Improving pix2code based Bi-directional LSTM

Yanbin Liu
Chongqing Key Laboratory on Big Data for Bio Intelligence
CQUPT
Chongqing, China

Qidi Hu
College of Computer Science and Technology
CQUPT
Chongqing, China

Kunxian Shu
Chongqing Key Laboratory on Big Data for Bio Intelligence
CQUPT
Chongqing, China
shukx@cqupt.edu.cn

***Abstract*—Pix2code is a framework based on deep learning to transform a graphical user interface screenshot created by the designer into computer coder with 77% of accuracy. The architecture is based on CNN and LSTM.LSTM has been broadly applied to natural language processing about language model, which is both general and effective at capturing long-term dependencies. However, the standard LSTM predicting in time sequence ignores the contextual information of the future, but sometimes it is not enough just to look at the previous word. Computer code is a relative spatial relationship and not only needs to recognize token but also fully understands the structure of all sequences. In order to solve the problem, the pix2code model is optimized by Bidirectional LSTM, which allows the output layer to get complete past and future context information for each point in the input sequence. The model's transforming accuracy in the test set has been significantly improved reaching 85%.**

Keywords—pix2code, CNN, Bi-directional LSTM

I. INTRODUCTION

It is a boring job for web engineers to compile the codes of static web pages code based on a graphical user interface (GUI) mockup which was created by the designer. And a programmer will spend plenty of a great amount of time for coding. It is a good solution that applies deep learning to generate web code automatically. Due to the high complexity of native web code, domain-specific languages (DSLs) are generally designed. DSL contains the syntax and the semantics which were needed for modeling, and DSL does not use loop statements or control statements, so it greatly shortens the vocabulary. A recent research is DeepCoder [1], which shows the computer program produced by a combination of machine learning and traditional search techniques that apply DSL to reduce search space. Recently, a Pix2code model was proposed which applies deep learning to generate code [2]. The system can generate GUI code according to the input user page graphics for Android, iOS or web interface. The architecture consists of three modules, which are the visual model, the language model, and the decoder. The visual model is based on VGGNet [3]. Its input is an image (GUI) with the size of 256*256, extracting image features from the graph to generate a fixed-length vector containing image features. The language model uses LSTM, which can handle random-length sequences and capture long -term dependencies [4], and the structure avoids gradient explosion and disappearance, to an extent. LSTM has excellent performance in text sequence problems and NLP tasks [5][6].This input of model is a one-hot vector mapping to DSL context. The language model performs token-level language modeling. The Decoder in the last layer concatenates the vectors that are produced by the language model and the visual model, learning the connections between the user interface graphics and the DSL words, and predicts code generation of DSL code. This work shows such a potential system that can automatically generate GUI code, but the study only developed a little potential. The current Pix2Code model consists of relatively a few parameters and trains on relatively small data sets. Applying more complex deep learning algorithms to build more complex models and training on larger data sets can significantly improve the quality of code generation. And various regularization method can also further enhance the quality of the generated code. At the same time, the one-hot encoding used in the model does not provide any information on the symbolic relationships and word-embedding models, for example, word2vec[7] may be better. CNN has unique advantages in the field of processing image [8]because its special structure of local weight sharing reduces the complexity of the network. The model performs unsupervised learning without softmax layer. It detects edges, extracts features, and maps a GUI image into a vector containing the learned features. The CNN model detects the size and position of elements in the image, the type of label, the color, spatial association [9], and so on, and places them in the generated vector. In experiments, ResNet50 [10] was applied in visual model to perform feature extraction. The training consumes a lot of calculations, and the accuracy has no obvious improvement. At present, VGGNet is a model that uses unsupervised learning to extract features, and it is still a good choice for accuracy and efficiency. The language model deals with the one-hot encoding corresponding to hierarchical structured DSL textual description associated with the input picture, and performs language modeling. The structure of the text content is shown in Figure 1. Although LSTM has wide application in the language modeling [11], the sequence content has a distinct hierarchy. Therefore, it is obviously not enough to rely solely on the previous word context information. BLSTM can obtain contextual information of previous words and subsequent words of each point in the input sequence [12]. The effect of applying the BLSTM-based language model has been significantly improved, showing excellent potential. Decoder exploits the advantages of CNN in feature extraction combined with the advantages of BLSTM in processing sequence problems to automatically generate code. CNN extracting features combined with LSTM/BLSTM processing sequence problem is a hot research[13][14][15]. This architecture allows the whole pix2code model to be optimized end-to-end gradient descent to predict a token at a time after it has seen both the image as well as the preceding and back tokens in the sequence with optimizing model by BLSTM. Using an BLSTM-based

978-1-5386-7861-9/18/$31.00 ©2018 IEEE

decoder, the architecture performs well in the automatic generation of processing code.

II. UES Bi-directional LSTM

In the programming languages, the children elements or instructions are contained within a block [2]. Where the number of children elements contained in a parent element is variable, it is important to model to capture long-term dependencies to be able to close a block that has been opened. LSTM is both general and effective at capturing long-term temporal dependencies. The current word can be predicted based on the previous word context information of the text when processing the sequence model, but for some tasks only based on the previous word is not very accurate, because the preceding and following words in a sentence are not independent. If the model has access to the context of the previous word context and the words behind it, it is very helpful for language modeling. The basic idea of the BLSTM is to use two LSTMs forward and backward for each training sequence, and both of them are connected to a fully connected layer. This structure allows the output layer to get complete past and future context information for each point in the input sequence. BLSTM has a large number of applications in named entity recognition (NER)[16][17] and sequence labeling [18] [19] [20].

This paper uses BLSTM to optimize the Pix2code model. The framework's language model and decoder both use the BLSTM to replace the original LSTM. In the pix2code paper, this first language model is implemented as a stack of two LSTM layers with 128 cells each, and decoder is implemented as a stack of two LSTM layers with 512 cells each. In the article, the language model in this article uses a stack of two BLSTM with 128 cells each, and our decoder uses a stack of two BLSTM with 512 cells each.

The architecture performs supervised learning during training. Its input is GUI I and a sequence xt. The CNN-based visual model extracting feature from the image I generates a vectorial representation p .The input token xt is encoded by a BLSTM-based language model into an intermediary representation qt. The vision-encoded vector p and the language-encoded vector qt are concatenated into a single feature vector rt which is then fed into a second BLSTM-based model (decoder) decoding the representations learned by both the vision model and the language model. The decoder thus learns to model the relationship between objects present in the input GUI image and the associated tokens present in the DSL code. This architecture based on BLSTM can be expressed mathematically in Equation (1)-(5).

$$p = CNN(I) \tag{1}$$

$$qt = BLSTM(xt) \tag{2}$$

$$rt = (q,pt) \tag{3}$$

$$yt = softmax(BLSTM(rt)) \tag{4}$$

$$xt+1 = yt \tag{5}$$

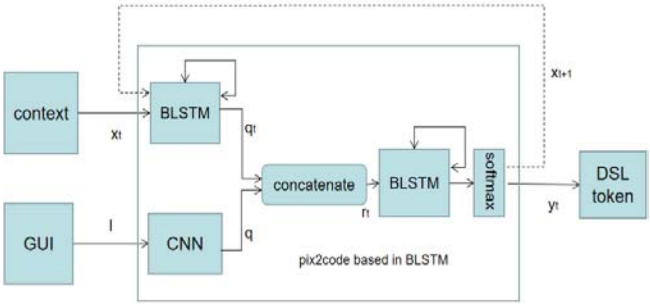

Fig. 1. Training.(BLSTM model)

Figure 1 shows the training process, an overview of the modified pix2code model framework using BLSTM. The model framework uses the BLSTM to reconstruct the original LSTM-based language model and the decoder. When the language model and the decoder process a sequence, it is possible to obtain previous word context information and the subsequent word context of the currently input for modeling and predicting.

III. Dropout In Decoder

Dropout is a very powerful tool that is effective in improving the performance of deep neural networks. The output dimension of the decoder layer is 512, and the output dimension of the language model is 128, so there may be overfitting. Dropout is useful for solving overfitting problems while training complex deep networks. A 25% dropout is set between the two layers of BLSTM at the decoder layer. It can be observed that the performance of the model has been obviously improved, the correct rate of the evaluation set has improved prominently. Subsequent experiments from Hinton have shown that[21][22][23], to some extent, dropout can effectively reduce the occurrence of overfitting and achieve regularization effect .

IV. Initialization

Initializing deep neural network weight distribution has an important impact on model convergence and model quality [24].The parameter is related to the state gradient obtained by backpropagation and the activation value. The saturation of the activation value will cause this layer state gradient information to be 0, and then cause the next layer parameters below to be 0. Randomization initializes parameters into very small random numbers. Small parameters will lead to relatively small gradients in the back propagation, and the neural network with a deep layer will generate gradient dispersion. When the model does not adopt Xavier initialization, there will often be no convergence, and the training accuracy will stop at about 28%. This paper, like the pix2code, only trained 10epoch. The gradient update is very slow in starting training the neural network, so less training epoch may cause that the model not be converge. Xavier normal initialization is set in this architecture. The experiment found that Xavier initialization can make faster the convergence speed of deep networks, greatly reducing the non-convergence.

V. Experimental Results

We use the data set that are provided on the github address of pix2code. The amount of data in the Web-based UI is significantly larger than the other two platforms, so only using the training set and the test set of the Web-based UI. We tried to improve the robustness of the deep learning framework to improve the quality of the system generated code. We don't know the evaluation method of the original paper, and this paper uses the evaluation function of keras. After using BLSTM, the framework does better tested by evaluation method of keras. The original model has a large range of correct rate fluctuations on the Web-based UI test set, and the accuracy is hard to exceed 75%.After modifying the language model and the decoder layer by BLSTM, setting its the learning rate to 0.00025,the correct rate of the test set is increased to about 80% with little fluctuation. Adding a 25% dropout layer between the two layers of BLSTM in the decoder model, the training curve becomes much more steep, but this architecture set has 85% of accuracy about test set, adding the dropout often does not converge, so we set xavier normal initialization. Results show that using BLSTM can obviously enhance the robustness of pix2code.

Github: https://github.com/liuyanbinssr/blstm-pix2code.git

TABLE I. EXPERIMENTAL RESULTS

Dataset type	**Error (%)**	
	Vanilla model	***BLSTM model***
Web-based UI	22.83	14.76

a. Results of test set.

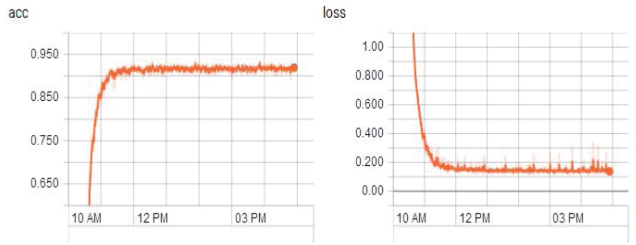

Fig. 2. Vanilla pix2code training loss and accuracy.

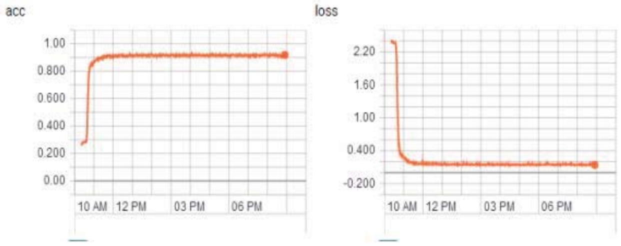

Fig. 3. BLSTM pix2code training loss and accuracy with 0.00025 learning rate.

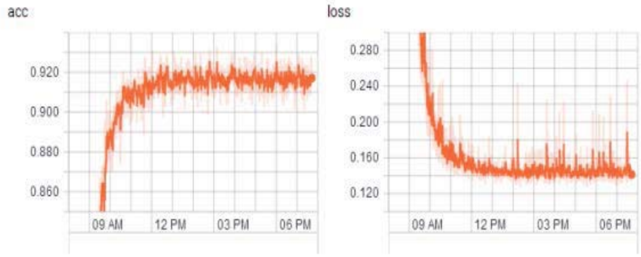

Fig. 4. BLSTM pix2code training loss and accuracy with 0.00025 learning rate and dropout in decoder later.

Training curves: The vanilla model training curve is smoother, and the accuracy of test set fluctuates between 68% and 77%, with relatively large fluctuation range .After using the BLSTM-based model and setting the learning rate to 0.0025, the convergence speed of the model training curve is significantly faster, and the training curve is relatively more stable. This architecture with BLSTM has shown great potential to generate code. The training curve becomes steep after the decoder layer was set 25% dropout, which makes the model not easy to fall into local minimum.

VI. Conclusion and Discussions

The training curve of the Vanilla model is more stable; the accuracy of the training set fluctuates between 90% and 94%; the accuracy of the test set fluctuates between 68% and 77%. The BLSTM model training curve is steeper; the accuracy of the training set is around 92%; the accuracy of the test set fluctuates between 74% and 85%.If only modifying model by BLSTM without adding the dropout layer in decoder, the accuracy of the test set fluctuates between 74% and 80%. It can be seen that accuracy is increased and the float is smaller. HTML/CSS code and DSL code have a rigorous structure. BLSTM can obtain complete past and future context information for each point in the input sequence, predicting the current word based on the previous word context information and the context information of the following words. BLSTM has obvious advantages in dealing with this contextual content with obvious structural and content association issues. The language modeling of the language model and the feature extraction of the visual model have a greater impact on the pix2code model. In this paper, we optimize the model on language modeling, and the feature extraction of the visual model should also have good optimization. For CNN, the relative orientation of the components and the spatial relationship are not very important, and there is no overall constraint judgment ability. Capsules [25] and dynamic routing were proposed by Hinton et al, a new component for deep learning, to better model the hierarchical relationship represented by internal knowledge in the neural network. Moreover. The difference between dynamic routing and maximum pooling is that the exact location information of the entities in the area is not discarded. Generating HTML/CSS code based on the GUI, each token of GUI has an exact location. Although CapsNet does not perform as well as CNN in some big data sets, it also has unique advantages, and there may be good performance on the pix2code model.

Acknowledgment

This study was financially supported by the Special Project of National Science and Technology Cooperation (2014DFB30010), National Natural Science Foundation of China (61501071).

References

[1] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow. “ Deepcoder: Learning to write programs. “Unpublished.

[2] Beltramelli T. “pix2code: Generating Code from a Graphical User Interface Screenshot.”Proc. ACM SIGCHI Symp. Eng. Interact. Comput. Syst,June 2018.

[3] Simonyan K, Zisserman A. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” Computer Science, 2014.

[4] Gers F A, Schmidhuber J, Cummins F. “Learning to forget: continual prediction with LSTM.” IEE Conf Publ,vol.2, 1999, pp.850-855.

[5] Sundermeyer, Martin, R. Schlüter, and H. Ney. “LSTM Neural Networks for Language Modeling.” Annu. Conf. Int. Speech Commun. Assoc., INTERSPEECH ,vol.1,2012,pp.194-197.

[6] Sundermeyer, Martin, and H. Ney. “From feedforward to recurrent LSTM neural networks for language modeling.”IEEE ACM Trans. Audio Speech Lang. Process. ,vol.23,2015,pp. 517-529.

[7] .Mikolov, T., Chen, K., Corrado, G., Dean, J. “Efficient estimation of word representations in vector space.” Computer Science, 2013.

[8] Shan, S. L., Khalil-Hani, M., Radzi, S. A., & Bakhteri, R. “Gender classification: A convolutional neural network approach.”Turk J Electr Eng Comput Sci ,vol.24,2016,pp.1248-1264.

[9] Ozeki, Makoto, and T. Okatani. “Understanding Convolutional Neural Networks in Terms of Category-Level Attributes.”Computer Vision -- ACCV 2014.Springer International Publishing,vol.9004,2015,pp.362-375.

[10] Qin, F., Gao, N., Peng, Y., Wu, Z., Shen, S., & Grudtsin, A.”Fine-grained leukocyte classification with deep residual learning for microscopic images.”Comput. Methods Programs Biomed.,vol.162, pp.243-252,August 2018.

[11] Zaremba, Wojciech, I. Sutskever, and O. Vinyals.”Recurrent Neural Network Regularization.” Eprint Arxiv, September 2014.

[12] Schuster, Mike, and K. K. Paliwal. “Bidirectional recurrent neural networks.” IEEE Trans Signal Process, vol.45,pp.2673-2681,Nov 1997.

[13] Chen, T., Xu, R., He, Y., Wang, X. “Improving sentiment analysis via sentence type classification using bilstm-crf and cnn.”Expert Sys Appl,vol.72,pp.221-230,April 2017.

[14] Zhu, X., Jiang, Y., Yang, S., Wang, X., Li, W., Fu, P., et al.”Deep Residual Text Detection Network for Scene Text.”Proc. Int. Conf. Doc. Anal. Recognit. ,vol.1,2017,pp.807-812.

[15] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., & Saenko, K., et al.”Long-term Recurrent Convolutional Networks for Visual Recognition and Description..” Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit,vol.07-12-June-2015,pp.2625-2634,October 2015.

[16] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.”Neural Architectures for Named Entity Recognition.”Conf. North Am. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol., NAACL HLT - Proc. Conf.,2016,pp.260-270.

[17] Pham, Thai Hoang, and P. Le-Hong.”End-to-End Recurrent Neural Network Models for Vietnamese Named Entity Recognition: Word-Level Vs. Character-Level.”Commun. Comput. Info. Sci., vol.781, 2018,pp.219-232.

[18] Ma X, Hovy E.”End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.”Annu. Meet. Assoc. Comput. Linguist., ACL - Long Pap.,vol.2,2016,pp.1064-1074.

[19] Cross J, Huang L.” Incremental Parsing with Minimal Features Using Bi-Directional LSTM.”Annu. Meet. Assoc. Comput. Linguist., ACL - Short Pap.,2016,pp.32-37.

[20] Yao Y, Huang Z. “Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation”International Conference on Neural Information Processing,vol. 9950 ,2016,pp.345-353.

[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.”Dropout: a simple way to prevent neural networks from overfitting.”J. Mach. Learn. Res., vol.15,2014, pp.1929-1958.

[22] Dahl, George E., T. N. Sainath, and G. E. Hinton. “Improving deep neural networks for LVCSR using rectified linear units and dropout.”ICASSP IEEE Int Conf Acoust Speech Signal Process Proc, pp.8609-8613,October 2013.

[23] Molchanov, Dmitry, A. Ashukha, and D. Vetrov. “Variational Dropout Sparsifies Deep Neural Networks.”Int. Conf. Mach. Learn., ICML,vol.5,2017,pp.3854-3863.

[24] Glorot, Xavier, and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks.”J. Mach. Learn. Res., vol.9,2010,pp.249-256.

[25] Sabour, Sara, N. Frosst, and G. E. Hinton. “Dynamic Routing Between Capsules.”Adv. neural inf. proces. syst,vol.2017-December, 2017, pp.3857-3867.