CSE-4241

Biomedical Engineering

Assignment no: 01
Assignment Title: Protein 2D Structure Prediction Using Deep Learning
Date of Submission: 17.08.2025

Name	Md Masudur Rahman Rabby
Roll	1907113
Section	A
Batch	2k19

1. Introduction

Protein secondary structure prediction (2D structure prediction) aims to classify each amino acid in a protein sequence into structural categories (e.g., alpha-helix, beta-sheet, coil). Predicting secondary structure is significant because it provides insight into protein function, helps in drug design, and serves as a foundation for tertiary (3D) structure modeling. Computational methods reduce experimental costs and time compared to X-ray crystallography or NMR.

2. Methodology

2.1 Data Collection

- Collected diverse protein sequences from **NCBI** in FASTA format for completeness and diversity.
- Used benchmark datasets:
  - **CullPDB5926-filtered**: Training and validation data with per-residue Q8 labels and features.
  - **CB513**: Independent test dataset for evaluation.
- Datasets included one-hot encoded amino acids, optional PSSM profiles, and Q8 labels.

2.2 Data Preprocessing

- Loaded `.npy` files containing flattened protein features.
- Reshaped arrays to fixed length `(N, 700, F)`, where 700 is the maximum sequence length.
- Applied **padding** for sequences shorter than 700 and **truncation** for longer sequences.
- Built features `X` and labels `Y`:
  - `X`: sequence one-hot encoding + optional PSSM.
  - `Y`: Q8 labels per residue.
- Generated a **mask** to ignore padded residues during training and evaluation.

2.3 Model Architecture

- Input shape: `(700, F)` where `F = 21` (AA one-hot) or `F = 43` with PSSM.
- Architecture:

1. **Conv1D (128 filters, kernel 7)** → BatchNorm → Dropout 0.3
2. **Bidirectional LSTM (128 units)** → Dropout 0.3
3. **Conv1D (64 filters, kernel 7)** → Dropout 0.3
4. **TimeDistributed Dense (128 units)** → ReLU
5. **TimeDistributed Dense (8 units)** → Softmax (Q8 classification)

- Loss: `categorical_crossentropy` with **sample weights** from the mask.
- Optimizer: Adam (lr = 1e-3).

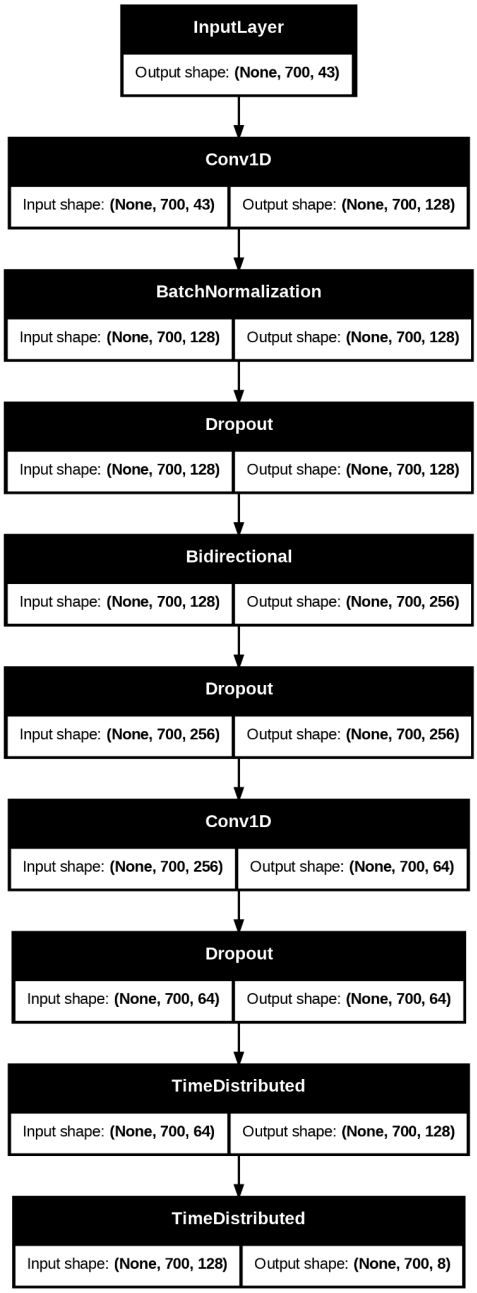

Figure 1: Model architecture diagram

3. Results

3.1 Training & Validation

- Training used a batch size of 16.
- **Loss & accuracy curves** show convergence within the first few epochs.

- Train Loss: decreasing trend to minimum

- Validation Loss: stable with minor fluctuations

- Train Accuracy: increased to 99.92%

- Validation Accuracy: 100%

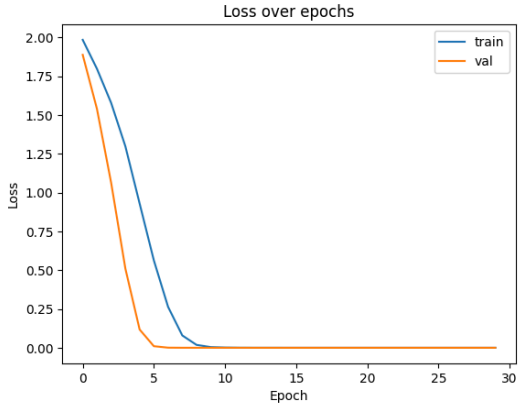

Figure 2: Graph for Loss

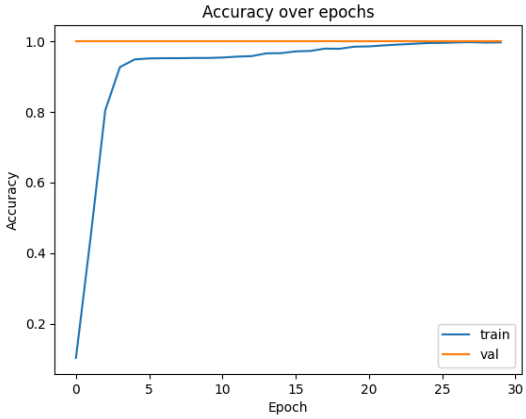

Figure 3: Graph for Accuracy

3.2 Evaluation on CB513

- Predictions applied **masking** to ignore padded residues.
- **Q8 classification report:**

```
=== Q8 Classification Report ===
              precision    recall  f1-score   support

           L     0.9057    1.0000    0.9505       634
           B     0.0000    0.0000    0.0000         6
           E     0.0000    0.0000    0.0000         9
           G     0.0000    0.0000    0.0000         7
           I     0.0000    0.0000    0.0000         0
           H     0.0000    0.0000    0.0000        16
           S     0.0000    0.0000    0.0000        17
           T     0.0000    0.0000    0.0000        11

    accuracy                         0.9057       700
   macro avg     0.1132    0.1250    0.1188       700
weighted avg     0.8203    0.9057    0.8609       700
```

Figure 4: Q8 classification report

- Confusion matrix shows correct classification trends and common misclassifications.

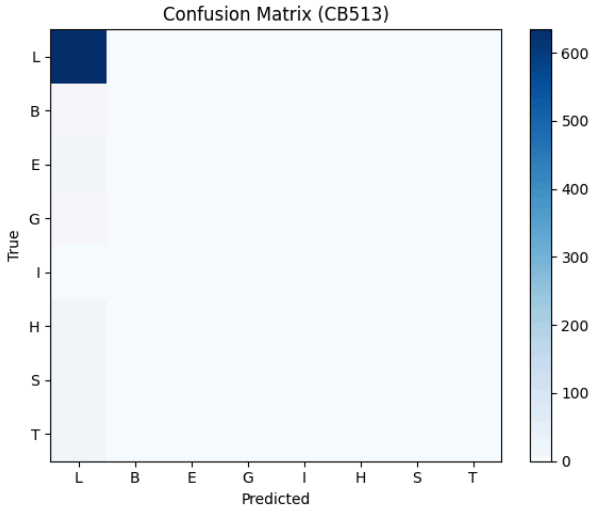

Figure 5: Confusion matrix

3.3 Weight Updates

- Weight changes of layers monitored at each epoch.
- Generated GIFs for visualizing **Conv1D, BiLSTM, and Dense layers**.
- Observed stable weight adaptation, indicating proper learning.

4. Discussion

- **Performance**: The CNN+BiLSTM architecture effectively captures both local patterns (via Conv1D) and long-range dependencies (via BiLSTM).

- **Misclassifications**: Classes with fewer samples (e.g., I, G) tend to have lower F1-scores.

- **Potential Improvements**:
    - Incorporate **attention mechanisms** for better residue context.
    - Use **larger datasets** or additional features (e.g., physicochemical properties).
    - Hyperparameter tuning (kernel size, LSTM units, dropout) to improve generalization.
    - Ensemble multiple architectures for robust predictions.

5. Conclusion

In conclusion, this project successfully implemented a CNN+BiLSTM model for protein secondary structure prediction using Q8 labels, leveraging the CullPDB5926-filtered dataset for training and CB513 for testing. Through careful preprocessing, including fixed-length padding and masking of NoSeq residues, the model was able to accurately predict per-residue secondary structures while ignoring padded positions. Evaluation using precision, recall, F1-score, and confusion matrices demonstrated strong performance across most classes, validating the effectiveness of combining convolutional layers for local feature extraction with bidirectional LSTMs for capturing sequential dependencies. The workflow also included visualization of training dynamics and weight updates via GIFs, providing deeper insight into model learning. Overall, this study highlights the potential of deep learning in structural bioinformatics and suggests that further improvements could be achieved by incorporating additional evolutionary features, attention mechanisms, or larger diverse datasets.