Protein structure prediction using Deep Learning

Kinh Huy Phan and Dang Thanh Hai*

Faculty of Information Technology,
University of Engineering and Technology,
Vietnam National University, Hanoi

***Abstract*— Understanding Protein secondary structure (PSS) plays a pivotal role for studying proteins' structures and functions. As experimenting for annotating protein structures is both time and money consuming, narrowing the amount of experiments for validating the structures is significantly important. Hence, many efficient Machine Learning methods for predicting PSS have been developed so far, contributed remarkably to the field of biology. In this report, we propose a model based on RNN architecture, which achieved roughly 55% Q8 accuracy and 75% Q3 accuracy**

I. INTRODUCTION

The 3D structure of a protein is determined largely by its amino acid sequence [1]. Unfortunately, predicting the 3D structure from sequences is a complicated and challenging problems for scientists [2]. Understanding protein structure is critically essential, causing the urge to study complex sequence-structure relationships as a topic of broad interests. Protein secondary structure (PSS) is important information for understanding its 3D structure, hence providing significant insights into its functions [3]

Machine learning, especially deep learning with its magnificent performance reported recently, has been applied for tackling many classification problems within a wide range of fields, including Computer Vision, Business analysis, Bioinformatics. Recurrent Neural Networks (RNNs)- based methods have archived outstanding results for sequence labeling problems, generally in other tasks but not protein structure prediction one.

In this report, we present a protein secondary structure prediction model based on a Recurrent Neural Network and Conditional Random Fields. Initial experimental results of our model on benchmark datasets are presented, in comparison with other state-of-the-art related models, demonstrating the potential of our model for prediction protein structures.

Example for input and output (3-state prediction). The former line is protein sequence, the latter is the corresponding conformation class:

- ABABABABCCQQFFFAAAQQAQQ
- HHHHCCCCCEEEECCCHHHHHHC

With H means the alpha helix, E means the beta strand, and C are the Coiled region of the protein secondary structure.

* Contact information: hai.dang@vnu.edu.vn

II. RELATED WORK

As stated above, understanding protein secondary structures is significantly important. Many researchers all over the world have developed plenty of algorithms for predicting PSS. Though, over last 20 years, there have been almost no algorithm can beat the state-of-the-art approach with 80% Q3 accuracy record, firmly held by PSIPRED [4] since 1999. Before that day, most of the predictors implemented was statistic-based, not until 90s of 20th century, when neural networks started to draw computer scientists attention with the predictor developed by Qian Sjnowskiseemed that coudl achieve 62.7% of prediction accuracy [5].

A template-based method for PSS prediction has been proposed by Baldi et al. in 2014 [6], which makes uses solved structures, yielding an amazing accuracy of roughly 93% on the *pdb_full* dataset. Baldi's method, however, is outperformed by PSIPRED when close templates are not avaiable. Cheng et al. [7] proposed a deep learning approach to 3state PSS prediction using a typical deep belief network model, in which each layer is a restricted Boltzmann machine (RBM) and is trained by contrastive divergence in an unsupervised manner. Zhou & Troyanskaya [8] reported another deep learning approach to 8state PSS prediction using a supervised generative stochastic network, which, to the best of our knowledge, may be the best 8state predictor. However, neither Cheng nor Zhou reported a better than 80% of accuracy for 3state PSS prediction.

Recently, in 2016, DeepCNF [9], which is a model based on a Convolutional Neural Network and Conditional Random Fields, has broken the long-hold record of PSIPRED. It can obtain results of 84% Q3 accuracy, and 72% Q8 accuracy.

III. BACKGROUND

A. Artificial Neural Network

Artificial neural Networks (ANNs) are inspired by the biological neural networks that constitute animal brains. Such systems "learn" (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming. A sound example for ANNs application is, in voice recognition, it might learn to identify phonemes represented by those waveform signal, by analyzing sample audio file which are manually labeled. The NNs themself know nothing about Fourier transform, handling Signal and System as well as Digital signal processing.

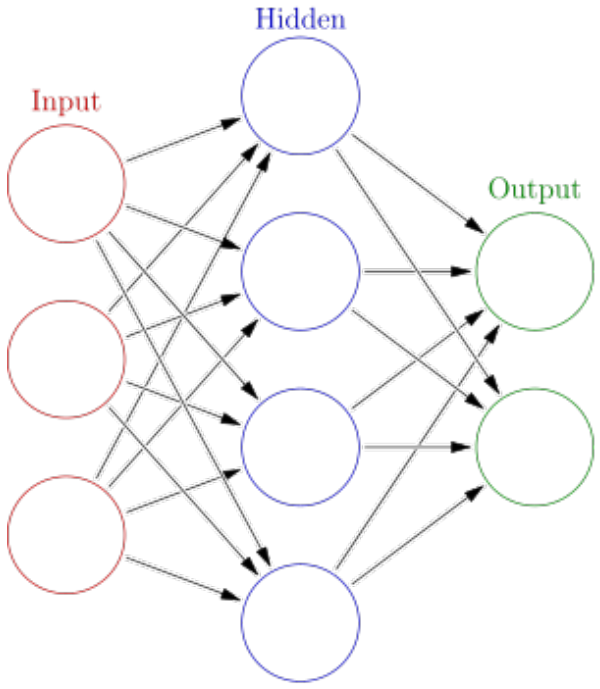

Fig. 1. A simple demonstration of ANN. Each circle is often called *unit* or *neuron*. Each directed arrow, which corresponds to *synapse* in human neuron, represents connection between neurons. Source: wikipedia

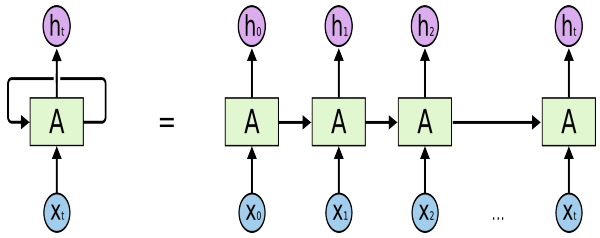

Fig. 2. **Left**: A is an arbitrary type of NN, which was fed with input x, then either pass the information to itself for looping or throw out an output for further processing. **Right**: A is unrolled to a chunk of sub-network. Source: colah

Instead, they evolve their own set of relevant, abstract and useful patterns, features and properties from the learning materials.

B. Recurrent Neural Network

Naturally, human nervous system keep on learning and preserve information consciously and unconsciously. Knowledge obtain from the past decide how we understand current context. We never throw everything away and start thinking from scratch for again. "Your thought have persistance" [10]

RNN was developed to address this issue. Basically, RNN is a network with loops inside, which help it memorize useful information for better understanding in future context.

C. Long Short-Term Memory network

Theoretically, many problems which are related to time-dependence (voice recognition, voice synthesis, sequence labeling, name entity recognition,) can be solved efficiently

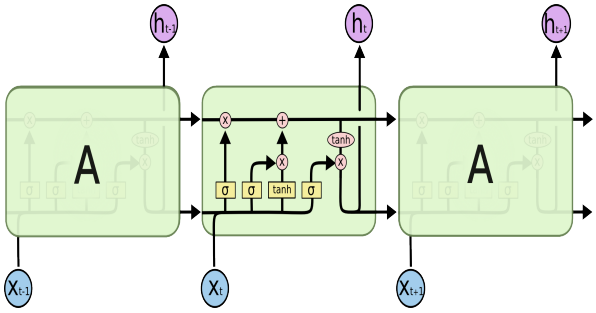

Fig. 3. An example of *LSTM cell*: it takes input from other units, then procession the forget and memorize procedure. Source: colah

with RNN. We just need to implement a very bulky network with large-enough RNN-cell units. In reality, though sometimes we need to persist long-time information, in most cases, we only consider some recent information, and discard many other useless one. This problem make traditional RNN become a waste of not only space for storing parameters, but also time for training the networks.

Long-short-term memory (LSTM) [11] was developed to address this issue of RNN. The basic idea of LSTM is instead of storing all information from the past like traditional RNN, it learns to sometimes forget some useless information, and decide which is important to *memorize*.

D. Bidirectional Recurrent Neural network

Though LSTM unit is effective for learning useful information in the past, in reality, there are many cases where future information make a huge contribution for understanding the present context.

For example: Given a problem: predict which possessive pronoun is suitable to fill in the blank: "We are talking about a national hero of Vietnam. And ____ name is Ho Chi Minh". There is no information about gender in the past, but after the blank, we can see the word "Ho Chi Minh". Since Mr. Minh is a well-known person in Vietnam and everyone knows Mr. Minh is a gentleman, we can easily fill *his* into the blank.

The basic idea of Bi-LSTM [12] is to present each training sequence to two separate recurrent networks: forwards and backwards, both of which are connected to the same output layer. This means that for every point in a given sequence, the Bi-LSTM has complete, sequential information about all points before and after it. [3]

E. Conditional Random Fields

CRF [13] is a method for labeling and segmenting structured data, such as sequences, trees,... "The underlying idea is defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. [14]" To put it another way, let say we have a sequence that need to be tagged, CRF will utilize labels from tagged part of the sequence for tagging the current position. Defining feature functions, also known as feature engineering, which represent relationship between states, is the keypoint to the performance of CRF

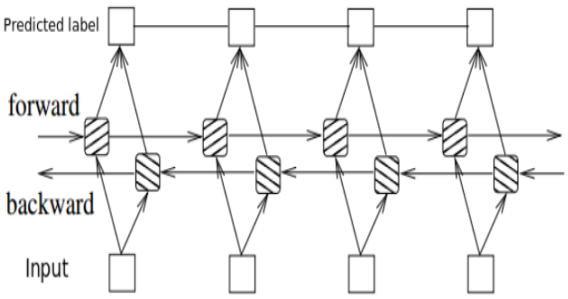

Fig. 4. A demonstration of Bi-LSTM structure with CRF layer on top of it. Each label was labeled based on information from *forward-backward procedure* from the Bi-LSTM, by the CRF layer with the information from previous tagged label(s). Source: [5]

1) Feature functions: To model those types of relationship, feature functions are used (For simplicity, We will consider linear-chain CRF, which mean current features depend on only its one previous label). In a CRF, each feature function "f(s, i, li, li-1)" is a function with respect to:

- A sentence *s*
- Position *i* of a word in the sentence
- Label l_{i} of the current word
- Label $l_{\text{i-1}}$ of the previous word

An example of feature function: if a word *W* is followed by the word very, there will be high probability that *W* is an adjective. Therefore we can easily tag *W* with *ADJ* tag.

There might be many feature functions for modeling relationship. Each of them might correspond to a specific weight to determine how significant this relationship is.

2) Label scoring: Given a sentence *s*, we define a score function demonstrating the score of label sequence *l* of *s*:

$$score(l \mid s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_{\text{j}} * f_{\text{j}}(s, i, l_{\text{i}}, l_{\text{i-1}})$$

which will iterate through each feature function *j* and sum over the score of each position *i* given label l_{i} . λ is the trainable parameter weight mentioned above. We will learn the best λ from data.

The scores then are converted into probabilities by exponentiating and normalizing:

$$P(l \mid s) = \frac{\exp[score(l \mid s)]}{\sum_k \exp[score(l_{\text{k}} \mid s)]}$$

IV. Materials and Methods

A. Datasets

Here we used the CullPDB [18] (8832 sequences in total, 6300 for training, 700 for validating and 1832 for testing) and CB513[17] (513 sequences) was used for testing.

Information about primary structure and its corresponding secondary structure was extracted from Protein Data Bank file by using Dictionary of Secondary Structure Protein [15].

B. Labels and features

For Q8 accuracy, there are 8 labels, represent 8 type of states of secondary structure, which are: *G* (3-turn helix, with 3 residues minimum length), *H* (4-turn helix, also known as α helix, with 3 residues minimum length), *I* (5-turn helix, also known as π helix, with 5 residues minimum length), *T* (hydrogen bonded turn), *E* (extended strand in parallel and anti-parallel β-sheet conformation, with 2 residues minimum length), *B* (residue in isolated β-bridge), *S* (bend), *C* (coil) Source: Kabsch W, Sander C [15] More information of protein secondary structure can also be found at [15]

For Q3, there are only 3 labels, which were reduce by grouping these above labels into groups. There are many ways for reduction, here was one of them [16]:

- E and B to E
- G and H to H
- Rest to C

C. Amino acid embeddings (AAE)

In natural language processing, a word embedding is an algorithm to learn a high-dimensional dense vector representation for words from a very large textual corpus (i.e. training corpus) with billions of words. It works based on the basic idea that the meaning of a word, respect to a particular context, is affected by surrounding words and some its own features.

In this work, each amino acid is embedded by concatenating some real-values vector, based on its physical and chemical properties, which are:

- The pre-trained AAindex - Amino Acid Index Database [19]
- Hydrophobicity
- Solvent Accessible Surface (predicted from raw sequence by RVP-net [20]
  And some other properties which shown ineffectiveness in improving performance of the model.

D. Network architecture

The network architecture of our PSS prediction model comprises a Conditional Random Fields on top of a Bidirectional LSTM network (Fig 4). In addition to the past input features and protein sequence level tag information used in a LSTM-CRF model, a BILSTM-CRF model can use the future input features, therefore helping the model to capture as much information from primary sequence as possible.

E. Evaluation

Here, only the Q3 and Q8 accuracy are considered. Since the accuracy is currently still outperformed by some other methods, We will not extend the evaluating metric to other score for time saving. In the future, Recall, Precision, F1_score for each class (state of residue in this problem) will be presented.

The Q3 (Q8) accuracy is defined as the percentage of residues for which the predicted secondary structures are correct over the sequence.

TABLE I

Q8 ACCURACY OF OUR MODEL IN COMPARISON WITH OTHER STATE-OF-THE-ART

Methods	SSpro	ICML2014	RaptorX-SS8	Deep-CNF	My method
CullPDB	66.6	72.1	69.7	**75.2**	55.3
CB513	63.5	64.4	64.9	**68.3**	53.9
CASP10	64.9	_	64.8	**71.8**	_
CASP11	65.6	_	65.1	**72.3**	_
CAMEO	63.5	_	66.2	**72.1**	_

TABLE II

Q3 ACCURACY OF OUR MODEL IN COMPARISON WITH OTHER STATE-OF-THE-ART

	Methods	SSpro	SPINE-X	RaptorX-SS8	Deep-CNF	My method
Q3 accuracy	CullPDB	79.5	81.7	81.2	85.4	73.4
	CB513	78.5	78.9	78.3	82.3	71.5
	CASP10	78.5	80.7	78.9	84.4	_
	CASP11	77.6	79.3	79.1	84.7	_
	CAMEO	77.5	80	79.4	84.5	_

V. TRAINING AND TESTING

Training and testing had been separated in 2 phases (The procedure inspired by DeepCNF [9] with 50 LSTM units for a single forward/backward phase, 0.5 dropout rate, batch size of 32. Adam optimizer is used, with starting learning rate 0.05. The model was trained through 100 epoches. These hyperparameters were chosen via grid search and manually selected for best result. The grid size quite small, so further tuning is necessary.

- **Phase 1**: Feed the model with 6300 training sequences from CullPDB, then validate on 700 sequences and perform testing with the rest of CullPDB data. The results are report completely dependently to the phase 2
- **Phase 2**: Feed the whole CullPDB dataset into the model for training and validating, then evaluate it with CB513 dataset (no testing on CullPDB anymore). Since in these above 2 phases, the test sets are absolutely disjoint, no cheating in training is presented here.
  Overall accuracy is mean accuracy of these 2 phases.

The training took about 3 hours, on Core i7-2780QM, 8GB of RAM, CPU only, using keras 2.1.5 with tensorflow 1.2.0 backend, numpy 1.14.1, on python 3.5.2, Ubuntu 16.04 operating system.

VI. EXPERIMENTAL RESULTS

Initial experimental performance results of our model in comparison with other state-of-the-art related models are detailed in Table I and Table II. Though DeepCNF outperforms other methods as demonstated in such two tables, SSpro approach (with templates) can beat it on CullPDB, CB513, CASP10 for Q8 and CullPDB, CB513 for Q3.

VII. CONCLUSION AND DISCUSSION

A. Conclusion

Here We present a model for predicting protein structure interaction,which based on Bidirectional Long-short-term-memory with Conditional Random Fields layer on top of it.

The overall performance of my method, which was outperformed by roughly 10% compared to state-of-the-art methods, is not quite promising, but still have many rooms for improvement, especially for Q3 prediction.

B. Discussion

There are some possible reasons which can account for the worse performance than expection of our model, for example: that our data preparation is still naive, which is significant for this type of problem. Other related methods, vice versa, employs elaborate data [pre-]processing.

VIII. FUTURE WORK

Tackling every possible causes that make our model performs worse than the best models (as mentioned above) are the most significant and obvious perspectives. Increasing the dataset for training and extends the feature set for each amino acid are also worth being considered.

ACKNOWLEDGMENTS

We would like to sincerely thank Dr. Do Duc Dong (from Department of Computational Sciences and Engineering, VNUH UET) for useful comments and the approval for publication of our work as a technical report. We also thank Mr. Trac Quang Thinh (BSc.) for his great assist to our work.

REFERENCES

[1] Baker, D. & Sali, A. Protein structure prediction and structural genomics. Science 294, 9396 (2001).

[2] Dill, K. A. & MacCallum, J. L. The protein-folding problem, 50 years on. Science 338, 10421046 (2012).

[3] Lin, K., Simossis, V. A., Taylor, W. R. & Heringa, J. A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics 21, 152159 (2005).

[4] McGuffin, L. J., Bryson, K., Jones, D. T. (2000). The PSIPRED protein structure prediction server. Bioinformatics 16(4), 404-405.

[5] K. K Senapati, G. Sahoo and D. Bhaumik Algorithm for Predicting Protein Secondary Structure.

[6] Pollastri, G., Przybylski, D., Rost, B. & Baldi, P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins: Struct. Funct. Bioinform. 47, 228235 (2002).

[7] Spencer, M., Eickholt, J. & Cheng, J. A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction. IEEE/ACM Trans. Comput. Biol. Bioinform. 12, 103112 (2015).

[8] Zhou, J. & Troyanskaya, O. Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction. Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Proceedings 32, 745-753 (2014).

[9] Wang, S. et al. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. Sci. Rep. 6, 18962; doi: 10.1038/srep18962 (2016).

[10] Colah - Understanding LSTM: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11] Sepp Hochreiter; Jrgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 17351780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.

[12] Alex Gravesa, Jrgen Schmidhuberab - Framewise phoneme classification with bidirectional LSTM and other neural network architectures - Neural Networks,Volume 18, Issues 56, JulyAugust 2005, Pages 602-610, doi.org/10.1016/j.neunet.2005.06.042

[13] John Lafferty, Andrew McCallum, Fernando C.N. Pereira - University of Pennsylvania, pereira@cis.upenn.eduProceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), pages 282-289.

[14] Edwin Chen - Conditional Random Fields: http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

[15] Kabsch W, Sander C (1983). "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". Biopolymers. 22 (12): 2577637. doi:10.1002/bip.360221211. PMID 6667333.

[16] Ralf Zimmer - LMU Institut for Informatik, Lehrstuhl fr Bioinformatik, WS 2010/2011: Bioinformatics Programming Course via https://drupal.bio.ifi.lmu.de/nfs/webfm/Lehre

[17] Zikrija Avdagic, Prof.Dr.Sci. et al. - Artificial Intelligence in Prediction of Secondary Protein Structure Using CB513 Database. Summit on Translat Bioinforma. 2009; 2009: 15. Published online 2009 Mar 1. PMCID: PMC3041573. PMID: 21347158

[18] Culled Protein Data Bank - CullPDB at http://www.princeton.edu/ jzthree/datasets/ICML2014/

[19] Kawashima S1, Ogata H, Kanehisa M. - AAindex: Amino Acid Index Database. Nucleic Acids Res. 1999 Jan 1;27(1):368-9.

[20] Ahmad S1, Gromiha MM, Sarai A. - RVP-net: online prediction of real valued accessible surface area of proteins from single sequences. Bioinformatics. 2003 Sep 22;19(14):1849-51.