Convolutional LSTM Networks for Subcellular Localization of Proteins: Paper Review

Nipun Agarwala, Oliver Bear Don't Walk, David Cohn, Yuki Inoue, Axel Sly

November 9, 2016

1 Background

In recent years, deep learning has gained significant traction in many fields. This paper (Sonderby et al.) describes an application of neural networks in biology, specifically the sub-cellular localization of proteins using protein sequence analysis. Previous work in this area of research has utilized basic neural network models and support vector machines (SVM), in conjunction with sophisticated methods for feature extraction incorporating expert domain knowledge. Furthermore, none of these previous models took into account sequences of varying length.

Sonderby et al. construct an LSTM model that will generalize the protein sorting problem, without the aforementioned constraints, while maintaining nearly as good, if not better, performance. Furthermore, the authors visualize the lower layers of the network using amino acid sequence logos to show how the model filters act as motif detectors. This technique is analogous to visualizing convolutional filters in a Convolutional Neural Network (CNN).

2 Experimental Review

2.1 Biological Review

The biological premise behind the work of Sonderby et al. is that the sequence of amino acids in a protein can be used to predict its localization; in turn, protein localization has interested biologists because the location of a protein can have a significant influence in dictating its function. Sonderby et al.'s work was generally sound from a biological perspective. The authors' reduction of each protein structure to the 1000 amino acids that are associated with the area around the N and C termini is mostly valid, as many target peptides, which are instrumental in directing a protein to a particular organelle in the cell, are found in either termini.

Nevertheless, the authors' approach could fail to account for the effects of some protein signal patches in determining protein localization, as the structure of these patches is not necessarily confined to either terminus. However, if the amino acid residues associated with these signal patches are, in fact, included in the 1000 analyzed amino acids, an LSTM model is seemingly the perfect approach to handle their effect; signal patches are composed of amino acid residues that are sequentially far apart (in terms of primary structure), but through tertiary structure folding, are brought close together to influence localization. Therefore, an LSTM model can learn the long-term dependencies of the various components of the signal patch on each other in order to collectively capture their effect.

In critically examining the results of Table 2 and Figure 5 through a biological lens, Sonderby et al.'s LSTM model captures the effect of signal peptides (N-terminus sequences) across multiple organelles involved in the secretory pathway very well. In particular, the presence of signal peptides in ER, Golgi, extracellular and lysosomal proteins are clearly shown in Figure 5. Furthermore, the effects of N-terminus presequences and transit peptides, which influence localization to the mitochondria and cholorplasts respectively, can also be readily identified.

Nevertheless, the model does not appear to identify the effect of C-terminus target peptides well, particularly ER KDEL sequences and peroxisome targeting signal 1 (PTS1) sequences. This fact, coupled with relatively small ER and peroxisome sample sizes, likely led to low localization classification accuracy (66.7% for ER proteins and 56.3% for peroxisome proteins) relative to the overall performance.

Furthermore, the model achieved only 81.5% accuracy on Nuclear proteins, which appears to be largely due to missed nuclear localization sequences (NLS); proteins that lack a signal peptide and a specialized signal sequence (such as NLS), by default, end up in the Cytoplasm, while proteins without a signal peptide, but with a NLS, end up in the nucleus. As such, the misclassification of 27 nuclear proteins as cytoplasmic proteins is likely due to the LSTM model failing to identify the NLS. This finding constitutes an additional downside to the authors' protein size reduction approach, as NLS can be found anywhere in the primary sequence, including possibly the "middle regions" of the proteins that were removed by the authors. Nevertheless, this result also highlights a common problem for experts in deep learning, in having to weigh computational considerations with the possibility of decreased model performance.

2.2 Neural Net Topology Review

One of the hardest aspects in using neural nets for any application is hyperparameter tuning. The flexibility afforded by a large number of parameters in neural nets allows them to handle many different types of data sets, but also makes them hard to train effectively. Further complicating matters, there is no analytical method to figure out the hyperparameter. Therefore, exploring different hyperparameter configurations is a necessary part of any paper, and being able to find an optimal topology can seriously affect the results of the application.

Unfortunately, this paper significantly lacks discussions on topology exploration. The paper does discuss in detail about the advantages of their neural net structure over existing structures in Section 2.2, but specific details regarding neural net, such as the number of LSTM units, are not mentioned at all. Section 2.6 does mention some hyperparameters, such as learning rate and convolutional layer filter sizes, but the values are only briefly mentioned. There is no mention of exploring the hyperparameter space at all, leaving readers to question where the authors obtained those values.

Another problem with this paper regarding neural net topology is that there is no diagram detailing their final network. The closest diagram that is included in the paper is Figure 3, which only provides a sketch of their neural network at a very high level. What is needed is essentially a diagram with the information contained in Section 2.6. For example, instead of writing out the overall structure (i.e. "The network architecture is a 1D convolutional layer followed by an LSTM layer, a fully connected layer and a final softmax layer."), there should be a diagram of this structure. Such a diagram would have made it easier for the readers to digest the information, as neural networks are very visual objects.

2.3 Data Presentation

Many visualizations were produced in order to enable the interpretation of the results from the LSTM network. The authors claim that filter weights denote relative importance of the scores to each other. This might not be the case, namely because this claim assumes that the features are trained relative to each other. However, when performing backpropogation, the weights are directly dependent on the trained output labels, not each other.

Later in the discussion section, it is claimed that protein sequences alone yield biologically relevant information, especially for visualizations, which is not always true. CNNs can give relevant biological information regarding which Motifs and/or proteins are important in a certain expression scheme without needing multiple sequences. We see in the DeMo Dashboard paper[1] that little sequences of DNA can help us extract local biological information. Though this paper uses protein sequences, the same principle applies. Nevertheless, they visualize filters to great effect, which

[1] https://128.84.21.199/pdf/1608.03644v3.pdf

gives the reader and potentially other researchers, great insight into how this technique can help learn new motifs for protein features. Though these visualizations may not be accurate, as noted in the previous paragraph, the idea and interpretation of using such visualizations is interesting.

Lastly, the LSTM importance scores and weights are visualized using a simple graph representation and t-SNE respectively. The importance scores are very useful to visualize, particularly for a researcher in the field, namely because it makes it easier to compare and contrast different proteins in an accurate manner. The t-SNE representations, on the other hand, presented reasonably accurate clusters, like those for many of the secretory organelles. But such visualizations would not be as useful due to severe dimensionality reduction problems. More specifically, when mapping a potentially 1000 dimension vector onto 2-dimensions, information is bound to be lost. This is obvious from the fact that some groups of clusters, like the organelles with signal peptides, despite having similar properties, are nevertheless different enough not be so close to one another (i.e. a group of plasma membrane proteins in the A-LSTM plot that are separated from the rest of the plasma membrane proteins etc.). Hence, more information can be extracted from other representations.

3 Significance and Novelty

The authors present an LSTM network with convolutions to predict subcellular localization, achieving better results than other models which also rely solely on protein sequence (Table 1). When compared to models which use protein sequence and metadata, Sonderby et al.'s model outperforms all but one model (SherLoc 2). The significance of these findings is that Sonderby et al.'s model is biologically naive, yet can still achieve state-of-the-art results. Furthermore, even though other models achieve higher accuracy percentages, these models require some degree of protein annotation, while Sonderby et al.'s model does not require any annotation.

Nevertheless, using neural networks to achieve strong results is nothing new, so the authors also provide a way of peering into the deep learning black box, in order to provide some understanding of why the model behaves the way it does. This allows those with domain knowledge to determine for themselves whether the model is picking up on expected or unexpected characteristics. The authors provide position context importance, as well as sequence importance for making classifications. These are important plots to provide in research papers, so that the neural network architecture can be made more accessible.

4 Comparison to Related Literature

The authors benchmark their performance to MultiLoc1/2 and SherLoc2. MultiLoc1 was implemented in 2006 by Hoglund et al as an SVM-based approach, which makes use of features like overall amino acid composition and the presence of known sorting signals. MultiLoc2 is an improvement over MultiLoc1 and was implemented in 2009 by Blum et al. In order to get improved performance, MultiLoc1 was extended to include phylogenetic (PhyloLoc) profiles and GO terms (GOLoc). SherLoc2 was also implemented in 2009 by Briesemeister et al., and incorporates the same features as MultiLoc2, but also makes use of the user's background knowledge of the protein, as the user can provide a short description of the protein. SherLoc2 is also an SVM-based model. As stated in the previous section, Sonderby et al.'s R-LSTM ensemble model outperforms MultiLoc1, both with and without phylogenetic and GO term metadata. Furthermore, Sonderby et al.'s model outperforms all but one model (SherLoc2) that utilizes both protein sequence data and metadata.