PROTEIN SECONDARY STRUCTURE PREDICTION USING DEEP CONVOLUTIONAL NEURAL FIELDS

1. Introduction and Motivation

In this project, we are implementing Protein secondary structure prediction by the use of DeepCNF for predicting the secondary structure of proteins.The motivation behind this is that understanding the structure of proteins is essential for many areas of biotechnology and drug discovery, and has important applications in fields such as medicine, agriculture, and environmental science. Proteins consists of amino acid chains linked by peptide bonds, and the flexibility of these chains enables multiple conformations to arise.

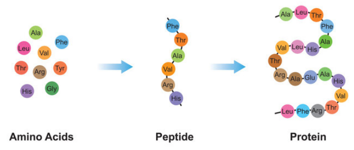


Figure 1. Image Caption

There are four levels of protein structure, the two relevant ones in our project are:

1. Primary structure: It refers to simply the linear sequence of amino acids that make up the protein

2. Secondary structure: The secondary structure of a protein refers to the local spatial arrangement of amino acids in the protein, including alpha helices, beta sheets, and coils. This is what we are interested in predicting. Depending on the database, the structures are divided into 3 classes or 8 classes of amino acids. SS prediction is usually evaluated by Q3 or Q8 accuracy, which measures the percent of residues for which a 3-state or 8-state secondary structure is correctly predicted.

3. The eight classes of protein secondary structure include three types of helices: alpha-helix, 3-10-helix, and pi-helix; three types of beta-sheets: beta-strand, beta-turn, and beta-bridges; and two other classes: bend and loop. The classification is based on the backbone dihedral angles of amino acid residues and the spatial arrangement of neighboring residues.

8-class	3-class	Name
H	H	α-helix
E	E	β-strand
L	C	loop or irregular
T	C	β-turn
S	C	bend
G	H	3_{10}-helix
B	E	β-bridge
I	C	π-helix

Figure 2. Eight Classes

1.1. Dataset

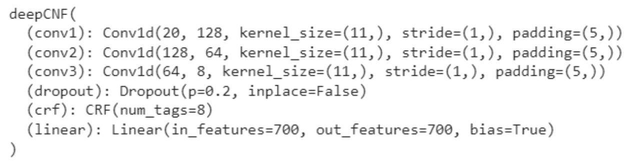
In the original research paper, the authors trained their model on the CullPDB53 dataset, which consisted of 6125 proteins. However, this dataset was not readily available for subsequent studies. As an alternative, we attempted to cull the proteins from the PISCES database, but this method failed due to the lack of access to the PDB IDs of these proteins.

As a result, we opted to use the CASP10 dataset for training and testing. It is worth noting that the CASP10 dataset only contained labeled data for 123 proteins, which is much smaller than the original CullPDB53 dataset used in the original research paper.

The CASP10 dataset that we used was provided in a dictionary format, with two keys: features and labels. The features key had a shape of (123,700,43), where the first dimension corresponds to the number of proteins in the dataset, the second dimension represents the sequence length of each protein being considered, and the third dimension represents the different features of each amino acid in the sequence. Out of the 43 features, only the top 20 features were selected which correspond to one-hot encoded forms of the amino acids. On the other hand, the label's key had a shape of (123,700,9), with the same first two dimensions as the features key and the third dimension representing the different features of each amino acid's secondary structure. Out of these 9 features, only the top 8 features were selected which correspond to the one-hot encoded form of the secondary structure of the amino acids. The last feature represented the padding sequence which is not relevant to this study. Therefore, the final input of the model consists of sequences of amino acids, and the final output corresponds to the sequences of secondary structures relating to the amino acids.

1.2. Model Architecture

Our models have the following architecture in general:(refer to Figure 2) These models have two parts: the



```
deepCNF(
  (conv1): Conv1d(20, 128, kernel_size=(11,), stride=(1,), padding=(5,))
  (conv2): Conv1d(128, 64, kernel_size=(11,), stride=(1,), padding=(5,))
  (conv3): Conv1d(64, 8, kernel_size=(11,), stride=(1,), padding=(5,))
  (dropout): Dropout(p=0.2, inplace=False)
  (crf): CRF(num_tags=8)
  (linear): Linear(in_features=700, out_features=700, bias=True)
)
```

Figure 3. Model Architecture

Conditional Random Fields (CRF) part and the convolutional neural network part. The CRF part has a top layer and a label layer, while the convolutional neural network part covers the input to the top layer.

The only difference between the models we have implemented is the number of layers in the convolutional neural network part. For example, in Figure 2, we have used three one-dimensional convolutional layers in the convolutional neural network part. We have repeated the same experiments with one, five, and seven layers in the convolutional neural network part. For the CRF part, we have used the pytorch-crf package in PyTorch.

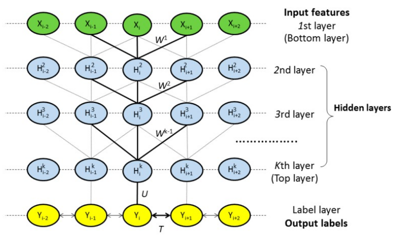


Figure 4. Model Architecture

The CRF layer is responsible for decoding the output of the DCNN module into a sequence of predicted labels. The top layer of the CRF module acts as an intermediate layer between the DCNN module and the label layer. The label layer consists of a set of output nodes, each representing a different label in the sequence. During training, the CRF module learns to assign appropriate labels to each input sequence.

The DCNN module, on the other hand, is responsible for extracting relevant features from the input sequences. It consists of a series of convolutional layers that perform feature extraction and pooling layers that downsample the output of the convolutional layers. The output of the DCNN module is then fed into the CRF module for decoding.

1.3. Metric Used

We have used Q8 accuracy as a metric for our model. In secondary protein structure prediction, Q8 accuracy refers to the accuracy of predicting the eight-state secondary structure classification of amino acid residues. The eight states are helix (H), extended strand (E), coil (C), and five types of turns (T). Q8 accuracy is calculated by comparing the predicted secondary structure sequence with the true secondary structure sequence on a per-residue basis and counting the number of correctly predicted residues.

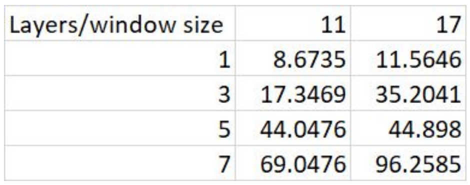
1.4. Training

To train our models for protein secondary structure prediction using DeepCNF, we followed a specific methodology.

We implemented different architectures of the DCNN part by varying the number of layers as (1, 3, 5, 7) and the window size (11, 17). The dropout layer is applied after the convolutional layer to prevent overfitting.

The model was trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and categorical crossentropy loss was used as the loss function. The model was trained for 20 epochs with a batch size of 49. The training data was fed to the model in batches. For each batch, the inputs and targets were loaded and converted to PyTorch tensors. The optimizer was then zeroed, and the model's output was computed. The loss was computed between the predicted and target labels, and the gradients were backpropagated through the model. Finally, the optimizer was stepped, and the running loss and accuracy were computed.

1.5. Result



Layers/window size	11	17
1	8.6735	11.5646
3	17.3469	35.2041
5	44.0476	44.898
7	69.0476	96.2585

Figure 5. Q8 accuracy

In our implementation, we selected the size of the sliding window to be 11, which was the same as that used in the original paper. The reason for selecting this window size was that the average length of an alpha helix is around eleven residues and that of a beta-strand is around six residues. Interestingly, we noticed that increasing the window size to 17 also led to an improvement in the Q8 accuracy of our model.

In addition to the window size, we also experimented with varying the number of layers in our model. We found that the Q8 accuracy gradually improved as we increased the number of layers. This might be because deeper networks are able to capture more complex features and learn hierarchical representations of the input data, which can ultimately lead to improved accuracy.

1.6. Conclusion and Future Works

In conclusion, our experiments show that increasing the window size and number of layers in a deep learning model for protein secondary structure prediction can lead to improved accuracy. Specifically, we found that a window size of 17 and a deeper network architecture resulted in the highest Q8 accuracy on our dataset. These results suggest that further exploration of larger window sizes and deeper networks may lead to even better performance on this task. Overall, deep learning approaches show promise for improving the accuracy of protein secondary structure prediction and have important applications in protein engineering and drug discovery.

A further improvement could be to include PSSM features along with amino acid features.PSSM (Position-Specific Scoring Matrix) is a matrix that represents the likelihood of finding a particular amino acid at a specific position in a protein sequence. PSSM is calculated using multiple sequence alignments of homologous proteins, and it can provide valuable information about the conservation and variability of amino acids at different positions in the sequence.

In the context of secondary structure prediction, PSSM can be used as an additional feature to improve the accuracy of the prediction. PSSM contains information about the evolutionary conservation of amino acids, which can be indicative of their structural and functional importance. By incorporating PSSM as a feature in a deep learning model for secondary structure prediction, the model can effectively capture the evolutionary information and use it to make more accurate predictions.

1.7. Inference

Given the amino acid sequence *MKLTASSVDC*, the corresponding predicted secondary structure labels are 3, 3, 3, 3, 3, 3, 3, 3, 3, and 6. This prediction was made using a deep learning model trained on the CASP10 dataset. The label ”3” indicates an alpha helix structure, while the label ”6” corresponds to a coil or loop structure. This prediction can be used to infer the likely three-dimensional structure of the protein, which can provide insight into its function and potential interactions with other molecules.

1.8. Summary

This project focuses on the implementation of a Protein Secondary Structure Prediction model using deep learning techniques. The key deep learning part of the model is the combination of Conditional Random Fields (CRF) and Deep Convolutional Neural Networks (DCNN). The DCNN module is responsible for feature extraction from the input amino acid sequences, consisting of convolutional layers and pooling layers to downsample the output. Meanwhile, the CRF layer decodes the DCNN output into a sequence of predicted labels, representing the secondary structure of the protein. The input of the model includes the one-hot encoded amino acid sequences, and the output corresponds to the sequences of secondary structures. The number of layers in the DCNN module can be varied, while the CRF layer is implemented using the pytorch-crf package in PyTorch. The model is trained and tested on the CASP10 dataset, which includes labeled data for 123 proteins. The accuracy of the model is evaluated using Q8 accuracy, measuring the percent of correctly predicted residues for 8-state secondary structures. The successful implementation of this model has important applications in biotechnology, drug discovery, medicine, agriculture, and environmental science, where understanding the structure of proteins is crucial.

For more information on protein structure prediction, see [1], [2], [3] and [4].

References

[1] Wikipedia. *Protein structure prediction*. https://en.wikipedia.org/wiki/Protein_structure_prediction 3

[2] Wikipedia. *Protein secondary structure*. https://en.wikipedia.org/wiki/Protein_secondary_structure 3

[3] 3

Pytorch CRF Implementation. *CRF*. https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html

[4] Wang, S. et al. *Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields*. https://arxiv.org/pdf/1512.00843.pdf 3