Project Report : Predicting gene expression from DNA sequences using neural networks

Liyueyue Liu* Ruixiang Huang* Qiuchen Wang* Cheng-Hsuan Li*
Georgia Institute of Technology
{lliu442, rhuang367, qwang634, ianli}@gatech.edu

Abstract

Phenotypes including expression of diseases related genes are governed by universal regulation rules embedded in DNA sequences. Understanding those DNA regulation rules will definitely contribute to the development of potential treatments in disease, recent studies have shown using neural network based models are ideal in predicting gene expression levels. However, training of such models is often computationally expensive and not ideal for most researchers with limited access to the computational resource. In this project, we aimed to find the best sample sizes, model infrastructures with public accessible computational resources such as google Colab platform. We managed to find infrastructures such as implementing 1D transformer, dilation convolutional layer, embedding layers and LSTM that can outperform LegNet, the original SotA with around 10% of original samples. In this report, we outline our motivation, design of experiments, potential technical pitfalls, implementation of different models, compare our models with SotA and analyze our results.

1. Introduction/Background/Motivation

DNA contained all necessary biological information and instruction necessary for any living creatures on earth to flourish and to decease [1]. Decoding information and instructions embedded in DNA will definitely increase our understanding towards both normal and pathological events in our bodies [2]. Behind such complex mechanisms regulated by DNA, the DNA sequence is simply composed of 4 different chemical elements: A, T, C, and G, and such an elegant rule is conserved among all species on earth [3]. Those facts allow us to harness DL models to learn the ATCG encoded pattern embedded in DNA sequence, to decode the universal DNA regulation mechanisms [4] that existed among all creatures, this also provides insights in understanding DNA mechanism regulating disease through studying normal progress, which allows us to switch diseased mechanisms into normal, provide potential application for treatment development [5].

In this task, we utilized the dataset containing millions of promoter and coordinate gene expression levels from yeast [6], aimed to model the most important DNA regulation mechanism based on one basic DNA regulatory element mentioned above: Promoters, the special sequence located before genes, functioned as switches of genes that were located behind them [7]. The state-of-the-art (SotA) in this dataset is LegNet, a neural network model based on convolutional neural network (CNN), and its backbone consists of six blocks, each incorporating grouped convolution, Batch Normalization (BatchNorm), Sigmoid linear unit (SiLU) activation, and residual connections to achieve approximately 0.96 – 0.98 correlation. The LegNet was trained on more than 13 million sequence [8], requires intensive computing resources, substantial labour investment, which is not applicable for most labs with limited resources. Here, we hope to create models that provide accurate correlations with normal or limited data that are better suited for typical research scenarios.

In this work, our primary objective is to develop models capable of effectively capturing complex and non-linear relationships in promoters and coordinating gene expression level while minimizing data requirements. We hypothesize that leveraging dilated convolutions, self-attention layers, or Transformers to enhance long-range context capture can yield more accurate correlations than fully CNN-based approaches, particularly in scenarios with limited data. Additionally, we focus on random promoter-expression data [4, 6, 9].

Raw experimental data was collected in 2020[6], The test set and train set are curated and provided by the Broad Institute of MIT and Harvard at 2022[9].The whole dataset is composed of 13 millions sequence measurements of randomly generated promoters in the single cell organism yeast, with 18 different expression levels labeled with fluorescent reporters. The previous transformer based SotA model built by Vaishnav[9] showed through utilizing cu-

*Random order. All authors made equal contribution.

rated dataset, the model can reach ultimate 0.879 pearson correlation coefficient, proving both the dataset and neural network based model are ideal for predicting gene expression level with promoter sequences provided[10]. The application of one-hot ATCG is to ensure consistent performances given from different models[11].

2. Approach

2.1. General Experiment Settings

This work used PyTorch as the main deep learning framework. All the models were trained on 10240 gene sequences and validated using 2048 sequences unless otherwise noted. The LegNet model trained on 10240 gene sequences were used as baseline in this study.

2.2. Data Process

We adapted data pipeline from LegNet for this study [12]. This enabled us to compare the performance of LegNet and our models in a fair and unbiased manner. In brief, we padded promoter sequence from 5' end to 150 base-pairs with a uniformed segment of plasmids from original experiments [3]. Then, sequencing was transformed into four dimensional one-hot encoding, resulting in 4-dimensional vectors for each ground truth label (i.e., A, T, G, C). The preprocessing procedures developed by current SotA LegNet ensures original biological context of promoters kept while reducing overfitting through adding paddings with uniform biological background.

2.3. Loss Function

We employ two distinct loss functions during training: KL divergence loss and MSE loss.

KL divergence loss:

$$\mathcal{L}_{KL}(P,Q) = \sum_i P(i)\log\left(\frac{P(i)}{Q(i)}\right)$$

MSE loss:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Using KL divergence loss and MSE loss in DNA training is highly effective due to their alignment with the specific requirements of DNA-related tasks. KL divergence is ideal for predicting probability distributions, such as nucleotide probabilities or sequence classifications, ensuring the model accurately captures probabilistic relationships. On the other hand, MSE loss is well-suited for regression tasks, like predicting gene expression levels or binding affinities, where precise numerical predictions are essential. Together, these loss functions effectively address both probabilistic and continuous prediction challenges in DNA modeling, making them an optimal choice for genomics tasks.

2.4. Dilation Convolutional Layer

Dilation convolutional layer inserts gaps between kernel units, providing a larger receptive field without increasing kernel size [13]. Moreover, the introduced gap mimics the complex local interaction between the building blocks of the gene sequence [14]. The number of gaps inserted (dilation rate) is a hyperparameter to determine how long the distance between the building blocks we want to consider.

Specifically, the first vanilla convolution layer in the repeated blocks was replaced by a modified convolution layer. The modified layer contains a dilation convolutional layer and the original vanilla layer, allowing the model to extract information from full range. To simplify this work, we used the same dilation rate configuration for all the dilation models. The dilation rates were 4, 3, 2, 2, 1, and 1, meaning the dilation layers were only effective to the first four convolutional layers. This decreasing ordered pattern also regulated the range of interactions we planned to consider. Additionally, the output of this modified layer was a linear combination of the vanilla convolution and dilation convolution layer, and the ratio was also a hyperparameter. If the ratio is 0, the output is the same as the original LegNet (no dilation). In the experiment, we plan to determine the best ratio between these two layers.

The potential problem of this approach is the risk of overfitting, as the model is trained on a relatively small dataset while incorporating additional CNN layers. The degree of overfitting should be controlled by the ratio of vanilla convolutional layer and dilation convolutional layer.

2.5. Embedding Layer

Brain uses a mechanism of compression to effectively maintain vital representation of inputs [15] . Embedding layers widely used in Natural language processing (NLP) models enable models to retain most important features through compression, allowing models to learn features efficiently while minimizing computational cost. Recent discussion on developing and understanding both artificial and natural intelligence, address the potential computational framework composed of compression [16]. To save utilize limited computational resources wisely, we plan to add compression through introducing embedding layers, hoping to retain most vital information provided from promoter sequence so as to accelerate training.

Although LegNet achieved 0.946 Pearson R, with best performances from all models so far through using 13 millions observations plus 13 millions more imputation in model training, it cannot guarantee to reach the similar performance through utilizing a small dataset as small as 10% to 20% of the original. The main consideration for learning through small datasets is to keep most important structures, ideally keep most important biological relevant information without learning all possible information, since small datasets are usually insufficient in providing enough statistical power of some less common patterns embedded in datasets. In the original LegNet preprocessing pipeline, to enable model learning as much as possible while reducing potential overfit, they padded original data around 80 bp to 150 bp, imputation original datasets through applying A-T, C-G paired rules with biological meaning [3]. Such settings might suggest data preprocessed methods provided by LegNet might not be suitable for few sample learning. To keep most important biological features from small datasets, we added an embedding layer with hidden size 70 bp, almost halved the input size, hoping this helped the original LegNet model to focus on most vital information. In this experiment, we are trying to see if applying embedding will improve the performance of the model.

2.6. Transformer

Transformers enhance sequence analysis by leveraging a self-attention mechanism that models long-range dependencies and global interactions across an input sequence. Unlike convolutional layers, which focus on detecting local patterns through fixed receptive fields, transformers allow each position in a sequence to dynamically attend to all others. This capability enables the capture of complex, distant relationships, making transformers particularly effective for tasks such as DNA sequence analysis.

The main components of a transformer are:

1. Input Embedding with Positional Encoding: Nucleotides (e.g., A, T, C, G) are converted into numerical vectors, with positional encodings added to preserve the sequence order.

2. Self-Attention Mechanism: At each position, the model computes queries (Q), keys (K), and values (V). Attention scores are obtained by calculating the dot product of Q and K, normalized by sequence length, and applied to V. This creates weighted context representations, enabling nucleotides to learn relationships with all other positions in the sequence.

3. Multi-Head Attention: Multiple attention heads capture diverse relational aspects within the sequence. Their outputs are combined to form richer representations.

4. Feedforward Layers: Outputs from the self-attention mechanism are refined through dense layers with activation functions. Residual connections and layer normalization enhance training efficiency and stability.

5. Stacked Layers: Multiple attention and feedforward blocks are stacked to learn hierarchical and complex dependencies.

6. Integrating transformers with a CNN in a hybrid model combines CNNs' local feature extraction capabilities with the global context modeling of transformers. For instance, CNNs can detect motifs like transcription factor binding sites in DNA sequences, while transformers analyze long-distance interactions between these motifs.

7. To mitigate overfitting and manage the computational demands of transformers, techniques like dropout in attention mechanisms and careful tuning of model size are employed, especially when working with smaller datasets.

8. By combining transformers with CNNs for DNA sequence analysis, the model captures long-range dependencies and global context, crucial for understanding interactions and patterns in sequences, while complementing the CNN's ability to identify local motifs.

2.7. LSTM

The LegNet with LSTM approach combines the strengths of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to enhance the analysis of short DNA regulatory regions. Building upon the efficient and scalable design of LegNet, which is inspired by EfficientNetV2, this hybrid model incorporates a LSTM layer after the convolutional feature extraction blocks. The CNN component excels at capturing spatial patterns and local dependencies in the input sequences, while the LSTM layer adds the capability to learn temporal and sequential relationships, allowing the model to better interpret context and long-range dependencies in genomic sequences. This integration makes the model particularly suitable for sequence-to-expression tasks, where understanding both local motifs and sequential structure is crucial. By leveraging this hybrid architecture, the LegNet with LSTM approach offers improved accuracy and robustness for applications such as promoter activity prediction, sequence variant analysis, and the rational design of regulatory sequences.

In this study, we further optimized the model through hyperparameter tuning, including evaluating single-directional and bi-directional LSTMs, varying hidden sizes and dropout rates for the LSTM layer, and adjusting weight decay for the optimizer. These refinements ensured a well-balanced model that effectively addresses the complexities of genomic sequence analysis.

2.8. Anticipated Problems

We anticipated that the huge dataset size would cause high computational and time cost. To solve this problem, we downsampled the data set. The downsampling works well.

We anticipated the potential issue of overfitting introduced by the modified models, particularly during the training of LegNet with LSTM. As expected, overfitting became a challenge during the actual training process. To address this, we began by modifying the LSTM configuration, switching from bi-directional to single-directional LSTM and adding a dropout rate. However, these initial adjustments did not resolve the issue. Consequently, we explored a more comprehensive approach, fine-tuning multiple hyperparameters simultaneously, including hidden size, weight decay, and dropout rate. Through this iterative process, we ultimately achieved improved results, mitigating overfitting and enhancing model performance.

3. Experiments and Results

3.1. Dilation Convolutional Layer

We trained the model with a dilation ratio ranging from 0 to 0.4. The baseline model was the LegNet (dilation ratio is 0) and its best Pearson correlation was 0.515. From the figure 1, we saw that the dilation convolutional layer increases the Pearson correlation coefficients up to 0.566 without overfitting. We also observed that the correlations improved more when the model incorporates more dilation and plateaued at 0.4 ratio. This indicated that the nearest gene building blocks are still dominant to the prediction of the overall biological functions. However, we noticed that the losses of dilation approaches were generally higher than vanilla convolutional approaches. One possible reason is that the confidence of some correct predictions dropped slightly. Nonetheless, this mismatch between loss and accuracy resulted in our concern about the source of the performance improvement.

We next investigated whether the performance improvement resulted from extracting information from longer-range interactions. Specifically, we trained the models on datasets of varying sizes, ranging from 5,120 to 20,480 samples. The dilation models should outperform LegNet regardless of data size, if the source of the performance boosting is from longer-range interactions. However, from the fig 2, we saw that dilation convolution models provided poorer performances than LegNet when the sample size was 5120, and their performance improvements were not obvious in the 20480-sample size case. This implies that longer-range interactions are not the main cause, or the dilation convolutional strategy does not capture longer-range interactions. Moreover, in these experiments, we showed that LegNet and its counterparts provide similar performance when the data size is in 10000 to 20000 range, suggesting 10000 gene sequences is a reasonable target for the projects with budget limitations.

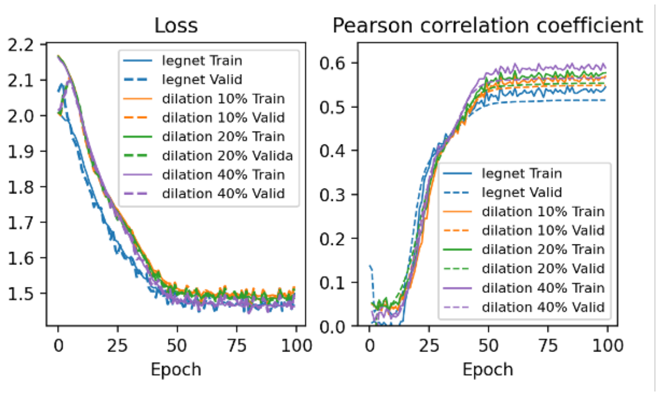

Figure 1. losses and Pearson correlation coefficients of LegNet and dilation convolutional models trained on 10240 gene sequences. The dilation convolutional models provide additional performance improvement.

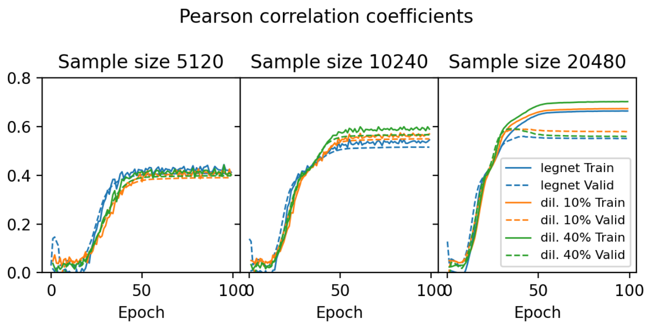

Figure 2. losses and Pearson correlation coefficients of LegNet and dilation convolutional models trained on 5120, 10240, and 20480 gene sequences. The inconsistent performances imply that the longer-range interactions of the gene building blocks are not the main cause of the performance improvement.

3.2. Legnet with embedding layer

We set the hidden size of the embedding layer to 70, almost equal to the original data input size, then train both the LegNet with embedding model and original LegNet model with 102410 experiment observation data on google colab with T4 NVDIA T4 GPU. Based on the result shown in fig 3, we can see that the LegNet model with embedding layer increases the Pearson correlation coefficient to 0.52 without overfitting. Also, adding the embedding layer reduces loss to 1.42 compared to the original LegNet performance. Both increasement in ultimate Pearson correlation coefficient and decrement in loss suggested, adding embedding layers can extract representative patterns with biological meanings effectively from input, which help models to focus on important patterns for prediction and result in the improvement of model generalization. The application of compression concept in the original model does help improve model's performances in the task with small samples.

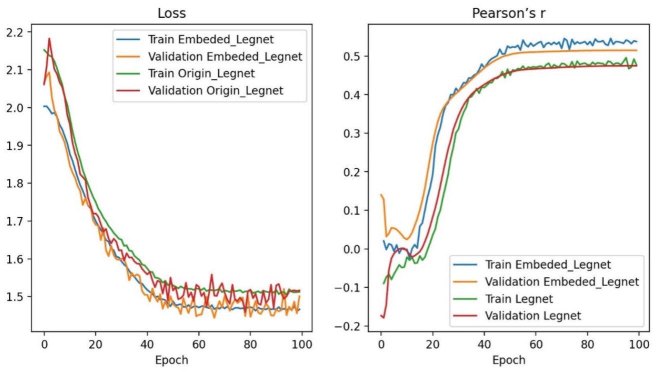

Figure 3. Loss and Pearson's r of Legnet with embedding layer (hidden size=70, sample size = 10240, epoch=100, batch size=512)

We decided not to go further trying different hidden sizes due to the fact that in biology, promoters in yeast genome are usually longer than 500 bps [17], and in experimental data, the promoters extracted are ranging from 60 bps to 80 bps, which are already much shorter. Compressing input to sizes shorter than 70 might result in potential loss of biological information. The choice of hidden size of embedding layer should consider domain knowledge.

3.3. Legnet(CNN) with transformer

Compared to the original LegNet using the same data size but with a different batch size, we can generally conclude that adding a transformer benefits the DNA training for LegNet. This is evident as the batch size in this test (512) is smaller than in the previous tests, yet the Pearson correlation and loss are better than those of the original model. To further explore this, we ran the model on different data sizes with the same settings, and the fig 4 shows the results for the varying data sizes.

We can observe that, except for the 5120 dataset, the training and validation results are similar across all sizes. The 5120 dataset shows signs of overfitting due to the limited diversity and fewer examples, which make it easier for the model to memorize the smaller dataset, resulting in a higher training correlation. Based on this experiment, we can conclude that adding the transformer improves the original LegNet, and using a dataset size of at least 10240 is recommended to avoid overfitting.

3.4. LegNet(CNN) with LSTM

The success was defined as improving the model's validation loss and pearson's r compared to the baseline (LegNet without LSTM), while maintaining a reasonable balance between training and validation losses to ensure generalization.

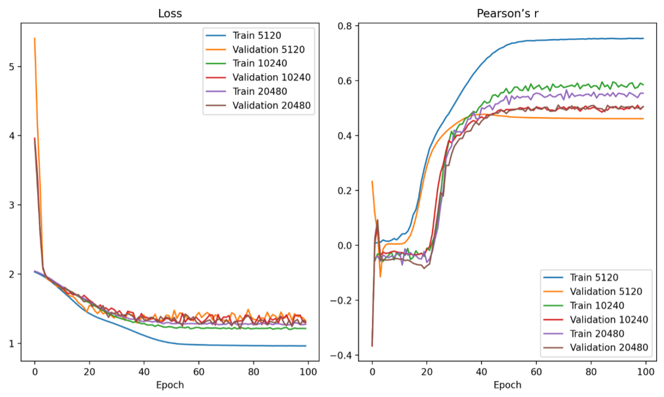

Figure 4. Loss and Pearson's r of Legnet with transformer layer (sample size = (5120, 10240, 20480), epoch=100, batch size=256)

The experiments conducted include baseline experiment (LegNet without LSTM) and LSTM integration which means adding a LSTM layer to LegNet. Five different hyperparameters are trained simultaneously:

1. Directionality: Single vs. bi-directional LSTM
2. Hidden size: 64, 128, and 256
3. Dropout rate: 0, 0.1, 0.2, 0.3, 0.5
4. Weight decay: 0.01, 0.001
5. Learning rates: 4e-7, 4e-5

The quantitative metrics used in this study include Pearson's r, the train-validation loss gap, and the train-validation Pearson's correlation coefficient gap. The qualitative metrics involve observations of overfitting and underfitting patterns. The detailed results are summarized in Table 2 in Appendix section. The findings indicate that larger models, particularly those with a hidden size of 256 and bidirectional structures, showed a strong tendency to overfit despite the application of dropout. This overfitting caused the validation performance to fall below the baseline. In contrast, smaller models with a hidden size of 64 consistently underfit, failing to capture sufficient patterns and relationships. The optimal configuration was identified as a balanced model with a hidden size of 128, a dropout rate of 0.3, a weight decay of 0.001, a single-directional LSTM, and a learning rate of 4e-7. This setup provided the best balance between performance and generalization, exceeding the baseline validation Pearson's correlation coefficient. These results demonstrate that mid-sized configurations with carefully tuned dropout and weight decay effectively mitigate overfitting and underfitting, delivering superior model performance, which indicates the success of the hybrid model.

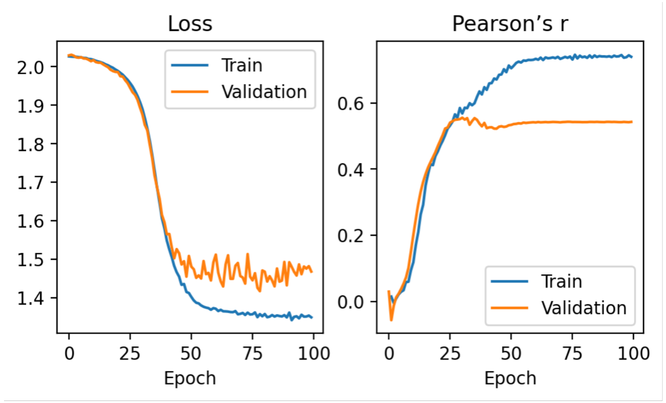

Figure 5. Loss and Pearson's r of Legnet with LSTM (sample size = 20480, epoch=100, batch size=512)

4. Summary

This study built upon the state-of-the-art (SotA) LegNet model, a CNN-based architecture designed for sequence-to-expression tasks in gene regulation. This work investigated various architectural modifications, including the integration of LSTM, embedding layers, dilation convolutional layers, and transformers, aiming to enhance performance and generalization while addressing computational challenges. These models are designed for accurate promoter-expression prediction with reduced data size.

By comparing all the modified models with the LegNet model (baseline), we can conclude that by adding different layers to the original LegNet model, the accuracy has all been improved. For example, the LegNet with LSTM achieved a good balance of accuracy and generalization with validation Pearson's r equaling to 0.539. The LegNet with transformer improved global dependency modeling and found out that the model is most beneficial for larger datasets. The LegNet with dilation layers achieved the highest Pearson's r to 0.566 and found out that the performance gains diminished with very small or large datasets. The LegNet with embedding layer also reduced the loss and it is found that it is ideal for small datasets and limited computational resources. These findings demonstrate the potential for hybrid and optimized architectures to advance DNA sequence analysis in diverse research settings.

References

[1] Crick, F. (1970). Central dogma of molecular biology. Nature, 227(5258), 561-563. 1

[2] Bushweller, J. H. (2019). Targeting transcription factors in cancer—from undruggable to reality. Nature Reviews Cancer, 19(11), 611-624. 1

[3] Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. Nature, 171(4356), 737-738. 1, 2, 3

[4] de Boer, C. G., & Taipale, J. (2024). Hold out the genome: a roadmap to solving the cis-regulatory code. Nature, 625(7993), 41-50. 1

[5] Lambert, S. A., Jolma, A., Campitelli, L. F., Das, P. K., Yin, Y., Albu, M., ... & Weirauch, M. T. (2018). The human transcription factors. Cell, 172(4), 650-665. 1

[6] de Boer, C. G., Vaishnav, E. D., Sadeh, R., Abeyta, E. L., Friedman, N., & Regev, A. (2020). Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. Nature biotechnology, 38(1), 56-65. 1

[7] Andersson, R., & Sandelin, A. (2020). Determinants of enhancer and promoter activities of regulatory elements. Nature Reviews Genetics, 21(2), 71-87. 1

[8] Penzar, D., Nogina, D., Noskova, E., Zinkevich, A., Meshcheryakov, G., Lando, A., I. V. (2023). LegNet: a best-in-class deep learning model for short DNA regulatory regions. Bioinformatics, 39(8), btad457. 1

[9] Vaishnav, E. D., de Boer, C. G., Molinet, J., Yassour, M., Fan, L., Adiconis, X., Regev, A. (2022). The evolution, evolvability and engineering of gene regulatory DNA. Nature, 603(7901), 455-463. 1

[10] Rafi, A. M., Nogina, D., Penzar, D., Lee, D., Lee, D., Kim, N., de Boer, C. G. (2024). A community effort to optimize sequence-based deep learning models of gene regulation. Nature biotechnology, 1-11. 2

[11] Choong, Allen Chieng Hoon, and Nung Kion Lee. "Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method." 2017 International Conference on Computer and Drone Applications (IConDA). IEEE, 2017. 2

[12] https://github.com/autosome-ru/LegNet/tree/main 2

[13] Yu F., Koltun V. (2016). Multi-scale context aggregation by dilated convolutions. ICLR, 2016. 2

[14] Gupta A., Rush A. (2017) Dilated convolutions for modeling long-distance genomic dependencies. ICML, 2017 2

[15] Ma, Y., Tsao, D., & Shum, H. Y. (2022). On the principles of parsimony and self-consistency for the emergence of intelligence. Frontiers of Information Technology & Electronic Engineering, 23(9), 1298-1323. 2

[16] Sridhar, S., Khamaj, A., & Asthana, M. K. (2023). Cognitive neuroscience perspective on memory: overview and summary. Frontiers in Human Neuroscience, 17, 1217093. 2

[17] Lubliner, S., Regev, I., Lotan-Pompan, M., Edelheit, S., Weinberger, A., & Segal, E. (2015). Core promoter sequence in yeast is a major determinant of expression level. Genome research, 25(7), 1008-1017. 5

Student Name	Contributed Aspects	Details
Cheng-Hsuan Li	1. Built a data pipeline	Developed a data pipeline from the LegNet open-source GitHub project, detailed in group_project.ipynb and helper_functions.py.
	2. Tuned hyper-parameters	Optimized hyper-parameters of LegNet to suit the project's data scale, as documented in group_project.ipynb and the project report.
	3. Trained model with dilation convolutional layer	Conducted training using a dilation convolutional layer and analyzed the model's performance, described in section 2.4 and 3.1
	4. Drafted Introduction (not shown in the final draft)	Authored the introductory sections, providing context, background, and motivation for the project.
Qiuchen Wang	1. Implemented LegNet with LSTM	Designed and implemented LegNet integrated with LSTM, trained the model with five different hyper-parameter configurations, and analyzed the performances. Detailed in group_project_LSTM.ipynb, legnetWithLSTM.py, and Section 2.7 and 3.4 of the report.
	2. Drafted Approach, Experiments, and Results	Authored sections detailing the LSTM approach, experimental setup, results analysis, and summary, as outlined in the report.
Liyueyue Liu	1. Project Conceptualization	Played a key role in formulating the project's concept, objectives, and overall approach.
	2. Data Preprocessing	Handled data preprocessing tasks to prepare the dataset for training and analysis.
	3. Drafted and Revised Key Sections	Authored and revised the Introduction, Background, Motivation, and Abstract sections of the report.
	4. Implemented and Trained Model	Developed and trained a model with an embedding layer, and analyzed its performance, ensuring insights into model behavior and outcomes.
Ruixiang Huang	1. Added Transformer/CNN to SeqNN Model	Integrated Transformer and CNN components into the SeqNN model, enhancing its capability for sequence modeling.
	2. Drafted Template	Created a comprehensive template draft to guide documentation and structure for the project deliverables.

Table 1. Contributions of team members.

Supporting Information

Hidden size	**Directional**	**Dropout**	**Learning rate**	**Weight decay**	**Train loss (last epoch)**	**Validation loss (last epoch)**	**Train Pearson's r (last epoch)**	**Validation Pearson's r (last epoch)**	**Loss gap**	**Pearson's gap**	**Observations**
256	Bi-directional	0	4e-7	0.01	0.7726	1.5596	0.7692	0.4355	0.787	0.3337	Overfit
256	Bi-directional	0.3	4e-7	0.01	0.7667	1.5626	0.7763	0.4761	0.7959	0.3002	Overfit, but dropout improves generalization slightly
256	Single-directional	0.3	4e-7	0.01	1.0118	1.4043	0.7525	0.4848	0.3925	0.2677	Overfit
256	Single-directional	0.5	4e-7	0.01	0.9134	1.5108	0.7567	0.4504	0.5974	0.3063	Overfit
256	Single-directional	0.3	4e-7	0.01	0.7851	1.5898	0.7974	0.4412	0.8047	0.3562	Overfit
256	Single-directional	0.3	4e-7	0.001	0.9120	1.5322	0.7604	0.4392	0.6202	0.3212	Overfit
128	Single-directional	0.3	4e-7	0.01	1.4200	1.5435	0.6120	0.4628	0.1235	0.1492	Balanced
128	Single-directional	0.3	4e-7	0.001	1.3494	1.4682	0.7396	0.5428	0.1188	0.1968	Best model
128	Single-directional	0.3	4e-7	0.0001	1.3733	1.5322	0.7651	0.5388	0.1589	0.2263	Balanced
128	Single-directional	0.3	4e-5	0.01	0.0652	2.4602	0.9786	0.6050	2.395	0.3736	Severe Overfit
128	Single-directional	0.5	4e-7	0.01	1.3555	1.5142	0.6355	0.4390	0.1587	0.1965	Balanced
64	Single-directional	0.3	4e-7	0.01	1.8347	1.8467	0.4279	0.3136	0.012	0.1143	Underfit
64	Single-directional	0.2	4e-7	0.01	1.7329	1.7701	0.1878	0.1266	0.0372	0.0612	Underfit
64	Single-directional	0.1	4e-7	0.01	1.7226	1.8114	-0.1187	-0.1670	0.0888	0.0483	Underfit
-	-	-	4e-7	-	1.4797	1.4815	0.5284	0.5280	0.0018	-0.0016	Baseline

Table 2. LegNet with LSTM Hyperparameter Tuning Results