Факультет
Компьютерных наук

Департамент Программной
Инженерии

Москва
2024

Идентификация вторичных структур ДНК с помощью нейросетей

AI-powered Identification of DNA Secondary Structures

Тип проекта: исследовательский
Вид проекта: индивидуальный

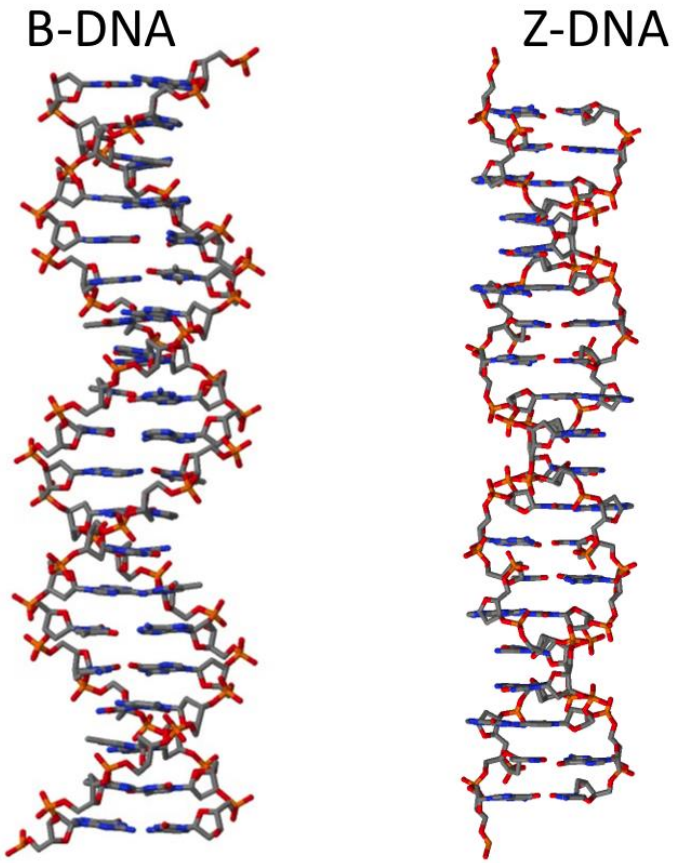
Автор проекта: Лаптева Анна, БПИ215
Руководитель проекта: Преподаватель НИУ ВШЭ, бакалавр, **Боревский Андрей Олегович**

Предметная область. Z-ДНК и B-ДНК

Помимо всем известной стандартной правой формы ДНК (B-ДНК), в 1979 году была обнаружена левосторонняя форма, называемая Z-ДНК.

Функциональная роль Z-ДНК разнообразна, и многие из её аспектов остаются недостаточно исследованными.

Установлено, что Z-ДНК может действовать как репрессор в промоторе гена ADAM-12, который известен своей чрезмерной активностью во многих формах рака у человека. Она также обнаружена в области гиппокампа мозговых образцов, сильно пораженных болезнью Альцгеймера. Белки ADAM, содержащие Z-ДНК в промоторной области, ассоциированы с различными метаболическими и воспалительными заболеваниями, включая диабет, сепсис, болезнь Альцгеймера и ревматоидный артрит.



Цель и задачи работы

Цель работы – разработка модели глубинного обучения, способной с высокой точностью предсказывать участок Z-ДНК на основе вторичных структур ДНК.

В рамках данной работы должны быть выполнены следующие задачи:

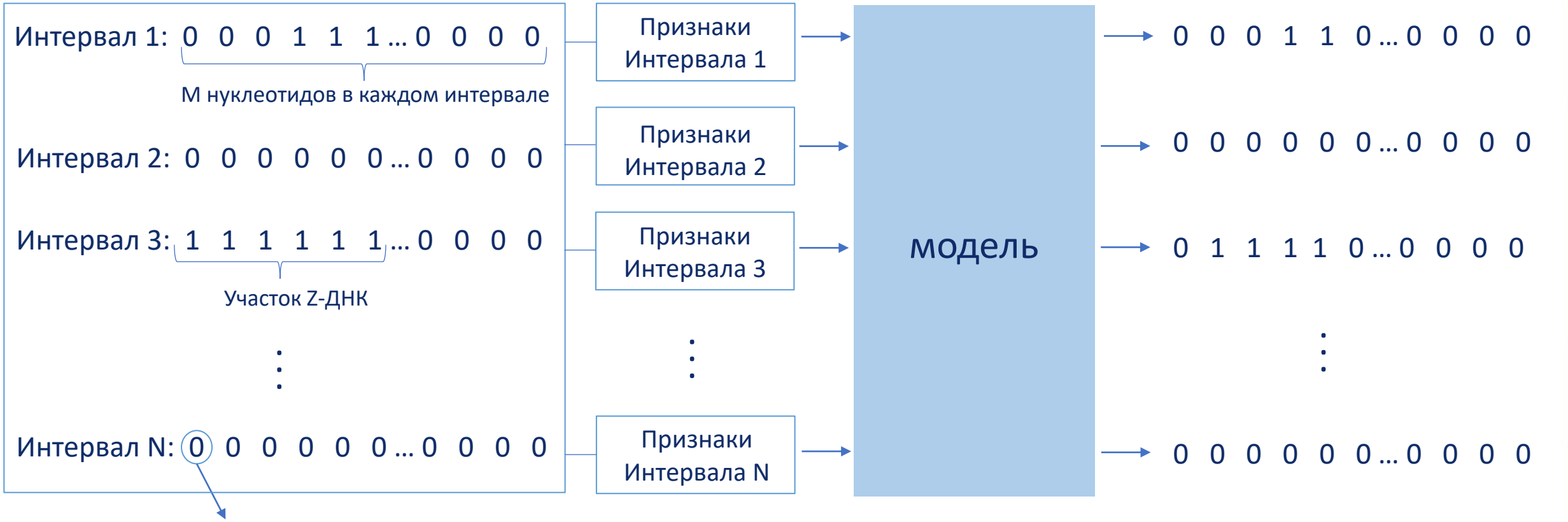
1) Реализация и оптимизация модели глубинного обучения, разработанной научно-учебной лабораторией искусственного интеллекта для вычислительной биологии НИУ ВШЭ [1].
2) Исследование эффективности различных архитектур моделей глубинного обучения для задач исследования ДНК и вторичных структур ДНК.
3) Реализация модели глубинного обучения, параметры которой можно проинтерпретировать в будущем.

Актуальность работы

Актуальность работы заключается в разработке эффективной модели глубинного обучения для предсказания участков Z-ДНК для получения инструмента для диагностирования участков Z-ДНК.

Аномальное образование участков Z-ДНК ассоциируется с рядом серьезных заболеваний. Среди них могут быть рассмотрены болезни, связанные с нарушениями структуры и функционирования ДНК, такие как рак, нейродегенеративные заболевания, аутоиммунные расстройства и другие патологии, что подчеркивает важность точного и надежного определения участков Z-ДНК.

Формат данных



Каждый нуклеотид имеет 1058 признака, анализируя признаки каждого нуклеоида в интервале, модель предсказывает, относится ли нуклеотид к Z-ДНК (выдает 1) или нет (выдает 0)

Анализ существующих решений

Лучшая модель лаборатории DeepZ, полученная после проведения более 150 экспериментов с различными архитектурами, основана на LSTM.

Метрики DeepZ на интервале размером 5000 нуклеотидов:

ROC-AUC = 85%, F1-score = 39.1%

Модель имеет достаточно низкий F1-score, поэтому первостепенной задачей стоит увеличение F1-score, и в дальнейшем для сравнение моделей будем опираться больше на значение этой метрики.

```
class DeepZ(nn.Module):
    def __init__(self):
        super().__init__()
        self.rnn = nn.LSTM(1058, 500, 2, bidirectional=True)
        self.seq = nn.Sequential(
                    nn.Dropout(0.5),
                    nn.Linear(2 * 500, 500),
                    nn.Sigmoid(),
                    nn.Dropout(0.5),
                    nn.Linear(500, 2)
        )

    def forward(self, x):
        x, (h_n, c_n) = self.rnn(x)
        x = self.seq(x)
        return F.log_softmax(x, dim=-1)
```

Метрики качества

В качестве основных метрик качества были выбраны метрики ROC-AUC и F1-score, так метрика F1-score устойчив к дисбалансу классов, а ROC-AUC оценивает качество классификатора в целом

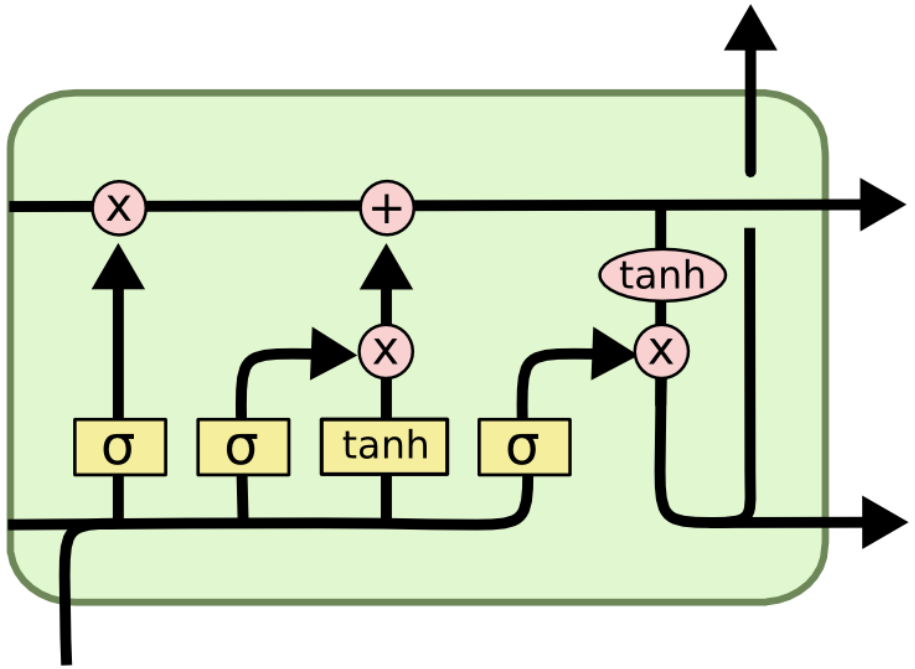
$$recall = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FP}$$

LSTM

LSTM за счет своего устройства, позволяет иметь память о том, что было до текущего элемента, и на основе всей последовательности делать предсказание.

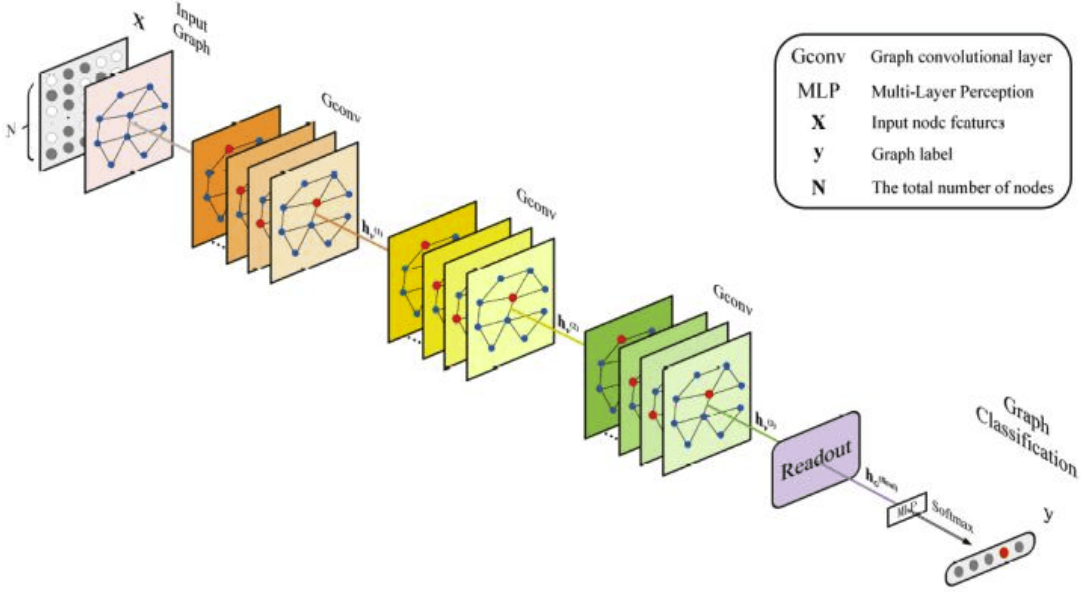


Входные данные в нашей модели состоят из последовательности нуклеотидов, а задача модели предсказать содержит ли последовательность Z-ДНК или нет, анализируя значение признаков для каждого нуклеотида. Из этого можно сделать вывод, что LSTM может стать эффективным решением для создания нужной модели, так как позволит исследовать последовательность в целом, а не отдельные нуклеотиды.

Графовые модели

ДНК может быть представлена в виде графа, где узлы представляют отдельные нуклеотиды, а ребра обозначают связи между ними. Такое представление позволяет сохранить информацию о структуре ДНК и взаимосвязях между различными участками.

Графовые модели позволяют учитывать пространственную структуру ДНК, такую как петли, косые соединения и другие структурные элементы, которые могут оказывать влияние на её функционирование.



Эксперименты

1. Эксперименты с длиной интервала.
2. Эксперименты с архитектурами графовых моделей.
3. Эксперименты с ребрами графовых моделей.
4. Эксперименты с одновременным использованием LSTM и графовых сверток.
5. Эксперименты с дистилляцией знаний.

Эксперименты с длиной интервала

Гипотеза: уменьшение размера интервала приводит к увеличению метрики F1-score.

Идея: в среднем участки Z-ДНК имеют размер в 400 нуклеотидов. Логично предположить, что важными данными для образования Z-ДНК являются признаки нуклеотидов из самого участка и окрестности участка. Соответственно, если размер слишком большой, то окрестность предполагаемого интервала содержит слишком много нуклеотидов, чьи признаки определяют B-ДНК а не Z-ДНК, что отражается на предсказаниях модели.

Результаты обучения DeepZ на интервале размера 1000 нуклеотидов:

F1 = 59.61%, AUC = 88.11%

Эксперименты с архитектурами графовых моделей

В работе были рассмотрены следующие графовые свертки: GCNConv, GraphConv, GATConv, GATv2Conv, SAGEConv из библиотеки torch_geometric.nn.

Метрики графовых моделей на интервале размера 5000

Свертка, на которой построена архитектура модели	Со слоями нормализации	
	F1-score	ROC-AUC
GCNConv	0.435	87.58%
GraphConv	0.464	86.40%
GATConv	0.32	93.71%
GATv2Conv	0.371	95.15%
SAGEConv	0.421	87.95%

Эксперименты с архитектурами графовых моделей

Метрики графовых моделей на интервале размера 1000

Свертка, на которой построена архитектура модели	Без слоев нормализации		Со слоями нормализации	
	F1-score	AUC-ROC	F1-score	ROC-AUC
GCNConv	0.582	88.90%	0.569	91.47%
GraphConv	0.605	90.74%	0.611	92.23%
GATConv	0.554	91.59%	0.558	91.62%
GATv2Conv	0.556	91.27%	0.55	91.92%
SAGEConv	0.634	88.19%	0.614	92.20%

Метрики моделей с большим количеством графовых сверток и с BatchNorm:

	F1-score	AUC-ROC
GraphConv	0.6425	90.79%
SAGEConv	0.6423	90.62%

Вывод: по итогам экспериментов с архитектурой видно, что уменьшение размера интервала и использование слоев BatchNorm увеличили значение F1-score для всех моделей.

Самыми эффективными свёртками оказались GraphConv и SAGEConv.

Эксперименты графовые модели + LSTM

Большая модель с GraphConv + LSTM:

ROC-AUC = 90.62%, F1-score = 0.603

Большая модель с SAGEConv + LSTM:

ROC-AUC = 92.01%, F1-score = 0.621

Как видно из экспериментов, большая модель со свертками SAGEConv показала себя лучше с LSTM, чем большая модель со свертками GraphConv

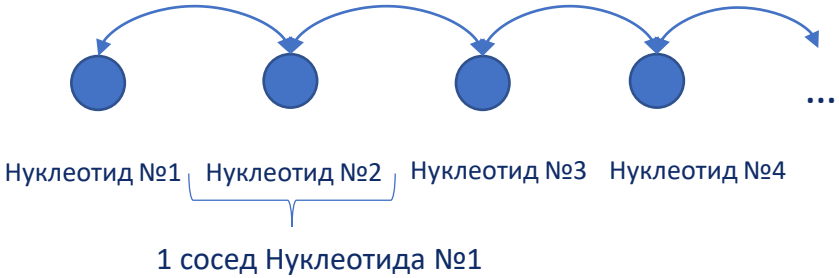
Модель GraphZGraphConvBatchNorm_v2LSTM будем использовать дальше.

```
class GraphZGraphConvNorm_v2LSTM(torch.nn.Module):
    def __init__(self):
        super(GraphZGraphConvBatchNorm_v2LSTM, self).__init__()
        self.lstm = torch.nn.LSTM(input_size=1058, hidden_size=500,
num_layers=1, batch_first=True, bidirectional=True)

        self.conv1 = GraphConv(500 * 2, 800)
        self.bn1 = torch.nn.BatchNorm1d(800)
        self.conv2 = GraphConv(800, 600)
        self.bn2 = torch.nn.BatchNorm1d(600)
        self.conv3 = GraphConv(600, 400)
        self.bn3 = torch.nn.BatchNorm1d(400)
        self.conv4 = GraphConv(400, 200)
        self.bn4 = torch.nn.BatchNorm1d(200)
        self.conv5 = GraphConv(200, 2)

    def forward(self, x, edge_index):
        lstm_out, _ = self.lstm(x)
        x = F.dropout(lstm_out, p=0.6, training=self.training)
        x = F.elu(self.bn1(self.conv1(x, edge_index)))
        x = F.dropout(x, p=0.6, training=self.training)
        x = F.elu(self.bn2(self.conv2(x, edge_index)))
        x = F.dropout(x, p=0.6, training=self.training)
        x = F.elu(self.bn3(self.conv3(x, edge_index)))
        x = F.dropout(x, p=0.6, training=self.training)
        x = F.elu(self.bn4(self.conv4(x, edge_index)))
        x = F.dropout(x, p=0.6, training=self.training)
        x = self.conv5(x, edge_index)

        return F.log_softmax(x, dim=1)
```

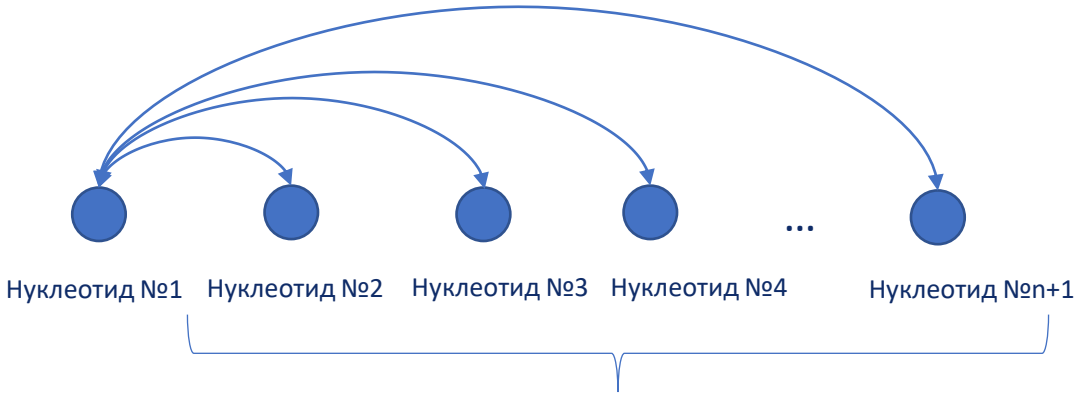
Эксперименты с ребрами графовых моделей

В графовой нейронной сети по очереди применяются слои, которые собирают информацию с соседей и обновляют информацию в вершине.
Нужно проверить гипотезу, повлияет ли увеличение количества соседей для каждого нуклеотида на эффективность графовой модели.

Классический вариант расставления ребер:



Экспериментальные варианты расставления ребер для каждого нуклеотида:



Эксперименты с ребрами графовых моделей

Все эксперименты с количеством соседей для каждого нуклеотида проводились на одной и той же графовой модели с помощью Kaggle, память которого не позволяет провести эксперименты более чем с 90 соседями, используя графовые сети.

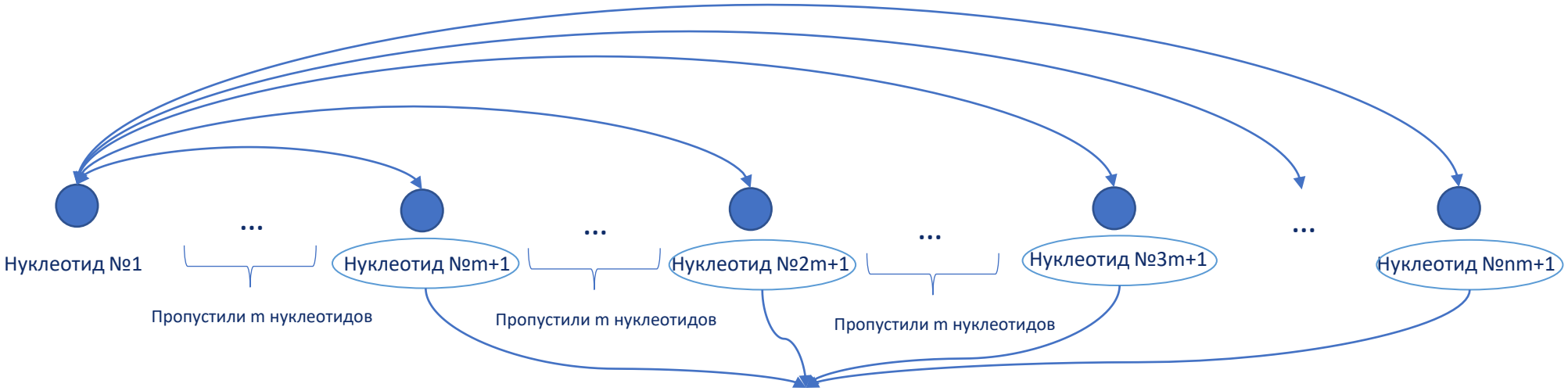
Как видно из таблицы, нельзя уверенно сказать, что увеличение кол-ва ребер, исходящих из каждого нуклеотида, прямо пропорционально качеству модели. Возможно, что для того, чтобы увидеть больший эффект, нужно чтобы нуклеотид был связан еще дальше, чем на 90 нуклеотидов от положения нуклеотида.

Количество соседей	F1-score	ROC-AUC
90	0.655	91.29%
80	0.611	89.38%
60	0.63	91.20%
40	0.633	91.30%
20	0.618	90.03%
10	0.604	89.86%
5	0.61	89.62%
1	0.614	92.20%

Эксперименты с ребрами графовых моделей

Попробуем увеличить диапазон соседей каждого нуклеотида за счет увеличения расстояния между соседями.

Экспериментальные варианты расставления ребер для каждого нуклеотида:



n соседей нуклеотида №1 , где $5 \leq n \leq 90$ и $1 \leq m \leq 50$

Эксперименты с ребрами графовых моделей

Попробуем увеличить диапазон соседей, соединяя нуклеотиды через m нуклеотидов после себя. Метрики моделей можно увидеть в таблице.

По результатам моделей можно заметить, что модель с соседями через одного хорошо улучшила свои показатели.

Вывод: по результатам экспериментов с ребрами можно сказать, что выгодно расставлять ребра так, чтобы у нуклеотида было влияние на более дальнее расстояние.

Модель GraphZSAGEConvBatchNorm с 90 соседями через одного, будем использовать далее.

	F1-score	ROC-AUC
Через одного	0.665	91.38%
Через 5	0.622	92.56%
Через 10	0.645	92.07%
Через 20	0.638	91.95%
Через 50	0.645	91.75%

Дистилляция знаний

Задача Дистилляции знаний — обучить меньшую модель (называемую учеником), которая работает лучше, чем сама, обученная с нуля, на основе использования большой эффективной модели (называемой учителем).

Обучение модели учителя проходит привычным способом.
Обучение ученика проходит со следующей функцией потерь :

$$student\_loss = \alpha * \underbrace{KD\_loss} + (1-\alpha) * \underbrace{CE\_loss}\ , \text{где } 0 < \alpha < 1$$

дивергенция Кульбака-Лейблера между мягкими предсказаниями модели ученика и модели учителя

Кросс-энтропия между предсказаниями модели ученика и истинными метками

В KD_loss вместо использования чистых предсказаний моделей ученика и учителя, используют “мягкие” предсказания, вычисленных по формулам:

$$soft\_teacher\_output = softmax(teacher\_output\ /\tau)$$
$$soft\_student\_output = softmax(student\_output\ /\tau)$$

- где, τ – температура, определяет коэффициент смягчения

Результаты экспериментов с дистилляцией

Эксперимент 1

Учитель: GraphZSAGEConvNorm_v2LSTM,
AUC-ROC = 92.01%, F1-score = 0.621
Ученик: DeepZ

Коэффициенты при обучении ученика: $\tau = 2$ и $\alpha = 0.5$

Метрики модели ученика после обучения с помощью дистилляции знаний:

AUC-ROC = 88.35%, F1-score = 0.5568

Итоговый ROC-AUC модели ученика выше ROC-AUC начальной модели DeepZ на **3%,** а итоговый F1-score выше на **16,58%.**

Эксперимент 2

Учитель: GraphZSAGEConvNorm,
AUC-ROC = 91.38%, F1-score = 0.665,
90 соседей у каждого нуклеотида через одного
Ученик: DeepZ
Коэффициенты при обучении ученика: $\tau = 2$ и $\alpha = 0.4$

Метрики модели ученика после обучения с помощью дистилляции знаний:

AUC-ROC = 87.20%, F1-score = 0.5430

Итоговый ROC-AUC модели ученика выше ROC-AUC начальной модели DeepZ на **2.2%,** а итоговый F1-score выше на **14,9%**.

Результаты

1. Проведение более 35 экспериментов, вошедших в итоговый отчет.
2. Эффективнее использовать меньший размер интервалов.
3. В графовых моделях эффективнее использование сверток GraphConv и SAGEConv.
4. В графовых моделях эффективнее использование больших диапазонов соседей для каждого нуклеотида.
5. Графовые модели справляются с задачей лучше чем модели с LSTM.
6. Лучшие F1-score и ROC-AUC у графовой модели равны 0.66 и 91.38% соответственно.
7. Лучшие F1-score и ROC-AUC у модели, которую можно проинтерпретировать, равен 0.59 и 88.11% соответственно.

Направления дальнейшей работы

1. Продолжение экспериментов с дистилляцией знаний
2. Получение доступа к серверам лаборатории
3. Продолжение экспериментов с ребрами
4. Увеличение количества признаков за счет добавления новых омиксных данных

Список использованных источников

1. Deep learning approach for predicting functional Z-DNA regions using omics data. [Электронный ресурс]// URL: https://www.nature.com/articles/s41598-020-76203-1 (Дата обращения: 07.04.2024)
2. Wang, A. H. et al. Molecular structure of a lef-handed double helical DNA fragment at atomic resolution. Nature 282, 680–686. (1979).
3. Liu, Rui; Liu, Hong; Chen, Xin; Martha, Kirby; O. Brown, Patrick; Zhao, Keji. Regulation of CSF1 Promoter by the SWI/SNF-like BAF Complex. Cell Journal, Volume 106, 309-318, 2001
4. van der Vorst, Emiel; Weber, Christian; Donners, Marjo. A Disintegrin and Metalloproteases (ADAMs) in Cardiovascular, Metabolic and Inflammatory Diseases: Aspects for Theranostic Approaches. Thrombosis and Haemostasis, 2018
5. Ho, P.S.; Ellison, M.J.; Quigley, G.J.; Rich, A. (1986). A computer aided thermodynamic approach for predicting the formation of Z-DNA in naturally occurring sequences.. The EMBO Journal, 5(10), 2737–2744
6. Champ P. C., Maurice S., Vargason J. M., Camp T., Ho P. S. Distributions of Z-DNA and nuclear factor I in human chromosome 22: a model for coupled transcriptional regulation. Nucleic Acids Res. 2004. — Vol. 32, no. 22. — P. 6501—6510
7. Графовые нейронные сети [Электронный ресурс]// URL: https://education.yandex.ru/handbook/ml/article/grafovye-nejronnye-seti (Дата обращения: 07.04.2024)
8. Репозиторий с кодом к работе [Электронный ресурс]// URL: https://github.com/LaptAAA/DNA(Дата обращения: 07.04.2024)
9. Understanding LSTM Networks [Электронный ресурс]// URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Дата обращения: 07.04.2024)
10. Knowledge Distillation: Principles, Algorithms, Applications. [Электронный ресурс]//URL: https://neptune.ai/blog/knowledge-distillation (Дата обращения: 07.04.2024)
11. Knowledge Distillation explained. [Электронный ресурс]//URL: https://www.kaggle.com/code/prameshgautam/knowledge-distillation-explained (Дата обращения: 07.04.2024)

Спасибо за внимание!

alapteva@edu.hse.ru