DREAM Challenge 2022
Predicting promoter sequences using millions of random promoter sequences

UTK Bioinformatics

August 2022

Abstract

For this challenge, we use a basic Bert transformer [2], training the labeled training in a 2-step process: whole data training and subset fine tuning. We first train the model with a designed regression trainer using all 6.7 million samples provided in the training data, then based on the bins of expression tiers, sample equally from 18 expression bins to create a close to uniform distribution subset of data, and use this subset to further tune the model to allow it to overcome it's tendency to bias toward the middle expression region that contains significantly more samples than low and high expression regions. Based on our evaluation metric, this step effective increased the performance of the model, especially for prediction accuracy on low and high expression genes.

1 Data Usage

In this challenge, we used the data mostly as it is, in the fine-tuning process, we sampled a subset from the entire set, as described in below training process, we use a 99-1 random train test split for both training and fine tuning. The training dataset contains all 6.7 million samples provided in the challenge, then based on the bins of expression tiers, we sample k promoter sequences randomly from each expression bin (bins 1-18) to create a close to uniform distribution subset of data, by looking at the distribution of the data and testing different k values, we used k=25000. Other values of k we have tested include 3000, 15000, 50000 and 1000000. Both files used in the two-step training process has been included in the GitHub.

2 Model

We use a basic transformer model [5], and specifically a basic configuration of BERT (Bidirectional Encoder Representations from Transformers) [2]. In addition to achieving state-of-the-art performance on natural language processing tasks, BERT has been successfully applied to prediction tasks on biological sequences [3], [1]. This BERT base model uses 12 layers of transformer blocks with a hidden size of 768 and the number of self-attention heads as 12 and, in total, around 110M trainable parameters.

3 Training Procedure

Our training procedure is a two step process based on provided labeled sequence training data, During both training, we treat each individual nucleotide as a word, and split the entire promoter sequence, typically composed of at most 110 bases, into a sentence with at most 110 words, separated by spaces. Because some sequences are larger, we use 128 as our max input size and any sequence with less than 128 words (nucleotides) are padded with ”[PAD]” tokens to ensure uniform input sizes. All training procedures and models were implemented in python, our training environment involves PyTorch and Hugging-face. [6].

3.0.1 Tokenizer

We first train a word piece tokenizer [4] customized to the input, due to the special nature of inputs converted from SNA sequences, vocabularies are guaranteed to be [’A’,’G’,’G’,’T’,’N’], while adding five special tokens commonly included in transformer models: ”[PAD]”, ”[UNK]”, ”[CLS]”, ”[SEP]”, ”[MASK]”, which results in a fixed size of vocabulary size of 10 tokens. In future training only two of the special tokens – the mask token ”[MASK]” and the pad token ”[PAD]” – are used in further training, the other special tokens were included during initial development to support alternative model formulations.

3.0.2 Training and Fine-Tuning

We first train a BERT model from scratch, using only the entire data provided for the contest (approximately 6.7 million sequences), with a designed regression model, using a MSE (mean squared error) loss function and

adamW optimization function. After training the model with all the samples and their expression labels, we further train the model with the a fine tuned subset. This subset used for fine tuning is sampled equally from 18 expression bins of the full dataset to create a close to uniform distribution subset. The two step training process shares the same infrastructure except that the fine tuned step uses a huber loss function instead of MSE loss function, the detailed parameter for the layers and parameters are included in below table:

Model Architectures	BertForSequence Classification	**Train Epoches**	3
Hidden Layer Activation Function	GELU	**Per Device Batch Size**	36
Hidden Layers	12	**Initial Learning Rate**	1e-5
Hidden Layer Size	768	**Optimizer**	adamW
Dropout_prob	0.1	**Loss Function**	MSE/Huber
Max Position Embeddings	512	**Layer_norm_eps**	1e-12

4 Other important features

5 Contributions and Acknowledgement

Name	Role	Affliation	Email
Zhixiu Lu	Team Lead	Department of EECS, University of Tennessee at Knoxville	zlu21@vols.utk.edu
Owen Queen	Team Member	Department of EECS, University of Tennessee at Knoxville	oqueen@vols.utk.edu
Ashley Babjac	Team Member	Department of EECS, University of Tennessee at Knoxville	ababjac@vols.utk.edu
Scott Emrich	Principal Investigator	Department of EECS, University of Tennessee at Knoxville	semrich@utk.edu

6 Reference

References

[1] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 02 2022.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021.

[4] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[6] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan

Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.