 1: AUC (%) of ILOBO for fishing activity detection.

Network Architecture. Figure 1 depicts the proposed architecture. It has a bidirectional pLSTM layer followed by multiple conventional bidirectional LSTM layers [7] and a Softmax layer on top of it. We used the ADAM optimizer [16] with mini-batches to optimize the sequence to sequence cross entropy loss. We used validation-based early stopping with a hold-out set to detect overfitting, and we used learning rate decay that gradually decreases the learning rate.

Classification results. We use AUC as the evaluation metric because the data is imbalanced. Table 1 lists the AUC of the fishing activity detection problem using Iterative-Leave-One-Batch-Out (ILOBO): in each iteration, a different trajectory is selected as testing data and the rest for training and validation. Table 2 shows the p-values between the pRNNs and conventional methods.

Comparison with RNN. Table 2 shows that pRNNs (pLSTM1, pLSTM2, pGRU1 and pGRU2) are significantly better than RNNs (LSTM and GRU) on the 95% confidence level. The reason is that trajectory features are heterogeneous and low-dimensional which can take advantage of the expressive power introduced by partition-wise activations. Table 1 shows that, for batch 11 and 12, in particular, pRNNs achieve about 40% improvement in AUC over RNNs.

Comparison with CRF. We find pLSTM and pGRU can both outperform CRF - the p-values are shown in Table 2 - among which the bidirectional pGRU performs the best. This demonstrates that pLSTM and pGRU can model more complex input-output relationships compared with CRF that uses feature engineering and discretization on the same set of features.

Effect of depth. Figure 2 shows the effect of depth on bidirectional pLSTM

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/.

Table 2: p-values of paired samples t-test between pRNNs and conventional methods.

	LSTM	GRU	\mathbf{CRF}
pLSTM1	0.002	0.013	0.013
pLSTM2	0.000	0.002	0.001
pGRU1	0.001	0.006	0.002
pGRU2	0.000	0.000	0.000
-			



Fig. 2: AUC with 95% confidence interval on various bidirectional pRNNs.



Fig. 3: (Best viewed in color) Probability distribution and learned activations.

and pGRU evaluated on the first batch. pGRU generally performs better and more stable under the same depth because the representation is more compact. **Visualizing learned activation functions.** Figure 3 shows three selected nodes for the calculation of candidate states of feature speed. While traditional activation functions (in green) are monotonic, the learned partition-wise activations (in bar plots) encompass more flexibility and learn the conditional probability distribution from data: the parameters have opposite signs when different classes dominate certain speed intervals (partition-wise activations at speed 3–7 and 11–13 have different signs). This demonstrates that the prior probability distributions are implicitly incorporated into the learned activation functions, which in part provides performance improvement to the model.

4 Conclusions

This paper proposes pRNNs that use partition-wise activations to improve pointbased trajectory classification systems on detecting fishing activities from AIS data—by learning the activation functions and feature compositions in a hierarchical structure. Experimental results demonstrate substantial improvements over the state-of-the-art systems. More recent experiments suggest the proposed method can be extended to other applications—GPS trajectory classification.

Acknowledgement. The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada for this research.

References

- Fabrizio Natale, Maurizio Gibin, Alfredo Alessandrini, Michele Vespe, and Anton Paulrud. Mapping fishing effort through AIS data. *PloS one*, 10(6):e0130746, 2015.
- [2] Erico N de Souza, Kristina Boerder, Stan Matwin, and Boris Worm. Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning. *PloS one*, 11(7):e0158248, 2016.
- [3] Xiang Jiang, Daniel L Silver, Baifan Hu, Erico N de Souza, and Stan Matwin. Fishing activity detection from AIS data using autoencoders. In *Canadian Conference on Artificial Intelligence*, pages 33–39. Springer, 2016.
- [4] Baifan Hu, Xiang Jiang, Erico Souza, Ronald Pelot, and Stan Matwin. Identifying Fishing Activities from AIS Data with Conditional Random Fields. In Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, volume 8, pages 47–52. IEEE, 2016.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.
- [7] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 6645–6649. IEEE, 2013.
- [8] Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.
- [9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.
- [10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Aistats, volume 15, page 275, 2011.
- [11] Yoshua Bengio. Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1):1–127, 2009.
- [12] Joseph Wang and Venkatesh Saligrama. Local supervised learning through space partitioning. In Advances in Neural Information Processing Systems, pages 91–99, 2012.
- [13] Hidekazu Oiwa and Ryohei Fujimaki. Partition-wise linear models. In Advances in Neural Information Processing Systems, pages 3527–3535, 2014.
- [14] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. Maxout networks. *ICML* (3), 28:1319–1327, 2013.
- [15] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830, 2014.
- [16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.