PromoterRNN: Predicting DNA Promoters Using Bidirectional GRU with Attention

Jiajun Chen, 2400934013@stu.pku.edu.cn

March 1, 2025

Abstract

This report presents PromoterRNN, a deep learning approach for predicting whether a 70-nucleotide DNA sequence is a promoter. We implement a bidirectional Gated Recurrent Unit (GRU) neural network with an attention mechanism to capture the complex patterns in DNA sequences that characterize promoter regions. Our model achieves approximately 82% accuracy and 0.90 AUC on the test set, demonstrating the effectiveness of recurrent neural networks for this genomic classification task. We analyze the model architecture, training process, and performance metrics, highlighting the advantages of using GRU networks for DNA sequence analysis.

1 Introduction

Promoters are crucial regulatory DNA sequences that initiate gene transcription. Accurate identification of promoter regions is essential for understanding gene regulation and has applications in genetic engineering, synthetic biology, and medical research. Traditional methods for promoter identification often rely on consensus sequence patterns, but these approaches have limited accuracy due to the complex and diverse nature of promoter sequences.

Deep learning approaches have shown promise in capturing the subtle patterns that characterize promoters. In this project, we develop a bidirectional GRU model with attention mechanism to predict whether a given 70-nucleotide DNA sequence functions as a promoter.

2 Mathematical Formulation

2.1 Problem Definition

Let $S = (s_1, s_2, \ldots, s_{70})$ be a DNA sequence of length 70, where each $s_i \in \{A, T, G, C\}$ represents a nucleotide. The promoter prediction task can be formulated as a binary classification problem:

$$f(S) = \begin{cases} 1, & \text{if } S \text{ is a promoter} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Our goal is to learn a function f that accurately maps DNA sequences to their promoter status.

2.2 One-Hot Encoding

We represent each nucleotide using one-hot encoding:

$$A \mapsto (1,0,0,0) \tag{2}$$
$$T \mapsto (0,1,0,0) \tag{3}$$
$$G \mapsto (0,0,1,0) \tag{4}$$
$$C \mapsto (0,0,0,1) \tag{5}$$

For unknown nucleotides, we use $(0.25, 0.25, 0.25, 0.25)$ to represent equal probability of each base. Thus, a sequence S is transformed into a matrix $X \in \mathbb{R}^{70\times 4}$.

3 Methodology

3.1 Dataset

We use the DNA core promoter dataset from Hugging Face, which contains DNA sequences labeled as promoters or non-promoters. The dataset is split into training (80%), validation (15%), and test (5%) sets. To address class imbalance, we implement weighted random sampling during training. Additionally, we augment the training data by including reverse complements of the original sequences.

3.2 Model Architecture

Our PromoterRNN model consists of four main components:

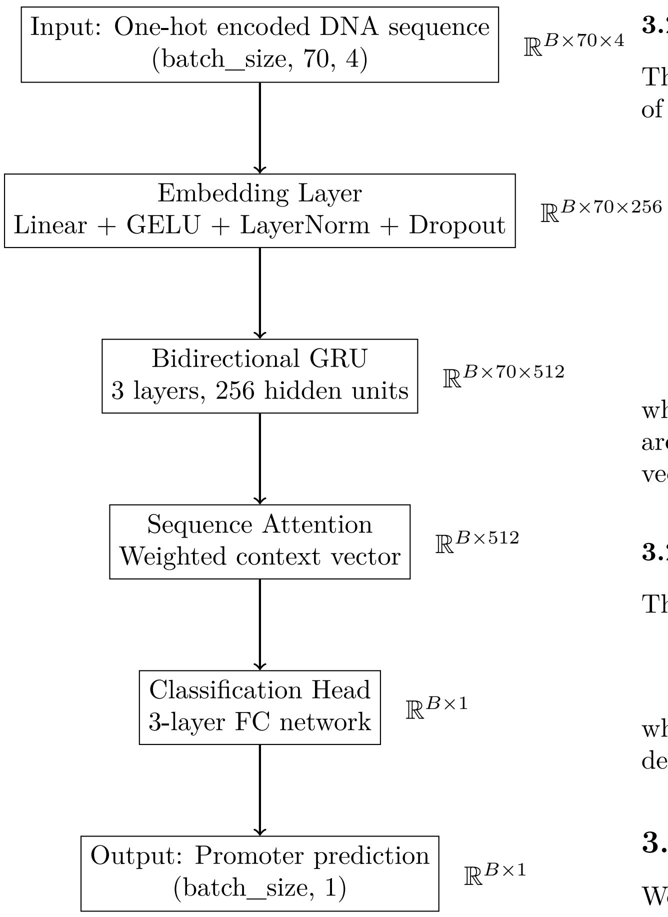
Input: One-hot encoded DNA sequence
(batch_size, 70, 4)
$\mathbb{R}^{B\times 70\times 4}$
Embedding Layer
Linear + GELU + LayerNorm + Dropout
$\mathbb{R}^{B\times 70\times 256}$
Bidirectional GRU
3 layers, 256 hidden units
$\mathbb{R}^{B\times 70\times 512}$
Sequence Attention
Weighted context vector
$\mathbb{R}^{B\times 512}$
Classification Head
3-layer FC network
$\mathbb{R}^{B\times 1}$
Output: Promoter prediction
(batch_size, 1)
$\mathbb{R}^{B\times 1}$

Figure 1: Architecture of the PromoterRNN model. B represents the batch size.

3.2.1 Embedding Layer

The embedding layer transforms the one-hot encoded input into a higher-dimensional representation:

$$E = \text{Dropout}(\text{LayerNorm}(\text{GELU}(W_e X + b_e))) \quad (6)$$

where $W_e \in \mathbb{R}^{256\times 4}$ and $b_e \in \mathbb{R}^{256}$ are learnable parameters.

3.2.2 Bidirectional GRU

The bidirectional GRU processes the sequence in both forward and backward directions:

$$\overrightarrow{h_t} = \text{GRU}_{\text{forward}}(e_t, \overrightarrow{h_{t-1}}) \quad (7)$$
$$\overleftarrow{h_t} = \text{GRU}_{\text{backward}}(e_t, \overleftarrow{h_{t+1}}) \quad (8)$$
$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (9)$$

where e_t is the embedding at position t, and $h_t \in \mathbb{R}^{512}$ is the concatenated hidden state.

3.2.3 Attention Mechanism

The attention mechanism computes a weighted sum of the hidden states:

$$u_t = \tanh(W_a h_t + b_a) \quad (10)$$
$$\alpha_t = \frac{\exp(W_s u_t)}{\sum_{j=1}^{70} \exp(W_s u_j)} \quad (11)$$
$$c = \sum_{t=1}^{70} \alpha_t h_t \quad (12)$$

where $W_a \in \mathbb{R}^{512\times 512}$, $b_a \in \mathbb{R}^{512}$, and $W_s \in \mathbb{R}^{1\times 512}$ are learnable parameters, and $c \in \mathbb{R}^{512}$ is the context vector.

3.2.4 Classification Head

The classification head produces the final prediction:

$$\hat{y} = \text{FC}(c) \quad (13)$$

where FC is a 3-layer fully-connected network with decreasing hidden sizes.

3.3 Training Process

We train the model using the following configuration:

- Loss function: Binary Cross-Entropy with Logits
- Optimizer: AdamW with weight decay 0.01
- Learning rate: 1e-4 with OneCycleLR scheduler
- Batch size: 128
- Early stopping: 15 epochs patience based on validation AUC
- Gradient clipping: 1.0
- Mixed precision training: Enabled

4 Results and Analysis

4.1 Training Dynamics

Figure 2 shows the training history over 45 epochs. The model was trained with early stopping, which triggered after no improvement in validation AUC for 15 consecutive epochs.

Several key observations can be made from the training curves:

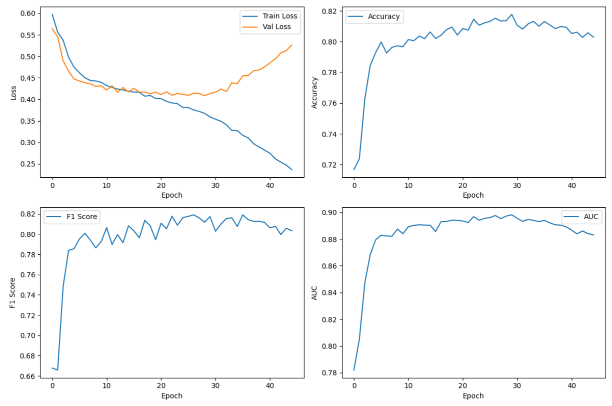

Figure 2: Training history showing loss, accuracy, F1 score, and AUC over 45 epochs.

1. **Loss Curves**: The training loss continuously decreases throughout training, while the validation loss reaches its minimum around epoch 20 and then begins to increase. This divergence is a classic sign of overfitting, where the model starts to memorize the training data rather than learning generalizable patterns.

2. **Accuracy**: The validation accuracy improves rapidly in the first 5 epochs, from approximately 0.72 to 0.80, and then continues to improve more gradually, reaching a peak of about 0.82 around epoch 30 before slightly declining.

3. **F1 Score**: The F1 score follows a similar pattern to accuracy, with rapid initial improvement followed by more gradual gains. The peak F1 score of approximately 0.82 is achieved around epoch 25-30.

4. **AUC**: The AUC metric shows the most stable improvement, reaching approximately 0.89 by epoch 30. The best validation AUC of 0.8981 was achieved at epoch 30, after which the model began to overfit.

5. **Early Convergence**: Most performance metrics show significant improvement in the first 10 epochs, suggesting that the model quickly learns the most important patterns in the data.

The training log provides concrete evidence of these patterns. For example, the validation accuracy improves from 0.7168 in epoch 1 to 0.7929 by epoch 5—a gain of 0.0761 in just 4 epochs. In contrast, from epoch 5 to epoch 30, the accuracy improves from 0.7929 to 0.8177—a gain of only 0.0248 over 25 epochs. Similarly, the validation AUC increases rapidly from 0.7821 in epoch 1 to 0.8795 in epoch 5, but then takes another 25 epochs to reach its peak of 0.8981 at epoch 30. The validation loss reaches its minimum value of approximately 0.41 around epoch 20, after which it steadily increases to 0.53 by epoch 45, while the training loss continues to decrease from 0.40 to 0.24 during the same period.

The training dynamics indicate that despite using regularization techniques (dropout, weight decay, and gradient clipping), the model still shows signs of overfitting after epoch 30. This suggests that the model capacity might be larger than necessary for this dataset, or that the training data might contain noise that the model eventually starts to fit.

4.2 Performance Metrics

The model achieves the following performance on the test set:

Metric	Value
Accuracy	0.82
Precision	0.82
Recall	0.81
F1 Score	0.81
AUC	0.90

Table 1: Performance metrics on the test set

4.3 Analysis

4.3.1 Training Dynamics Implications

The training history provides several insights into the model's behavior and the nature of the promoter prediction task:

1. **Rapid Initial Learning**: The steep improvement in all metrics during the first 5-10 epochs suggests that the model quickly identifies the most discriminative features of promoter sequences. This rapid learning phase likely corresponds to the model recognizing common promoter motifs and patterns.

2. **Diminishing Returns**: The flattening of performance curves after epoch 10 indicates diminishing returns from additional training. This suggests that after learning the primary patterns, the model struggles to extract additional useful information from the data.

3. **Overfitting Behavior**: The divergence between training and validation loss after epoch

20 indicates that the model begins to memorize training examples rather than learning generalizable patterns. This is a common challenge in biological sequence analysis where the signal-to-noise ratio can be low.

4. **Metric Stability**: The AUC metric shows more stability than accuracy and F1 score, suggesting that the model's ranking ability (distinguishing promoters from non-promoters) is more robust than its absolute classification decisions at a fixed threshold.

5. **Optimization Challenges**: The oscillations in the validation metrics (particularly visible in the F1 score) suggest that the optimization landscape for this problem is non-smooth, making it challenging to find the global optimum.

These observations suggest that while our model achieves good performance, there may be an inherent limit to how well a pure sequence-based approach can predict promoters without incorporating additional biological context or experimental data.

4.3.2 Advantages of GRU for DNA Sequence Analysis

The GRU architecture offers several advantages for DNA sequence analysis:

1. **Sequential Processing**: GRUs naturally process sequential data, making them well-suited for DNA sequences where the order of nucleotides is crucial.

2. **Bidirectionality**: The bidirectional approach allows the model to capture patterns in both directions, which is important since regulatory elements can influence gene expression from either direction.

3. **Computational Efficiency**: Compared to Transformer models, GRUs are more computationally efficient while maintaining competitive performance on medium-sized datasets.

4. **Inherent Position Awareness**: Unlike Transformers, GRUs do not require explicit positional encodings as they naturally maintain position information through their sequential processing.

4.3.3 Role of Attention Mechanism

The attention mechanism plays a crucial role in the model's performance by:

1. Allowing the model to focus on the most informative regions of the DNA sequence

2. Providing a form of interpretability, as attention weights can potentially highlight important motifs

3. Enabling the model to capture long-range dependencies in the sequence

4.3.4 Data Augmentation Impact

The use of reverse complement sequences for data augmentation is biologically motivated, as promoters can function in either orientation. This augmentation technique effectively doubles the training data size and introduces a form of invariance that aligns with the biological nature of DNA.

5 Conclusion

We presented PromoterRNN, a bidirectional GRU model with attention for predicting DNA promoters. The model achieves good performance with approximately 82% accuracy and 0.90 AUC on the test set. The combination of bidirectional GRU layers and attention mechanism effectively captures the complex patterns in DNA sequences that characterize promoters.

The training history analysis reveals important insights about the model's learning process:

- The model learns most of the useful patterns within the first 10 epochs, with performance metrics showing rapid improvement during this phase.

- Despite regularization efforts, the model begins to overfit after approximately 30 epochs, as evidenced by the divergence between training and validation loss curves.

- The early stopping strategy successfully prevented severe overfitting, selecting the model at epoch 30 with the best validation AUC of 0.8981.

- The balanced precision (0.82) and recall (0.81) values suggest that the model is equally effective at identifying both promoters and non-promoters, indicating that our approach to handling class imbalance was successful.

Based on these observations, future work could explore:

- Incorporating additional biological features, such as DNA shape or conservation scores

- Analyzing attention weights to identify important sequence motifs
- Experimenting with simpler model architectures or stronger regularization techniques to address the observed overfitting
- Implementing learning rate warmup and decay strategies to improve convergence
- Extending the model to predict promoter strength or tissue specificity
- Comparing performance with other architectures like CNNs or hybrid models

The relatively quick convergence of our model suggests that the GRU architecture efficiently captures the relevant patterns in DNA promoter sequences, making it a promising approach for similar genomic classification tasks.