Bert-GCN: multi-sensors network prediction

Peng Liu*, *Zhuang* Li, *Yang* Cong, and *Yuheng* Xu

China Ship Research and Development Academy, 100101 Beijing, China

Abstract. With the application of neural network technologies such as GCN and GRU in sensor networks, the accuracy and robustness of multi-sensor prediction have been greatly improved. GCN effectively uses the spatial characteristics of the sensor network, and GRU effectively uses the temporal characteristics of the sensor network, so the PROPOSED T-GCN model has achieved excellent results. However, there are still shortcomings: i) The prediction is only for a single sensor feature, and multiple features cannot be trained at the same time. ii) Only the connections between sensors are considered, while the connections between multiple features of sensors are ignored. iii) Modeling for multiple features leads to the deepening of the model from 2d to 3D, resulting in slow model training and poor learning effect. To solve the above problems, this paper proposed the Bert-GCN model. Bert pre-training was added on the basis of the original GCN-GRU model to effectively improve the learning effect of multiple features of a single sensor.

Keywords: Multi-sonsor network, Graph convolutional network (GCN), Gated recurrent unit (GRU), Bert, Transformer.

1 Introduction

With the change of the world military strategy and the deepening of the reform of the world military struggle system, information-oriented system operation and system confrontation have become the main form of the future war. As the core of battlefield target perception, multi-sensor prediction also faces more severe challenges. How to realize efficient multi-sensor prediction is the key to solve the problem.

This topic starts with improving the prediction performance of multi-sensor to solve three problems existing in the training process of multi-feature of single sensor. i) It is slow to train multiple features at the same time to predict only a single sensor feature. ii) Only the connections between sensors are considered, while the connections between multiple features of sensors are ignored. iii) Modeling for multiple features deepens the model from two-dimensional to three-dimensional, resulting in poor learning effect.

Aiming at the above three problems, this paper proposes a training model based on Bert-GCN. The aim is to improve the resource scheduling capability of future sensor systems to cope with increasingly severe electronic interference environment and increasing threats of various targets. By constructing a networked, collaborative and information-based sensor

* Corresponding author: 15248156291@qq.com

resource scheduling system, the sensor configuration is optimized to realize the complementation of sensor cooperative detection capabilities and realize all-round, three-dimensional and multi-level collaborative resource scheduling. Sensor networks can continuously track and detect small targets, low altitude targets, high speed targets and high maneuvering targets in a wider and more flexible working mode. Bert is used to preprocess sensor features, which can not only compress the model to improve the training speed, but also effectively find the relationship between multiple features to reduce information loss.

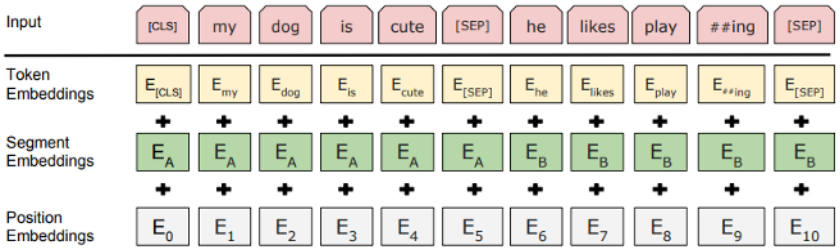

Fig. 1. Bert Embedding.

2 Relate works

2.1 Bert

Bert is a language processing model based on neural network. [1,2] Bert model pays more attention to identify the relationship between words in sentences or between sentences. It adopts semi-supervised learning and language to express the model. Bert is a bi-directional Transformer model [3]. It can adjust both left-to-right and right-to-left transformers. In the pre-training stage, Bert performs pre-training with unsupervised predictive tasks, including the Masked Language Model MLM(MLM) below. After pre-training, The Bert Model performs fine-turning for downstream tasks to fine-tune Model parameters. To achieve the most adaptive effect. Bert's depth bi-directional Transformer embodies this philosophy in fig.1.

There can be loops in the use of two-way interpretation, which can lead to a misunderstanding of the word itself. Bert adopted MLM model to solve this misunderstanding. The MLM model randomly masks the words of the input sentence (OpenAIGPT takes a similar approach) [4]. The word encoding of Bert model is not simple word encoding, but the combination of three layers of meaning encoding. The first layer of encoding is the encoding of the word itself. During Bert initialization, there will be an external input thesaurus for encoding, which will contain all the words of this natural language. The second layer of encoding is embedded based on the position information of words. In order to reflect the position information of words in sentences, Bert will position every word in every sentence. The third layer of coding is sentence-level coding. In order to embody the independence of sentences (Bert called it segment embedding), Bert uses the way of two-sentence stitching to construct coding [5]. After the three layers of embedding are completed, Bert combines the three kinds of embedding and finally determines the word vector.

2.2 Graph neural network

Graph is a kind of data structure, and graph neural network should be some models, methods and applications of deep learning on graph structure data. A common graph structure is

composed of nodes and edges. Nodes contain entity information and edges contain information about relationships between entities. Many learning tasks, such as modeling physical systems, learning molecular fingerprinting, predicting protein interfaces, and classifying diseases, require models to learn from the input of graph structures. GNN is generally divided into the following four categories: I) graph convolutional network [6-8] and graph attention network [9,10]. This kind of graph neural network believes that every node in the graph changes its state all the time due to the influence of neighbors and distant points until the final equilibrium. The closer the relationship is, the more influence the neighbor has. Ii) spatial network of graph [11,12] this model can effectively capture complex local spatio-temporal correlations through a well-designed spatio-temporal synchronization modeling mechanism. At the same time, several modules in different time periods are designed in the model to effectively capture the heterogeneity in the local spatio-temporal map. iii) self-coding of graphs [13,14]. In this model, the known graph is encoded to learn the distribution of node vector representation, and the vector representation of nodes is sampled from the distribution, and then the graph is reconstructed by decoding (link prediction). Iv) Graph generation network [15,16]. The model generates new graphs given a set of observed graphs.

3 Our work

Our work in this paper consists of Bert, GCN and GRU. Firstly, multiple parameters of multiple sensors are taken as Bert input. The purpose of Bert layer is to uniformly encode multiple features of a single sensor, compress data dimensions, and retain the internal relationship between parameters. Bert's output serves as the input of GCN. Multiple GCNS need to be built for multi-head Transformer, and the purpose of GCN layer is to build the spatial relationship between multi-sensor networks. The output of GCN is the input of GRU, and the purpose of GRU layer is to find the correlation on the timing of multi-sensor network. The network structure diagram is in Fig.2.

Fig. 2. Overview. Multiple features of sensors are used as Bert layer input, and the output matrix is input to GCN layer as feature matrix. Finally, time sequence correlation is realized by GRU layer.

Bert is a Transformer model using bidirectional encoder. It is made by stacking enconder structures of multiple Transformers. The model structure is shown in figure. In Transformer enconder, the data is first given a weighted feature vector Z by the self-attention module, as $Attention(Q, K, V)$.

$$Z = A\ \text{ttention}\ (Q, K, V) = \text{softma}\,\text{x}\left(\frac{QK^T}{\sqrt{d}_k}\right)V \tag{1}$$

Feed Forward Neural Network (FFN):

$$FFN(Z) = max(0, ZW_1 + b_1)W_2 + b_2 \tag{2}$$

The above is the structure of one-layer Transformer, Bert is the stack of multi-layer Transformer, and the model structure is shown in the fig. 3.

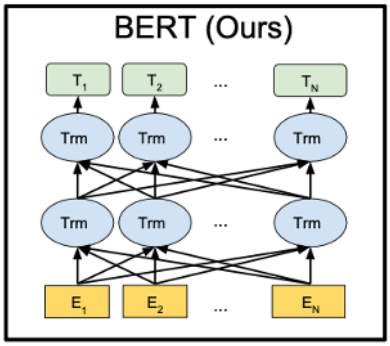

Fig. 3. Bert model.

For graph $G = (V, E)$,V is the set of nodes, E is the set of edges. For each node i, it has its characteristic x_i, which can be represented by matrix X_{N*D}. Where N represents the number of nodes, and D represents the feature dimension of each node, or the dimension of feature vector. Any graph convolution layer can be written as a nonlinear function: $v_i \in V\ (v_i, v_j) \in E$

$$H^{l+1} = f(H^l, A) \tag{3}$$

$H^0 = X$ is the input of the first layer, $x \in R^{N*D}$, and A is the adjacency matrix. Different models are selected for different problems, and the difference lies in the realization of the function F.

Given A adjacency matrix A and feature matrix X, the GCN model constructs A filter in the Fourier. The filter acts on the nodes of the graph and captures the spatial features between nodes through its first-order neighborhood. Then, the GCN model is constructed by superimposing multiple convolutional layers, which can be expressed as:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\hat{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}\theta^{(l)}\right) \tag{4}$$

We choose the 2-layer GCN model to capture spatial dependence, which can be expressed as:

$$f(X, A) = \sigma(\hat{A})ReLU(\hat{A}XW_0)W_1 \tag{5}$$

To prevent overfitting, L2 regularization is added.

$$loss = \left|\left|Y_t - \hat{Y}_t\right|\right| + \lambda L_{reg} \tag{6}$$

4 Experiment

4.1 Train model

In this section, we describe the concrete implementation of the Bert-GCN model. A Transformer encoder unit consists of a multi-head-attention + Layer Normalization + feedforword + Layer Normalization. Each BERT layer consists of one of these Encoder units.

In the large BERT model, there are 24 encoder layers, 16 Attention layers in each layer, and the dimension of word vector is 1024. In the smaller BERT model, there are 12 layers of encoder, each layer has 12 Attention, and the word vector dimension is 768. In all cases, set the size of the feed-forward/filter to 4H (H is the dimension of the word vector), i.e. 3072 when H = 768 and 4096 when H = 1024.

To evaluate the prediction performance of the Bert-GRU model, we use five metrics to evaluate the difference between the real traffic information Y_t and the prediction $\hat{Y}_t$, including:

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_t - \hat{Y}_t\right)^2} \tag{7}$$

(2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_t - \hat{Y}_t\right| \tag{8}$$

(3) Accuracy:

$$\text{Accuracy} = 1 - \frac{|Y-\hat{Y}|_F}{|Y|_F} \tag{9}$$

(4) Explained Variance Score (Var):

$$var = 1 - \frac{Var\{Y-\hat{Y}\}}{Var\{Y\}} \tag{10}$$

Root mean square error (RMSE) and MAE are used to measure the prediction error. The smaller the value, the better the prediction effect. Prediction accuracy is determined by precision: the higher the number, the better the prediction. Using Var to calculate the correlation coefficient, it shows the ability of the prediction results to match the actual data: the larger the value, the better the prediction effect.

4.2 Experimental results

We took GCN-GRU as baseline and compared it with the Bert-GCN model proposed in this paper, and compared the prediction effect of the two models in multi-sensor and multi-feature scenarios. It can be seen from the figure that Bert-GCN is more effective than the original GCN-GRU model in terms of accuracy and robustness. The following are the outcomes:

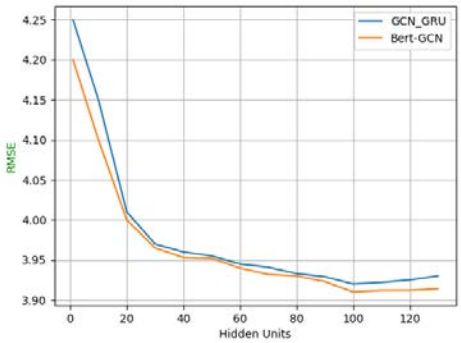

Fig.4. RMSE comparison of Bert-GCN and GCN-GRU.

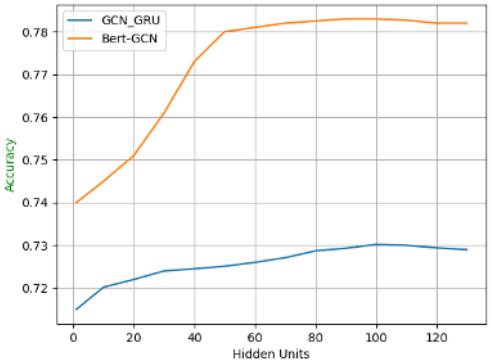

Fig.5. The accuracy comparison between the Bert-GCN and GCN-GRU.

5 Conclusion

In multi-sensor network prediction, based on GCN-GRU, this paper optimizes the complex scenario in which a single sensor contains multiple features. On the one hand, the data dimension is reduced and the performance is improved. On the other hand, the relationship between multiple features is effectively preserved and a good effect is achieved.

References

1. Tenney, Ian, Dipanjan Das, and Ellie Pavlick. "BERT rediscovers the classical NLP pipeline." arXiv preprint arXiv: 1905. 05950 (2019).
2. Polignano, Marco, et al. "Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets." 6th Italian Conference on Computational Linguistics, CLiC-it 2019. Vol. 2481. CEUR, 2019.
3. Gao, Zhengjie, et al. "Target-dependent sentiment classification with BERT." Ieee Access 7 (2019): 154290-154299.
4. Masala, Mihai, Stefan Ruseti, and Mihai Dascalu. "Robert–a romanian bert model." Proceedings of the 28th International Conference on Computational Linguistics. 2020.
5. Moradshahi, Mehrad, et al. "HUBERT untangles BERT to improve transfer across NLP tasks." arXiv preprint arXiv: 1910. 12647 (2019).
6. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
7. Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018, June). Modeling relational data with graph convolutional networks.
8. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019, May). Simplifying graph convolutional networks. In International conference on machine learning (pp. 6861-6871). PMLR.
9. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
10. Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., & Tang, J. (2019, January). Session-based social recommendation via dynamic graph attention networks. In

Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (pp. 555-563).

11. Zhang, K., Huang, Y., Du, Y., & Wang, L. (2017). Facial expression recognition based on deep evolutional spatial-temporal networks. IEEE Transactions on Image Processing, 26(9), 4193-4203.
12. Cho, Y. S., Galstyan, A., Brantingham, P. J., & Tita, G. (2013). Latent self-exciting point process model for spatial-temporal networks. arXiv preprint arXiv:1302.2671.
13. Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. Neurocomputing, 184, 232-242.
14. Lange, S., & Riedmiller, M. (2010, July). Deep auto-encoder neural networks in reinforcement learning. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
15. Deng, L., Seltzer, M. L., Yu, D., Acero, A., Mohamed, A. R., & Hinton, G. (2010). Binary coding of speech spectrograms using a deep auto-encoder. In Eleventh Annual Conference of the International Speech Communication Association.
16. Sainath, T. N., Kingsbury, B., & Ramabhadran, B. (2012, March). Auto-encoder bottleneck features using deep belief networks. In 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4153-4156). IEEE.