МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра математического моделирования и анализа данных

НОВИКОВ Антон Андреевич

АЛГОРИТМЫ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ПРЕДСКАЗАНИЯ СТРУКТУРЫ БЕЛКОВЫХ КОМПЛЕКСОВ

Магистерская диссертация
специальность 7-06-0533-05 «Прикладная математика и информатика»

Научный руководитель
Тузиков Александр Васильевич
доктор физико-математических наук,
профессор

Допущена к защите
«___» _____ 2025 г.
Зав. кафедрой математического
моделирования и анализа данных
Малюгин Владимир Ильич
доктор экономических наук, кандидат
физико-математических наук, профессор

Минск, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Задача предсказания взаимодействия белков популярна среди биоинформатиков и специалистов в области структурной биологии уже на протяжении нескольких десятков лет. В последнее десятилетие, вместе с ростом актуальности нейронных сетей, были предложены новые подходы для решения этой задачи [1].

В предыдущих работах [2] было предложено новое представление для белковых молекул – матрица косинусов. Данное представление использовалось в качестве входных и выходных данных полносвёрточной нейронной сети [3], предсказывавашей взаимодействие белок-белковой пары. В рамках данной работы предпринята попытка далее развить данное представление, чтобы получить возможность применить формализм теории поля для описания белковых взаимодействий, после чего применить подходы глубокого машинного обучения, для нахождения соответствующих функций поля. Таким образом, направление работы, в определённой степени, вернулось к классическому методу потенциалов.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Ключевые слова: кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g

Задачи исследования:

1. пункт 1
2. пункт 2

Цель работы: тут цель
Объект исследования является
Предмет исследования является
Методы исследования: методы методы
Результаты работы
Области применения

АГУЛЬНАЯ ХАРАКТЫРЫСТЫКА РАБОТЫ

Ключавыя словы: кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g

Мэта работы: тут цель

Задачи исследования:

1. пункт 1
2. пункт 2

Аб'ектам даследавання является

Метады даследавання методы методы

Вынікі работы

Вобласць ўжывання

GENERAL DESCRIPTION OF WORK

Keywords: кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g кейвордс dct ghbdt ndctgfd mfjgm jdfgm jmfg mjdjf gmjldfmjl gmdfl g

The object: тут цель

The objective:

1. item one
2. item two

Research methods: методы методы

The results

Application

ГЛАВА 1
ОПИСАНИЕ БЕЛКОВЫХ МОЛЕКУЛ И КОМПЛЕКСОВ И СВЯЗАННЫХ ЗАДАЧ

Полипептиды – биополимерные органические соединения, являющиеся основой для всей, известной на данный момент, жизни. Они состоят из аминокислотных остатков, связанных пептидной связью. Полипептиды условно разделяют на пептиды (состоят из менее чем 50 остатков) и белки (более чем 50 остатков). Далее, будем говорить в основном о белках. Части аминокислотных молекул, участвующие в пептидной связи, образуют главную цепь белка. К главной цепи белка присоединены многочисленные боковые цепи – части аминокислотных остатков, не участвующие в пептидной связи, но определяющие функциональную роль соответствующих участков белка. Существует 20 основных аминокислот, встречающихся повсеместно. Кроме 20 основных, существуют другие, более редкие аминокислоты, однако на практике, ими всегда пренебрегают. При изучении белков, выделяют 4 уровня структуры:

1. Первичный уровень – это линейная последовательность аминокислотных остатков. Эта последовательность устанавливается с секвенированием. С одной стороны, этого уровня, самого по себе, не достаточно для решения большинства задач обработки белковых соединений. Однако, данный уровень кодирует все более высокие структурные уровни. Кроме того, первичный уровень позволяет устанавливать эволюционную близость между двумя белковыми соединениями.

2. Вторичный уровень – последовательные участки главной цепи белковой молекулы склонны образовывать регулярные структуры: β-листы, α-спирали; нерегулярные структуры: изгибы и петли; а также неупорядоченные участки (которые можно считать нерегулярной последовательностью очень коротких регулярных участков).

3. Третичный уровень – полная трёхмерная структура молекулы. Включает в себя структуры вторичного уровня, их взаимное расположение, а также положение всех боковых цепей. Представляет собой основной интерес, так как позволяет строить фармакофорную модель, проводить докинг с малыми молекулами, проводить анализ молекулярной динамики и т.д. Трёхмерную структуру полипептидной модели называют её свёрткой.

4. Четвертичный уровень – существует для белковых молекул, образующих комплексы – соединения, образованные слабыми связями. Представляет собой третичные структуры всех компонент и их взаимное расположение. Часто встречаются комплексы, состоящие из нескольких одинаковых компонент. В зависимости от числа таких компонент,

выделяют различные виды симметрий [4]. Важным отличием белков от пептидов в контексте четвертичного уровня является то, что большие белковые молекулы почти не изменяются структурно при взаимодействии друг с другом. В то же время пептид, при взаимодействии с большой белковой молекулой, может значительно поменять собственную свёртку. Важными фундаментальными задачами являются предсказание сворачивания белка, предсказание взаимодействия белков и дизайн белков. Первая представляет собой предсказание третичной структуры по первичной; вторая – предсказание четвертичной структуры по третичной; третья – поиск первичной структуры, удовлетворяющей заданной третичной. Рассмотрим существующие методы и подходы к решению задачи предсказания взаимодействия.

1.1. Подходы к предсказанию взаимодействия

Выделим несколько групп методов, используемых для предсказания взаимодействия. Первой такой группой будут методы свободного докинга. В них могут использоваться эмпирические поля, скоринговые функции (в том числе, полученные с помощью машинного обучения). Эти поля и функции соответствуют свободной энергии комплекса. Минимум энергии должен соответствовать искомой структуре. Таким образом решается многомерная (в случае взаимодействия двух неодинаковых молекул – 6-мерная) задача оптимизации [5]. В результате работы алгоритма получают множество потенциально корректных моделей, которое ранжируется. Оценивают эффективность алгоритма на основании того, как высоко окажется правильная модель в полученном рейтинге.

Второй группой являются методы, основанные на белковой ко-эволюции. Эти методы изначально разрабатывались для предсказания сворачивания белка (это AlphaFold2 [6], AlphaFold-Multimer [7], новый AlphaFold3 [8] и вдохновленные им алгоритмы [9]) и оказались применимы и к задаче предсказания взаимодействия. Данные методы, используют информацию о родственных белках (предоставленную в явном виде или выученную), чтобы построить третичную и четвертичную структуру. И хотя, в задаче предсказания свёртки, такие алгоритмы показывают точность, сравнимую с экспериментальными методами (рентгеноструктурный анализ, ЯМР-спектроскопия и т.д.), с задачей предсказания взаимодействия они успешно справляются примерно в половине случаев, что, в свою очередь, является лучшим результатом в области.

Третьей группой обозначим методы на основе шаблонов [10]. В некотором смысле, такие алгоритмы являются классической версией алгоритмов второй группы. Этот подход так же применим к обеим задачам. Он основан на том наблюдении, что схожие участки в разных белках имеют тенденцию сворачивать/взаимодействовать сходно. Таким образом, имея базу известных

свёрток/взаимодействий, поискав в ней фрагменты неизвестного соединения можно восстановить его структуру.

К последней группе отнесём различные методы глубокого обучения, предсказывающие интерфейс взаимодействия [1]. Их можно разделить на несколько категорий в зависимости от того, какие архитектуры сетей используются и как представлены входные данные.

- Методы, использующие в качестве входных данных первичную структуру, используют свёрточные и рекуррентные сети.
- Методы, использующие представление, в которых используется трёхмерная структура, заданная через трёхмерные воксели, используют трёхмерные свёрточные нейронные сети.
- Методы, использующие в качестве входных данных поверхности белковых молекул [11], используют топологические свёрточные сети [12].
- Методы, использующие графовые представления белковых молекул, используют графовые нейронные сети.

В эту же категорию можно записать подход, при котором входными данными является представление структуры белка в виде двумерной карты расстояний [13]. При таком подходе используют полносвёрточные нейронные сети. Имея интерфейс взаимодействия, если необходимо, можно восстановить полную модель [14].

Подход, описываемый в данной работе, является развитием как раз последнего подхода. Далее, отметим несколько методов оценки качества предсказаний.

1.2. Оценка качества предсказаний

Конвенциональным методом оценки качества предсказания является метрика RMSD (root mean squared deviation) – среднеквадратическое отклонение [15], измеряемое в ангстремах. Обычно для расчёта этой метрики используют лишь Cα-атомы главной цепи. Для димерных комплексов обычно фиксируют одну из компонент, и считают отклонение атомов второй (предсказанной) компоненты от их истинной позиции. Модель считают допустимой, если её RMSD меньше 10 ангстрем. Допустимые модели считаются тем качественнее, чем меньше их RMSD.

При оценке алгоритмов, предсказывающих интерфейс взаимодействия, полные модели зачастую не строят. Вместо этого оценивают сами карты интерфейсов. Предсказание этих карт, по сути, представляет собой задачу бинарной классификации, поэтому их оценивают метриками, применимыми

в задачах бинарной классификации: точностью, полнотой, коэффициентом корреляции Мэтью [1], площадью под кривой ROC (AUROC) [16].

Существуют метрики оценки качества предсказаний, комбинирующие в себе два описанных выше метода. Такая метрика, например, используется в соревновании CAPRI [17]. Такие метрики используются потому, что нередки ситуации, когда хорошо предсказанный интерфейс порождает плохую трёхмерную модель.

1.3. Ранее полученные результаты

В предыдущих работах [2] был разработан подход, при котором белковые молекулы и комплексы кодировались с помощью матриц косинусов (см. *Главу 2*). Это позволяло представить входные данные в независимой от сдвигов и поворотов форме, а также получать выходные данные в такой же форме. На вход подавалась информация о первичной и третичной структуре молекул, а на выходе получалась информация о повороте молекул и направлении их взаимодействия, откуда довольнно легко получалась готовая четвертичная структура. Использовались полносвёрточные нейронные сети FCN5 (см. *Главу 5*). Результат был следующим:

- Обучающий набор данных содержал 23.344 комплексов, тестовый содержал 5.854;
- 67% обучающей выборки предсказывались корректно (70% гомодимеров и 64% гетеродимеров);
- 51% тестовой выборки предсказывалось корректно (59% гомодимеров и 45% гетеродимеров).

Дальнейший анализ (по методике [18]) показал, что хотя численные показатели из [2] выглядят хорошо, общее покрытие различных протеом (семейств белков) весьма бедно, что знначит, что нейронная сеть выучила наиболее распространённые (не в природе, но в банке PDB) семейства белковых комплексов, а значит использует не столько информацию о структуре молекул, сколько выученные ко-эволюционные представления. При поступлении на вход совершенно незнакомых сети белков, сеть производит совершенно неразборчивые матрицы косинусов, то есть описаннное выше явление может быть замечено даже при визуальном анализе выхода нейронной сети.

В рамках текущей работы, основным направлением работы было построение более широкой модели белок-белкового взаимодействия. Крайне желательно, чтобы такая модель могла предсказывать белок-пептидные взаимодействия (третичная структура пептидов часто меняется при взаимодействии с большой молекулой) и учитывать белок-лигандные взаимодействия.

Также желательно иметь возможность моделировать не только взаимодействие пары белков, но и комплексы более высокого порядка.

ГЛАВА 2
МАТРИЦЫ КОСИНУСОВ

Матрицы косинусов как математический объект были предложены ранее в рамках курсовых и дипломной работ. Теоретические результаты предыдущих работ здесь будут изложены, по возможности, кратко и без доказательств, но систематично. Здесь они необходимы, прежде всего, для правильного изложения новых результатов, касающихся нового объекта – релятивистской матрицы косинусов. Кроме того, терминология данной главы может местами отличаться от терминологии предыдущих работ, что связано с тем, что она ещё не устоялась, и требует некоторых изменений для большей систематичности.

2.1. Базовые определения

Пусть имеется некоторое гильбертово пространство $\mathcal{H}$ (в частном случае некоторое конечномерное евклидово пространство $\mathbb{R}^k$). Пусть в данном пространстве задана последовательность точек (ломаная) $X = (\vec{x}_i)$, где $i = \overline{0,n}$. Обозначим $\vec{p}_i = \vec{x}_i - \vec{x}_{i-1}$, где $i = \overline{1,n}$, тогда *ненормализованной матрицей косинусов* по ломаной X будем называть квадратную матрицу вида:

$$C_E = (< \vec{p}_i, \vec{p}_j >)_{ij} \in \mathbb{R}_{n,n}. \tag{2.1}$$

Если выполняется условие:

$$||\vec{p}_i|| = 1, i = \overline{1,n}, \tag{2.2}$$

то говорят о *нормализованных матрицах косинусов* или собственно *матрицах косинусов*. На практике, удобнее всего работать с нормализованными матрицами косинусов, ненормализованные матрицы косинусов могут возникнуть при некоторых операциях над обычными матрицами косинусов. Далее в работе, если не оговорено противное, будет подразумеваться наличие нормализации.

Название матриц следует из того факта, что при условии (2.2):

$$C_E = (< \vec{p}_i, \vec{p}_j >)_{ij} = (||\vec{p}_i||||\vec{p}_j|| \cos \vec{p}_i \wedge \vec{p}_j)_{ij} = (\cos \vec{p}_i \wedge \vec{p}_j)_{ij}. \tag{2.3}$$

Объект, определенный выше, строится по и описывает одну ломаную. Пусть имеется две последовательности точек: $X' = (\vec{x'}_i)$, где $i = \overline{0,m}$ и $X'' = (\vec{x''}_j)$, где $j = \overline{0,n}$. Тогда аналогично (2.1) можно построить матрицу косинусов по двум ломаным X' и X'':

$$C_{X',X''} = (< \vec{p'}_i, \vec{p''}_j >)_{ij} \in \mathbb{R}_{m,n}. \tag{2.4}$$

Определения (2.1) и (2.4) здесь фундаментальны, на практике полезны также матрица косинусов, кодирующая поворот ломаной C_A, и кодирующая вектор относительно ломаной C_ν – они являются частными случаями (2.4). Здесь определим C_ν, в следующей секции – C_A:

Пусть задан некоторый вектор $\vec{\nu} \in \mathbb{R}^k$ и $X = (\vec{x}_i)$ с $i = \overline{0,m}$, тогда *матрицей косинусов задающей вектор* назовём матрицу вида:

$$C_\nu = (< \vec{p}_i, \vec{\nu} >)_{ij} \in \mathbb{R}_{m,n}, \tag{2.5}$$

здесь $n \geq 1$ не зависит от ни от X, ни от $\vec{\nu}$, а выбирается из других практических соображений.

Непрерывным обобщение матрицы косинусов можно назвать *поверхностью косинусов*. Пусть $x(\alpha)$ - гладкая кривая $x : [0,l] \to \mathcal{H}$ и $\vec{p}(\alpha) = \frac{\partial x(\alpha)}{\partial \alpha}$. Тогда (2.1) можно переписать как:

$$C(\alpha_1, \alpha_2) =< \vec{p}(\alpha_1), \vec{p}(\alpha_2) > . \tag{2.6}$$

Таким образом, получаем действительную функцию от двух параметров

$$C : [0,l] \times [0,l] \to \mathbb{R}. \tag{2.7}$$

Условие нормализации для (2.6) выглядит следующим образом:

$$||\vec{p}(\alpha)|| = 1, \forall i \in [0,l]. \tag{2.8}$$

Использование непрерывной (2.6) для вычислений затруденено. Гораздо удобнее исследовать и использовать матрицы.

2.2. Свойства матриц косинусов

Для исследования свойств описанных матриц, необходимо записать их в более удобном виде. Пусть речь идёт об евклидовом простанстве $\mathbb{R}^k$, тогда в нём можно ввести репер $O = (\vec{x}_0, \vec{v}_1, ..., \vec{v}_k)$, пораждающий декартову систему координат. Тогда каждой точке $\vec{x}_i$ и вектору $\vec{p}_i$ можно поставить в соответствие координатный вектор-столбец $p_i \in \mathbb{R}_{k,1}$. Полученные столбцы координат p_i можно объединить в матрицу $P = [p_1, ..., p_n] \in \mathbb{R}_{k,n}$. Тогда (2.1) и (2.4) можно переписать как:

$$C_E = P^{\mathrm{T}} P, \tag{2.9}$$

$$C_{X_1,X_2} = {P_1}^{\mathrm{T}} P_2. \tag{2.10}$$

Теперь, имея некоторую матрицу $A \in \mathbb{R}_{k,k}$ можно определить *матрицу косинусов преобразования* A:

$$C_A = P^{\mathrm{T}} A P. \tag{2.11}$$

2.2.1. Базовые свойства

Приведём основные свойства матриц косинусов (без доказательств):

1. Все значения матриц косинусов лежат на отрезке $[-1, 1]$;
2. Все диагональные элементы матрицы C_E равны 1 (при отстутствии нормализации – $||\vec{p}_i||$);
3. Матрицы косинусов не зависят от выбора декартовой системы координат;
4. Ранг матрицы C_E равен размерности ломаной X;
5. Матрица C_E - симметричная. Все её собственные значения неотрицательны, и в сумме равны n;
6. Ломаную X (ломаные X_1 и X_2) можно восстановить из C_E (C_{X_1,X_2}) с точностью до выбора системы координат, причём как для нормированных, так и для ненормированных матриц (в случае последних с C_{X_1,X_2} необходимо знание длин векторов $\vec{p}_i$).

2.2.2. Восстановление исходных ломаных

Последнее свойство наиболее важно, так как оно позволяет использовать матрицы косинусов как инвариантное представление геометричеких ломаных (что крайне полезно при исследовании белковых молекул и их взаимодействий). Зная (2.9) и (2.10) легко построить соответсвующие матрицы. Ниже приведём без доказательств алгоритмы, которые позволяют строить ломаные по этим матрицам:

1. *Восстановление X из C_E* Из описанного выше свойства 5 следует, что матрица C_E будет обладать k неотрицательными собственными значениями $\lambda_1, ..., \lambda_k$ и собственными векторами $v_1, ..., v_k \in \mathbb{R}_{n,1}$. Тогда в некоторой системе координат матрица P примет вид:

$$P = \mathrm{diag}(\sqrt{\lambda_1}, ..., \sqrt{\lambda_k})[v_1, ..., v_k]^{\mathrm{T}}, \tag{2.12}$$

   где координаты точек ломаной X восстанавливаются из P через кумулятивную сумму. Формула (2.12) работает и для ненормализованных матриц.

2. *Восстановление X_2 из C_{X_1,X_2} при известном X_1* Введём следующую крайне важную матрицу:

$$T_P = (PP^{\mathrm{T}})^{-1}P, \tag{2.13}$$

данная матрица (с поправкой на транспонирование) представляет собой матрицу вычисления параметров линейной регрессии. Из (2.10) и (2.13) следует, что:

$$T_{P_1} C_{X_1,X_2} = (P_1 {P_1}^{\mathrm{T}})^{-1} P_1 {P_1}^{\mathrm{T}} P_2 = P_2, \tag{2.14}$$

данная формула позволяет посчитать P_2 из C_{X_1,X_2} при известном P_1, из которого, с помощью кумулятивной суммы, можно получить X_2. Формула (2.14), опять же, работает и для ненормализованных матриц. Также с помощью такого подхожа можно восстановить вектор $\vec{\nu}$ из (2.5).

3. *Восстановление A из C_A при известном X* Данная операция выполняется с помощью двойного применения матрицы (2.13):

$$T_P C_A T_P^{\mathrm{T}} = (PP^{\mathrm{T}})^{-1} P(P^{\mathrm{T}} A P) P^{\mathrm{T}} (PP^{\mathrm{T}})^{-1} = A, \tag{2.15}$$

4. *Восстановление X_1 и X_2 из C_{X_1,X_2}* Для выполнения данной операций существует следующий алгоритм, доказательство которого было приведено в предыдущей работе:

---

1. Пусть K_1 – матрица собственных векторов $C_{X_1,X_2} C_{X_1,X_2}^{\mathrm{T}}$;
2. Пусть K_2 – матрица собственных векторов $C_{X_1,X_2}^{\mathrm{T}} C_{X_1,X_2}$;
3. $L_1 \leftarrow K_1^{\mathrm{T}} C_{X_1,X_2}$;
4. $L_2 \leftarrow (C_{X_1,X_2} K_2)^{\mathrm{T}}$;
5. $M \leftarrow T_{L_2} C_{X_1,X_2} T_{L_1}^{\mathrm{T}}$;
6. Пусть N – матрица собственных векторов MM^{T};
7. $O \leftarrow N^{\mathrm{T}} M L_2$;
8. $s \leftarrow T_{[O]^2} e$, где e – единичный вектор-столбец;
9. $\hat{P}_1 = \mathrm{diag}(\mathrm{sqrt}(s)) O \,\mathrm{diag}(||\vec{p'}_1||, ..., ||\vec{p'}_m||)$;
10. $\hat{P}_2 = T_{\hat{P}_1} C_{X_1,X_2}$.

Описанные выше алгоритмы могут работать не только с истинными матрицами косинусов, но и с некоторыми приближёнными/зашумлёнными матрицами. Для этого достаточно производить нормализации по $||\vec{p}_i||$ там, где возникают матрицы P. В предыдущих работах было показана устойчивость описанных методов при добавлении случайных шумов.

2.2.3. Арифметические свойства матриц косинусов

Сложение матриц C_E

Пусть имеется две ломаные X' в $\mathbb{R}^{k_1}$ и X'' в $\mathbb{R}^{k_2}$ имеющие соответствующие $P' = [p'_1, ..., p'_n] \in \mathbb{R}_{k_1,n}$ и $P'' = [p''_1, ..., p''_n] \in \mathbb{R}_{k_2,n}$. Найдём сумму

соответствующих им матриц C'_E и C''_E:

$$C'_E + C''_E = P'^{\mathrm{T}} P' + P''^{\mathrm{T}} P'' = \begin{bmatrix} P' \\ P'' \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} P' \\ P'' \end{bmatrix} = P^{+\mathrm{T}} P^{+} = C^{+}_E, \tag{2.16}$$

где получили новую ломанную $X^+ = X' \oplus X''$ в $\mathbb{R}^{k_1+k_2}$, а также

$$P^+ = \begin{bmatrix} p'_1 & \dots & p'_n \\ p''_1 & \dots & p''_n \end{bmatrix} \in \mathbb{R}_{k_1+k_2,n}. \tag{2.17}$$

Таким образом сумма двух матриц косинусов – ненормализованная матрица косинусов, которой соответствует ломаная, являющаяся прямой суммой ломаных-операнд.

Вычитание матриц C_E

Используя обозначения из (2.16) запишем:

$$C'_E - C''_E = P'^{\mathrm{T}} P' - P''^{\mathrm{T}} P'' = \begin{bmatrix} P' \\ iP'' \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} P' \\ iP'' \end{bmatrix} = P^{-\mathrm{T}} P^{-} = C^{-}_E. \tag{2.18}$$

Применяя разность в общем случае, придётся выйти из поля действительных чисел, и перейти к полю комплексных чисел. При том, что матрица C^-_E всё еще действительная, порождающая её ломаная $X^- = X' \oplus iX''$ уже принадлежит $\mathbb{C}^{k_1+k_2}$. Для матрицы, полученной из (2.18), в общем случае не будет выполняться свойство о неотрицательности собственных значений. Из-за появления отрицательных собственных значений, при восстановлении X^- из C^-_E необходимо переписать (2.12) как:

$$P^- = \mathrm{diag}(\sqrt{\lambda^+_1}, ..., \sqrt{\lambda^+_{k_1}}, i\sqrt{-\lambda^-_1}, ..., i\sqrt{-\lambda^-_{k_2}})[v^+_1, ..., v^+_{k_1}, v^-_1, ..., v^-_{k_2}]^{\mathrm{T}}. \tag{2.19}$$

Произведение матриц косинусов

Раннее было показано, что произведение матриц косинусов может быть использовано для некоторых приближённых вычислений. Причины такого поведения, а также исследование точности приближений здесь опустим, приведём лишь сами приближённые выражения.

Пусть $C_{X_1,X_2} \in \mathbb{R}_{m,n}$ и $C_{X_2,X_3} \in \mathbb{R}_{n,l}$, тогда:

$$C_{X_1,X_2} C_{X_2,X_3} \approx \frac{n}{k} C_{X_1,X_3}. \tag{2.20}$$

Выражение (2.20) приводит к следующим выражениям:

- $C_A C_B \approx \frac{n}{k} C_{AB}$;
- Для ортогональных $A^{\mathrm{T}} = A^{-1}$ выполняется $C_A^{\mathrm{T}} C_A \approx C_A C_A^{\mathrm{T}} \approx \frac{n}{k} C_E$;
- $C_E C_{X_1,X_2} \approx \frac{m}{k} C_{X_1,X_2}$;
- $C_E^2 \approx \frac{n}{k} C_E$.

2.3. Релятивистская матрица косинусов

В данном разделе представлен полностью новый результат, касающийся матриц косинусов. *Релятивистская матрица косинусов* C_{rel} отличается от обычной C_E тем, что позволяет закодировать не только лишь некоторую последовательность точек, но точку, из которой эта последовательность наблюдается. Таким образом получается инвариантное относительно выбора декартовой системы координат (а вообще говоря – Лоренц-инвариантное) представление пары точка-ломаная, что позволяет задавать функции (поля), порождаемые ломаной (белковой молекулой), как преобразования матриц косинусов. Для релятивистских матриц будет дано полное определение, в соответствии со специальной теорией относительности (СТО), и приближённая запись, используемая на практике.

2.3.1. Полное определение

Пусть существует некоторая последовательность точек X, точки которой перемещаются в пространстве $\mathbb{R}^k$ со скоростями, не превышающими скорость света c:

$$X(t) = (\vec{x}_l(t)), l = \overline{0,n}; \tag{2.21}$$
$$\vec{x}_l : \mathbb{R} \to \mathbb{R}^k;$$
$$||\frac{\partial \vec{x}_l}{\partial t}|| < c.$$

Следуя Минковскому [19], в пространстве-времени можно ввести систему координат, в которой точке $\vec{x}_l$ в момент времени t будет поставлен в соответствие вектор-столбец $x_l = [x_l^1(t), ..., x_l^k(t), ict]^{\mathrm{T}} \in \mathbb{C}_{k+1,1}$.

Пусть имеется некоторая точка $\vec{x}_{ref}$ и момент времени t_{ref}, из которых наблюдается ломаная X. Этой точке в пространстве-времени соответствуют координаты $x_{ref} = [x_{ref}^1(t), ..., x_{ref}^k(t), ict_{ref}]^{\mathrm{T}}$. Каждая точка из X будет наблюдаться в $\vec{x}_{ref}$ в момент t_{ref} при пересечении траектории $\vec{x}_l(t)$ и светового конуса, порождённого $\vec{x}_{ref}$ и t_{ref}. Из того, что все скорости не превышают c, следует, что такое пересечение единственно, и для каждой точки $\vec{x}_l$ произойдёт в некоторый момент времени $t_l^* \leq t_{ref}$. Таким образом, в $\vec{x}_{ref}$ в момент t_{ref} будет наблюдаться X^* – образ ломаной $X(t)$ вида: $X^* = (\vec{x}_l^*)$, где $l = \overline{0,n}$ и координаты $\vec{x}_l^*$ равны $x_l^* = [x_l^1(t_l^*), ..., x_l^k(t_l^*), ict_l^*]^{\mathrm{T}}$.

Если теперь посчитать $\vec{p_l} = \vec{x_l^*} - \vec{x_{l-1}^*}$ и использовать произведение Минковского, (2.1) или (2.9) можно получить *релятивистскую матрицу косинусов*:

$$C_{rel} = (<\vec{p_l}, \vec{p_j}>)_{lj} = \left(\sum_{s=1}^{k} p_l^s p_j^s - c^2 (t_l^* - t_{l-1}^*)(t_j^* - t_{j-1}^*) \right)_{lj} \in \mathbb{R}_{n,n}. \quad (2.22)$$

Отметим, что требование нормализации (2.2) для релятивистских матриц косинусов будет иметь вид:

$$\sum_{s=1}^{k} (p_l^s)^2 = 1; l = \overline{1,n}, \quad (2.23)$$

т.е. единичную длину должны быть только первые k действительных компонент p_l – мнимая компонента p_l^{k+1} может быть любой.

2.3.2. Упрощённая формулировка

Теперь, для упрощения, положим, что $\forall l : \vec{x}_l(t) = \vec{x}_l = \text{const}$. Также, не нарушая общности, положим $t_{ref} = 0$. При таких предположениях вычисление пересечения светового конуса и траекторий заметно упростится, и можно будет записать:

$$t_l^* = t_{ref} - \frac{||\vec{x}_l - \vec{x}_{ref}||}{c} = -\frac{1}{c} \sqrt{\sum_{s=1}^{k} (x_l^s - x_{ref}^s)^2}, \quad (2.24)$$

и на основании (2.24) получим следующие значения координат:

$$x_l^* = \begin{bmatrix} x_l^1 \\ \dots \\ x_l^k \\ -i\sqrt{\sum_{s=1}^{k} (x_l^s - x_{ref}^s)^2} \end{bmatrix}. \quad (2.25)$$

По (2.25) можно посчитать $P^* = \begin{bmatrix} P \\ -id \end{bmatrix} \in \mathbb{C}_{k+1,n}$ и по (2.9) посчитать:

$$C_{rel} = \begin{bmatrix} P \\ -id \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} P \\ -id \end{bmatrix} = P^{\mathrm{T}} P - d^{\mathrm{T}} d = C_E - d^{\mathrm{T}} d \in \mathbb{R}_{n,n}. \quad (2.26)$$

В случае, когда $||\frac{\partial \vec{x}_l}{\partial t}|| \ll c$ разница между $\vec{x}_l(t_l^*)$ и $\vec{x}_l(t_{ref})$ будет иметь порядок вычислительной погрешности, а следовательно использование (2.22) неоправдано. Далее свойства матриц косинусов будем исследовать исключи-

тельно на основе (2.26).

Можно видеть, что по аналогии с обычной C_E можно получить матрицы аналогичные C_{X_1,X_2}, C_A, C_ν:

$$C_{rel,X_1,X_2} = P_1^{*\mathrm{T}} P_2^* = \begin{bmatrix} P_1 \\ -id_1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} P_2 \\ -id_2 \end{bmatrix} = C_{X_1,X_2} - d_1^{\mathrm{T}} d_2 \in \mathbb{R}_{m,n}, \tag{2.27}$$

$$C_{rel,A} = P_1^{*\mathrm{T}} A P_2^* \in \mathbb{C}_{n,n}. \tag{2.28}$$

Если матрица $A \in \mathbb{C}_{k+1,k+1}$ имеет структуру:

$$A = \begin{bmatrix} A_{11} & iA_{12} \\ iA_{21} & -A_{22} \end{bmatrix}, \tag{2.29}$$

где $A_{11} \in \mathbb{R}_{k,k}$, $A_{12} \in \mathbb{R}_{k,1}$, $A_{21} \in \mathbb{R}_{1,k}$, $A_{22} \in \mathbb{R}_{1,1}$; то для (2.28) можно гарантировать:

$$\begin{aligned} C_{rel,A} = &\begin{bmatrix} P \\ -id \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} A_{11} & iA_{12} \\ iA_{21} & -A_{22} \end{bmatrix} \begin{bmatrix} P \\ -id \end{bmatrix} = \\ &= C_{A_{11}} + d^{\mathrm{T}} A_{12} P + P^{\mathrm{T}} A_{21} d + d^{\mathrm{T}} A_{22} d \in \mathbb{R}_{n,n}. \end{aligned} \tag{2.30}$$

Аналогично, можно построить $C_{rel,\nu}$ и гарантировать, что она будет действительной, если $\nu = [\nu_1, ..., \nu_k, i\nu_{k+1}]^{\mathrm{T}}$ ($\nu_l \in \mathbb{R}$).

Матрицы $C_{rel,A}$ и $C_{rel,\nu}$ позволяют кодировать матрицы и векторы в пространстве-времени, также как и C_A, C_ν кодировали матрицы и векторы в евклидовом пространстве. Кроме того, аналогично (2.6) можно ввести релятивистскую поверхность косинусов. Этот объект пригодится для доказательства одного из свойств C_{rel}.

2.3.3. Свойства релятивистских матриц косинусов

Свойства релятивистских матриц аналогичны свойствам обычных матриц:

1. Все значения C_{rel} лежат на отрезке $[-2, 2]$;
2. Все диагональные элементы матрицы C_{rel} лежат на отрезке $[0, 1]$;
3. Релятивистские матрицы косинусов не меняются от преобразований Лоренца (см. (3.2));
4. Матрица C_{rel} - симметричная. Одно её собственное значение отрицательно, остальные – неотрицательны.

Приведём доказательство первых двух свойств, а потом отметим особенности восстановления структур из описанных матриц.

Первое доказательство дадим для непрерывного случая. Пусть $\vec{x}(\alpha)$ - гладкая кривая $\vec{x} : [0, l] \to \mathbb{R}^k$, и пусть, не нарушая общности, она наблюдается из $x_{ref} = [0, 0, 0, 0]^{\mathrm{T}}$. Тогда координаты вектора $\vec{x}(\alpha)$ будут иметь вид:

$$x(\alpha) = \begin{bmatrix} x_{real}(\alpha) \\ -i||x_{real}(\alpha)|| \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ -i\sqrt{x_1^2 + x_2^2 + x_3^2} \end{bmatrix}, \tag{2.31}$$

и тогда:

$$p(\alpha) = \frac{\partial x(\alpha)}{\partial \alpha} = \begin{bmatrix} \frac{\partial x_1}{\partial \alpha} \\ \frac{\partial x_2}{\partial \alpha} \\ \frac{\partial x_3}{\partial \alpha} \\ -i\frac{x_1\frac{\partial x_1}{\partial \alpha} + x_2\frac{\partial x_2}{\partial \alpha} + x_3\frac{\partial x_3}{\partial \alpha}}{\sqrt{x_1^2+x_2^2+x_3^2}} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_{real}(\alpha)}{\partial \alpha} \\ -i\cos\left(x_{real}(\alpha) \wedge \frac{\partial x_{real}(\alpha)}{\partial \alpha}\right) \end{bmatrix}. \tag{2.32}$$

Условие нормализации (2.23) тут имеет вид:

$$||\frac{\partial x_{real}}{\partial \alpha}|| = 1. \tag{2.33}$$

Теперь, аналогично (2.6):

$$C_{rel}(\alpha_1, \alpha_2) =< p(\alpha_1), p(\alpha_2) >= \cos\left(\frac{\partial x_{real}(\alpha_1)}{\partial \alpha} \wedge \frac{\partial x_{real}(\alpha_2)}{\partial \alpha}\right) - \\ - \cos\left(x_{real}(\alpha_1) \wedge \frac{\partial x_{real}(\alpha_1)}{\partial \alpha}\right)\cos\left(x_{real}(\alpha_2) \wedge \frac{\partial x_{real}(\alpha_2)}{\partial \alpha}\right). \tag{2.34}$$

Из (2.34) очевидно следует, что:

$$\forall \alpha_1, \alpha_2 : -2 \leq C_{rel}(\alpha_1, \alpha_2) \leq 2, \tag{2.35}$$

а для случая $\alpha_1 = \alpha_2$:

$$C_{rel}(\alpha, \alpha) = 1 - \cos^2\left(x_{real}(\alpha) \wedge \frac{\partial x_{real}(\alpha)}{\partial \alpha}\right) = \sin^2\left(x_{real}(\alpha) \wedge \frac{\partial x_{real}(\alpha)}{\partial \alpha}\right), \tag{2.36}$$

$$\forall \alpha : 0 \leq C_{rel}(\alpha, \alpha) \leq 1. \tag{2.37}$$

Формула (2.34) крайне любопытна, так как она даёт альтернативную аналитическую форму записи поверхности косинусов, имеющую довольно понятное трактование. Предложенное доказательство верно для гладкого случая, однако, если речь идёт о ломаной (и, соотвественно, матрице, а не поверхности), следует дать другое доказательство, которое, впрочем, также

довольно тривиальное.

Пусть имеется $X = (\vec{x}_l)$, где $l = \overline{0,n}$, которая будет наблюдаться из точки O. Для $\forall l \in \{1,...,n\}$ можно обозначить $A = \vec{x}_{l-1}$ и $B = \vec{x}_l$. Пусть $\overline{OA} = d$ и $\overline{OB} = d + \Delta$. Предполагая нормализацию, будем также иметь $\overline{AB} = 1$. По известному свойству треугольников, будем иметь:

$$\begin{aligned} \overline{OB} &\leq \overline{OA} + \overline{AB}, \\ d + \Delta &\leq d + 1, \\ \Delta &\leq 1, \end{aligned} \tag{2.38}$$

и

$$\begin{aligned} \overline{OA} &\leq \overline{OB} + \overline{AB}, \\ d &\leq d + \Delta + 1, \\ -1 &\leq \Delta. \end{aligned} \tag{2.39}$$

Из (2.38) и (2.39) следует $|\Delta| \leq 1$. Тогда C_{rel} запишется как:

$$C_{rel} = \left(\begin{bmatrix} p_l^{real} \\ -i\Delta_l \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} p_j^{real} \\ -i\Delta_j \end{bmatrix} \right)_{lj} = (p_l^{real\,\mathrm{T}} p_j^{real} - \Delta_l \Delta_j)_{lj}, \tag{2.40}$$

откуда аналогично (2.34)-(2.37) получаем первые 2 свойства.

Восстановление исходных ломаных из релятивистских матриц

Формула (2.15), а также (2.14) (для $C_{rel,\nu}$) применяются здесь так же, как и для обычных матриц косинусов. В то же время применение (2.12) затруднено, и хотя формула (2.19), казалось бы, здесь применима, но наличие элемента $-i\sqrt{\sum_{s=1}^{k}(x_l^s - x_{ref}^s)^2}$ в (2.25) приводит к тому, что требуются некоторые дополнительные преобразования для согласования координат точек $\vec{x}_l$ и их расстояний до точки наблюдения. Задача несколько упрощается, если известна структура ломаной, и необходимо восстановить только позицию точки наблюдения.

Хотя описанную выше задачу можно сформулировать и решать в терминах задачи оптимизации, эффективного алгоритма её решения пока не предложено. Эксперименты показали, что её можно эффективно решать с помощью нейросетевых алгоритмов.

ГЛАВА 3
МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ БЕЛКОВ

Цель данной главы – предложить класс физических моделей, которые потенциально могут описывать взаимодействие белковых молекул. В предыдущих работах [2] предлагалось напрямую предсказывать комформацию белковых комплексов с помощью полносвёрточных нейронных сетей, и матрицы косинусов использовались в качестве инвариантной кодировки таких конформаций. В данной работе предлагается подход, близкий к классическому методу потенциалов.

3.1. Лагранжев формализм классической релятивистской теории поля

Опираясь на [19] кратко изложим суть данного формализма. Пусть имеется четырёхмерное пространство-время (трёхмерное пространство + время). Пусть в данном пространстве можно задать n обобщённых переменных:

$$u_1(x_1, x_2, x_3, x_4), ..., u_n(x_1, x_2, x_3, x_4); \tag{3.1}$$

где $x_4 = ict$. У этих переменных можно вычислить частные производные по каждой из координат: $\frac{\partial u_1}{\partial x_1}, ..., \frac{\partial u_n}{\partial x_4}$. Также могут использоваться производные более высоких порядков, но в большинстве теорий обходятся лишь первым порядком.

Зададим действительнозначную $L(u_1, ..., u_n, \frac{\partial u_1}{\partial x_1}, ..., \frac{\partial u_n}{\partial x_4})$, известную как *функция Лагранжа* или *лагранжиан*. В релятивистской теории на лагранжиан накладывается требование *Лоренц-инвариантности*.

Преобразования Лоренца – это такие преобразования, которые сохраняют скалярные произведения в пространстве времени, т.е.:

$$< f(x), f(y) >=< x, y >, \tag{3.2}$$

при условии, что для пространства-веремени скалярное произведение записывается как:

$$< x, y >= x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4 = x_1y_1 + x_2y_2 + x_3y_3 - c^2t_xt_y. \tag{3.3}$$

Из курса линейной алгебры известно, что преобразование f, обладающее свойством (3.2), является изометрическим, и ему будет соответствовать четырёхмерная комплексная ортогональная матрица A ($A^{\mathrm{T}} = A^{-1}$) (не стоит путать с унитарной матрицей ($\overline{A}^{\mathrm{T}} = A^{-1}$)).

Если обобщённые координаты (3.1) задают скаляры, то их производные будут векторами, если переменные задают векторы, то их производные будут

матрицами. При преобразовании, заданном матрицей A ($x \to Ax$), векторы и матрицы, входящие в лагранжиан, также будут изменяться ($v \to Av$; $M \to AMA^{\mathrm{T}}$). Требование Лоренц-инвариантности заключается в том, что преобразования Лоренца не должны изменять значение функции Лагранжа.

Если имеется Лоренц-инвариантный лагранжиан, то можно задать *функцию действия*:

$$S = \int_{spacetime} L(u_1, ..., u_n, \frac{\partial u_1}{\partial x_1}, ..., \frac{\partial u_n}{\partial x_4})\, dx_1\, dx_2\, dx_3\, dx_4. \tag{3.4}$$

Поведение системы, таким образом, описывается с помощью *принципа стационарного действия*:

$$S \to \min . \tag{3.5}$$

Задачи типа (3.2) обычно решают с помощью вариационного исчисления. Из принципов вариационного исчисления и инвариантности относительно преобразований (3.2) следует теорема Нётер и автоматически выводимые из неё фундаментальные физические законы сохранения. Поэтому при моделировании белковых взаимодействий, довольно привлекательно выглядит свойство Лоренц-инвариантности, так как с помощью него можно попытаться избежать явного учёта законов сохранения.

3.2. Общая модель взаимодействия белков с применением релятивистских матриц косинусов

Белковые молекулы являются полимерными молекулами. Они являются линейными последовательностями аминокислотных остатков, соединённых пептидными связями. В каждом из остатков находится атом Cα. В общем случае, знания пространственного расположения этих атомов и знания соответствующих аминокислтных остатков достаточно, чтобы полностью задать структуру белковой молекулы. Кроме того, расстояние между соседними Cα-атомами является практически константным, и равно 3,8Å. Эти факты способствуют тому, чтобы можно было использовать матрицы косинусов для задания белковых структур. Введение релятивистской матрицы косинусов (2.22) позволяет задать как структуру молекулы, так и некоторую точку пространства-времени относительно неё. При этом, важным достоинством релятивистской матрицы является её Лоренц-инвариантность. Действительно, легко видеть, что от скалярных произведений, составляющих C_{rel} в (2.22), в (3.2) требуется инвариантность. Таким образом, если обычная матрица косинусов была инвариантна относительно изометрических преобразований в обычном евклидовом пространстве, то релятивистская матрица инвариантна относительно преобразований Лоренца в пространстве-времени. Следовательно, можно задавать обобщенные функции поля как функционалы над

C_{rel}. Также необходимо отметить, что, так как C_{rel} содержит в себе закодированным, в том числе, расстояние между точкой наблюдения и наблюдаемыми атомами, но делает это в неявной форме и без обращения этих расстояний, то при использовании описываемого подхода можно обойти проблему бесконечного роста функций поля при сближении атома и точки наблюдения.

Теперь дадим формальное описание данной модели в наиболее общем случае. Для начала, запишем:

$$C_{rel} = C(X, \vec{x}_{ref}) \in \mathbb{R}_{m,m}, \tag{3.6}$$

где X - последовательность точек – Cα-атомов описываемой молекулы в пространстве-времени, $\vec{x}_{ref}$ – точка наблюдения, которой могут соответствовать координаты $x_{ref} = [x^1_{ref}, x^2_{ref}, x^3_{ref}, x^4_{ref}] = [x^1_{ref}, x^2_{ref}, x^3_{ref}, ict_{ref}]$. Тогда (3.1) можно выразить с помощью некоторого набора функционалов, принимающих на вход матрицу и данные об аминокислотной последовательности:

$$u_l = F_l(C_{rel}, A) : \mathbb{R}_{m,m} \oplus \mathbb{A} \to \mathbb{C}; l = \overline{1, n}. \tag{3.7}$$

Тогда функция действия (3.4) для одной белковой молекулы запишется как:

$$S = \int_{spacetime} L(F_1(C(X, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}), A), ..., \frac{\partial F_1(C(X, x), A)}{\partial x_1}, ...)\, dx_1\, dx_2\, dx_3\, dx_4. \tag{3.8}$$

Если (3.4) параметризовывалась обощенными функциями поля (т.е. решением были значения поля во всех точках пространства времени), то (3.8) параметризуется ломаной $X(t)$ в пространстве-времени. Поэтому принцип стационарного действия (3.2) запишется как:

$$X^{sol} = \operatorname{argmin}_X S. \tag{3.9}$$

Выражения (3.8) и (3.9) описывают одну белковую молекулу, и, таким образом, задают модель её свёртки (без явного учёта растворителя и внешних сил). Данные выражения легко обобщить на случай взаимодействия двух молекул, и далее на любое число белковых молекул, и даже на лиганды (для них, впрочем, в общем случае подход с релятивистской матрицей косинусов и функционалом не подходит из-за нелинейности). Запишем модель для двух белковых молекул, описывамых парами (X_1, A_1) и (X_2, A_2):

$$u_l(x) = F_l(C(X_1, x), A_1) + F_l(C(X_2, x), A_2), l = \overline{1, n}; \tag{3.10}$$

$$S = \int_{spacetime} L(F_1(C(X_1, x), A_1) + F_1(C(X_2, x), A_2), ..., \frac{\partial F_1(C(X_1, x), A_1)}{\partial x_1} + \frac{\partial F_1(C(X_2, x), A_2)}{\partial x_1}, ...) \, dx_1 \, dx_2 \, dx_3 \, dx_4; \tag{3.11}$$

$$(X_1^{sol}, X_2^{sol}) = \operatorname{argmin}_{X_1, X_2} S. \tag{3.12}$$

Таким образом получили наиболее общую модель описания взаимодействия. При отстутствии явных выражений для (3.7) аналитическое исследование такой системы невозможно. Впрочем, даже при наличии таких выражений, оно скорее всего было бы крайне сложным. Численный анализ и поиск таких функционалов также крайне затруднён, особенно с учётом того, что модель описывает полную динамику белковой молекулы, а экспериментальных данных такого рода де-факто не существует. Экспериментально получены лишь статические модели белковых молекул, а динамику получают с помощью расчётов, выполненных на основе теоретических соображений.

3.3. Упрощённая модель взаимодействия белков

Упростим предложенную модель по следующим пунктам:

1. Так как скорости движения белковых молекул крайне малы, по сравнению со скоростью света, то вместо полной релятивистской матрицы косинусов (2.22) следует использовать (2.26).

2. Так как функционалы (3.7) на практике будут аппроксимациями, полученными с помощью машинного обучения, то использование частных производных от таких аппроксимаций может давать весьма сомнительные результаты (что связано с тем, что оператор дифференцирования не является непрерывным в теории функционального анализа), поэтому опустим все явные вхождения частных производных в лагранжиан.

3. Так как вместе с этим опустится производная по времени, то, тем самым, будет получена статическая модель для белковых взаимодействий.

4. Для снижения количества степеней свободы, можно положить, как часто делают, что большие белковые молекулы жёсткие. Тогда они будут моделироваться как абсолютно твёрдые тела, и будут параметризованы своим поворотом и сдвигом в пространстве.

Применяя данные упрощения, мы более не будем иметь дело с релятивистской моделью, однако упрощённая модель всё еще может считаться некоторым классическим механическим приближением такой общей модели.

Рассмотрим последствия предложенных упрощений:

$$X_1(t) = \text{const}, \tag{3.13}$$
$$X_2(t) = \text{const}, \tag{3.14}$$
$$X_1 = X_1(\alpha), \tag{3.15}$$
$$X_2 = X_2(\alpha), \tag{3.16}$$

где α - обобщённый параметр, кодирующий взаимную конформацию молекул X_1 и X_2. Выражение (3.7) остаётся без изменений. Функция действия для одной молекулы примет вид:

$$S_1(\alpha) = \int_{\mathbb{R}^3} L(F_1(C(X_1(\alpha), x), A_1), ..., F_n(C(X_1(\alpha), x), A_1))\, dx_1\, dx_2\, dx_3 = \\ = \text{const}, \tag{3.17}$$

что обусловлено тем, что повороты и сдвиги не приведут к изменению суммарной функции действия. Функция действия двух молекул будет иметь вид:

$$S(\alpha) = \int_{\mathbb{R}^3} L(F_1(C(X_1(\alpha), x), A_1) + F_1(C(X_2(\alpha), x), A_2), ..., \\ F_n(C(X_1(\alpha), x), A_1) + F_n(C(X_2(\alpha), x), A_2))\, dx_1\, dx_2\, dx_3. \tag{3.18}$$

Принцип стационарного действия примет вид:

$$\alpha^{sol} = \text{argmin}_\alpha\, S. \tag{3.19}$$

Из-за константности (3.17) можно ввести *частичный лагранжиан*:

$$L_{partial}(\alpha; x) = L(F_1(C(X_1(\alpha), x), A_1) + F_1(C(X_2(\alpha), x), A_2), ...) - \\ - L(F_1(C(X_1(\alpha), x), A_1), ...) - L(F_1(C(X_2(\alpha), x), A_2), ...) = \\ = L(\alpha; x) - L_1(\alpha; x) - L_2(\alpha; x), \tag{3.20}$$

вычислив интеграл по которому, получим:

$$\int_{\mathbb{R}^3} L_{partial}(\alpha; x)\, dx_1\, dx_2\, dx_3 = S(\alpha) - \text{const} - \text{const}, \tag{3.21}$$

а следовательно:

$$\alpha^{sol} = \text{argmin}_\alpha \int_{\mathbb{R}^3} L_{partial}(\alpha; x)\, dx_1\, dx_2\, dx_3 = \text{argmin}_\alpha\, S. \tag{3.22}$$

Использование (3.20) будет полезно с численной точки зрения, а также весьма удобно при определённой степени линейности лагранжианов.

3.4. Возможные варианты выбора функции Лагранжа

3.4.1. Скалярное поле

В случае отсутствия частных производных, лагранжиан для скалярного поля u примет вид:

$$L(u) = au^2, \tag{3.23}$$

где a - некоторая константа. Система ведёт себя принципиально разно при положительных и отрицательных a.

Модель с таким лагранжианом крайне бедна. Несколько обогатить её можно если использовать не одно, а n скалярных полей: $U = [u_1, ..., u_n]^{\mathrm{T}}$. Тогда имея некоторую постоянную $A \in \mathbb{R}_{n,n}$ запишем:

$$L(U) = U^{\mathrm{T}} A U. \tag{3.24}$$

Таким образом получаем мультискалярное поле, функция Лагранжа которого есть квадратичная форма. Мультискалярное поле отличается от векторного поля тем, что оно не меняется при преобразованиях Лоренца (ну или хотя бы при поворотах пространства). Не нарушая общности, для квадратичных форм, можно считать A – симметричной ($A = A^{\mathrm{T}}$). Частичный лагранжиан будет иметь вид:

$$\begin{aligned} L_{partial}(U_1, U_2) &= L(U_1 + U_2) - L(U_1) - L(U_2) = \\ &= (U_1 + U_2)^{\mathrm{T}} A (U_1 + U_2) - U_1^{\mathrm{T}} A U_1 - U_2^{\mathrm{T}} A U_2 = \\ &= U_1^{\mathrm{T}} A U_2 + U_2^{\mathrm{T}} A U_1 = U_1^{\mathrm{T}} (A + A^{\mathrm{T}}) U_2 = \\ &= 2 U_1^{\mathrm{T}} A U_2. \end{aligned} \tag{3.25}$$

Таким образом, частичный лагранжиан – билинейная форма.

Матрица A выражает взаимодействие различных полей друг с другом. Так как она симметричная, её можно диагонализировать (получим диагональную A'), и тем самым получить набор новых полей $U' = [u'_1, ..., u'_n]^{\mathrm{T}}$, которые не будут взаимодействовать друг с другом, а только с самими собой. Вообще говоря, абсолютные значения элементов A' не важны, важны только их знаки, поэтому можно задать (3.24) как:

$$L(U') = U'^{\mathrm{T}} A' U' = \begin{bmatrix} U^+ \\ U^- \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} E & 0 \\ 0 & -E \end{bmatrix} \begin{bmatrix} U^+ \\ U^- \end{bmatrix}, \tag{3.26}$$

где лишние скалярные поля можно полагать нулевыми.

Функционал, способный задать скалярное поле, можно задать различными способами. Эксперименты показали, что наиболее практичный путь - следующий: имея скаляр u, будем задавать вектор $v = [0, 0, 0, -iu]^{\mathrm{T}}$. Тогда

(3.7), с помощью некоторого преобразования G, можно представить так:

$$G : \mathbb{R}_{m,m} \oplus \mathbb{A} \to \mathbb{R}_{m,m}, \tag{3.27}$$

$$C_{rel,v} = G(C_{rel}, A), \tag{3.28}$$

$$u = F(C_{rel}, A) = \begin{bmatrix} 0 & 0 & 0 & i \end{bmatrix} T_X G(C_{rel}, A) \begin{bmatrix} \frac{1}{m} \\ \dots \\ \frac{1}{m} \end{bmatrix}. \tag{3.29}$$

Такое задание скалярное поля - это частный случай задания векторного поля, где собственно векторная часть нулевая. При таком задании, повороты в пространстве не изменят поля, но преобразования Лоренца в общем случае могут вернуть векторную компоненту, что, впрочем, не является большой проблемой, так как (3.23) можно переписать как:

$$L(v) = au^2 = a < v, v > . \tag{3.30}$$

3.4.2. Векторное поле

Векторное поле задаётся вектором $v = [u_1, u_2, u_3, u_4]^{\mathrm{T}} = [v_1, v_2, v_3, iv_4]^{\mathrm{T}}$. Такое векторное поле не является чисто векторным, но также содержит в себе и скалярное. Его лагранжиан можно задать с помощью (3.30) (легко видеть, что лагранжиан Лоренц-инвариантен) и аналогично (3.24) обобщить на мультивекторный случай. Формулу (3.29) можно переписать так:

$$v = F(C_{rel}, A) = T_X G(C_{rel}, A) \begin{bmatrix} \frac{1}{m} \\ \dots \\ \frac{1}{m} \end{bmatrix}. \tag{3.31}$$

В общем и целом векторные поля, заданные таким образом, мало отличаются от скалярных. Главным отличием здесь является чувствительность к поворотам в пространстве, что добавляет некоторых трудностей при обучении задаче предсказания взаимодействия белков. При этом модель оказывается не на много лучше мультискалярной модели.

3.4.3. Поле, заданное матрицей

Для описания данного поля, следует, для начала, обратиться к общей формулировке векторного поля [19]:

$$A(x_1, x_2, x_3, x_4) = \begin{bmatrix} A_1(x) \\ A_2(x) \\ A_3(x) \\ A_4(x) \end{bmatrix}. \tag{3.32}$$

Для (3.32) имеется следующий вид лагранжиана:

$$L(x) = \alpha \sum_i A_i^2 + a \sum_{ij} \left(\frac{\partial A_i}{\partial x_j} \right)^2 + b \sum_i \left(\frac{\partial A_i}{\partial x_i} \right)^2 + c \sum_{ij} \frac{\partial A_i}{\partial x_j} \frac{\partial A_j}{\partial x_i}. \quad (3.33)$$

Если записать матрицу производных:

$$\nabla A(x_1, x_2, x_3, x_4) = \begin{bmatrix} \frac{\partial A_1}{\partial x_1} & \frac{\partial A_1}{\partial x_2} & \frac{\partial A_1}{\partial x_3} & \frac{\partial A_1}{\partial x_4} \\ \frac{\partial A_2}{\partial x_1} & \frac{\partial A_2}{\partial x_2} & \frac{\partial A_2}{\partial x_3} & \frac{\partial A_2}{\partial x_4} \\ \frac{\partial A_3}{\partial x_1} & \frac{\partial A_3}{\partial x_2} & \frac{\partial A_3}{\partial x_3} & \frac{\partial A_3}{\partial x_4} \\ \frac{\partial A_4}{\partial x_1} & \frac{\partial A_4}{\partial x_2} & \frac{\partial A_4}{\partial x_3} & \frac{\partial A_4}{\partial x_4} \end{bmatrix}, \quad (3.34)$$

то (3.33) можно записать в матричном виде как:

$$L(x) = \alpha A^{\mathrm{T}} A + a \operatorname{tr} \nabla A^{\mathrm{T}} \nabla A + b \operatorname{tr} [\nabla A]^2 + c \operatorname{tr} \nabla A^2. \quad (3.35)$$

Для электро-магнитного поля, согласно [19], (3.33) принимает вид:

$$L(x) = -\frac{1}{2} \left(\sum_{ij} \left(\frac{\partial A_i}{\partial x_j} \right)^2 - \sum_{ij} \frac{\partial A_i}{\partial x_j} \frac{\partial A_j}{\partial x_i} \right), \quad (3.36)$$

что в матричном виде записывается как:

$$L(x) = -\frac{1}{2} \left(\operatorname{tr} \nabla A^{\mathrm{T}} \nabla A - \operatorname{tr} \nabla A^2 \right) = -\frac{1}{2} \operatorname{tr} \left((\nabla A^{\mathrm{T}} - \nabla A) \nabla A \right). \quad (3.37)$$

Теперь, если имеется некоторая матрица B, которую аналогично (3.31) записать как:

$$B = F(C_{rel}, A) = T_X G(C_{rel}, A) T_X^{\mathrm{T}} = T_X C_{rel,B} T_X^{\mathrm{T}}, \quad (3.38)$$

после чего переписать (3.37) как:

$$L(x) = -\left(\operatorname{tr} B^{\mathrm{T}} B - \operatorname{tr} B^2 \right) = \operatorname{tr} \left((B - B^{\mathrm{T}}) B \right). \quad (3.39)$$

Легко видеть, что выражение (3.39) – Лоренц-инвариантно:

$$L(x) = \operatorname{tr} \left(S B S^{\mathrm{T}} \right)^2 - \operatorname{tr}(S B^{\mathrm{T}} S^{\mathrm{T}})(S B S^{\mathrm{T}}) = \operatorname{tr} B^2 - \operatorname{tr} B^{\mathrm{T}} B. \quad (3.40)$$

С практической точки зрения, более интересен частичный лагранжиан этого поля, получаемый от взаимодействия двух матричных полей A и B:

$$L_{partial}(A, B) = L(A + B) - L(A) - L(B), \quad (3.41)$$

$$
\begin{aligned}
L(A+B) &= \operatorname{tr}\left((A+B-A^{\mathrm{T}}-B^{\mathrm{T}})(A+B)\right) = \\
&= \operatorname{tr}\left(((A-A^{\mathrm{T}})+(B-B^{\mathrm{T}}))(A+B)\right) = \\
&= \operatorname{tr}\left((A-A^{\mathrm{T}})A\right) + \operatorname{tr}\left((B-B^{\mathrm{T}})B\right) + \\
&+ \operatorname{tr}\left((A-A^{\mathrm{T}})B\right) + \operatorname{tr}\left((B-B^{\mathrm{T}})A\right) = \\
&= L(A) + L(B) + \operatorname{tr} AB + \operatorname{tr} BA - \operatorname{tr} A^{\mathrm{T}}B - \operatorname{tr} B^{\mathrm{T}}A = \\
&= L(A) + L(B) + 2\operatorname{tr} AB - 2\operatorname{tr} A^{\mathrm{T}}B = \\
&= L(A) + L(B) + 2\operatorname{tr}\left((A-A^{\mathrm{T}})B\right)
\end{aligned} \tag{3.42}
$$

$$
L_{partial}(A,B) = 2\operatorname{tr}\left((A-A^{\mathrm{T}})B\right). \tag{3.43}
$$

Легко видеть некоторую однородность между (3.39) и (3.43).

Модели, задаваемые (3.39) получаются наиболее интересными. Также, как и с (3.24), можно усложнить модель, используя сумму нескольких полей. По своему построению, данное поле крайне близко к электромагнитному, но не требует использования операции дифференцирования. Кроме того, кодировка матрицы с помощью $C_{rel,B}$ оказвается более естественным, чем кодировки вектора с помощью $C_{rel,v}$.

3.5. Интегрирование по интерфейсу

Описанные сверху лагранжианы, должны, в общем случае (3.4) – интегрироваться по всему пространству-времени, а в упрощённом (3.18) – интегрироваться по всему пространству. В следующей главе будет показано, как можно приближённо выполнять подобное интегрирование. Гораздо проще выполнять не интегрирование по всему пространству, а выполнять суммирование по интерфейсу.

Если имеются модели двух взаимодействующих молекул (x', a') и (x'', a''), где x – координаты Cα-атомов и a – тип аминокислотного остатка, то интерфейс взаимодействия данных молекул можно определить как:

$$
I = \{(i,j) : ||x'_i - x''_j|| < d\}, \tag{3.44}
$$

где d – некоторая заранее выбранная граница, для которой все более близко находящиеся остатки считаются взаимодействующими. Тогда (3.4) можно посчитать так:

$$
S = \sum_{i,j \in I} L(\frac{x'_i + x''_j}{2}; a'_i; a''_j), \tag{3.45}
$$

или, если добавить некоторые веса w разным типам взаимодействующих остатков:

$$
S = \sum_{i,j \in I} w_{a'_i, a''_j} L(\frac{x'_i + x''_j}{2}; a'_i; a''_j). \tag{3.46}
$$

Использование (3.45) или (3.46) заметно упрощает процесс моделирования взаимодействия. Однако с чисто формальной точки зрения, использование некоторого фиксированного расстояния d для выявления взаимодействий не вполне сочетается с СТО, а также является большим упрощением по отношению к природе белок-белковых взаимодействий. Впрочем, для вычислительных экспериментов с упрощённой моделью взаимодействия белков, использование (3.45) или (3.46) вполне обосновано.

ГЛАВА 4
ВСПОМОГАТЕЛЬНЫЙ МАТЕМАТИЧЕСКИЙ АППАРАТ

В рамках данной главы описаны методы, которые применялись для компьютерной реализации теоретических моделей, изложенных в предыдущей главе.

4.1. Конечномерная аппроксимация поля в пространстве

Согласно (3.1), для моделирования физической системы необходимо уметь задавать некоторые обобщённые переменные в пространсте-времени (пространстве). В данной работе основной формой такого задания являются нейронные сети (по схеме (3.7)). Однако, для численного поиска минимума (3.9) необходимо много раз производить интегрирование, и каждое интегрирование требует своего определённого набора точек. В то же время, использование нейронной сети для вычисления в каждой такой точке может быть крайне трудоёмко, поэтому вполне логично, имея нейронную сеть и некоторую белковыую молекулу, попробовать построить аппроксимацию поля, создаваемого данной молекулой, и задаваемого нейронной сетью. Использование подобной аппроксимации, хотя и само по себе трудоёмко, может в несколько раз ускорить процесс оптимизации (3.9). Далее описан подход, позволяющий с определённой долей успеха выполнить вышеописанную задачу.

4.1.1. Определение

Для простоты начнём с одномерного случая. Пусть имеется непрерывная функция $f(x)$, $x \in \mathbb{R}$ такая, что $\exists \int_{-\infty}^{+\infty} f(x)\,dx \in \mathbb{R}$ и $\lim_{x\to\pm\infty} f(x) = 0$. С помощью сигмоиды, переменной $x \in \mathbb{R}$ можно поставить в соответствие $t \in [0, 1]$:

$$t = \frac{1}{1 + e^{-x}}. \tag{4.1}$$

Обратное преобразование выполняется с помощью функции логит:

$$x = \ln \frac{t}{1 - t}. \tag{4.2}$$

Зададим непрерывную функцию $g(t) = f(x) = f(\ln \frac{t}{1-t})$. Из свойств f ясно, что:

$$g(0) = g(1) = 0, \tag{4.3}$$

а также, что $\int_0^1 g(t)\,dt \in \mathbb{R}$.

Из-за (4.3) функция g может быть продолжена как нечётная на $[-1,1]$, а следовательно – быть разложена в ряд Фурье по синусам:

$$g(t) = \sum_{i=1}^{\infty} a_i \sin \pi i t, \tag{4.4}$$

где:

$$a_i = 2 \int_0^1 g(t) \sin \pi i t \, dt. \tag{4.5}$$

Ряд (4.4) сходится равномерно. Через него можно записать функцию $f(x)$:

$$f(x) = \sum_{i=1}^{\infty} a_i \sin \frac{\pi i}{1+e^{-x}}, \tag{4.6}$$

а коэффициент a_i перепишется как:

$$a_i = 2 \int_0^1 f(\ln \frac{t}{1-t}) \sin \pi i t \, dt = 2 \int_{-\infty}^{+\infty} f(x) \sin(\frac{\pi i}{1+e^{-x}}) \frac{e^{-x}}{(1+e^{-x})^2} \, dx. \tag{4.7}$$

Раз (4.4) сходится равномерно, то и (4.6) сходится равномерно.

Далее будет необходимо воспользоваться двумя известными функциями:

- $\mathrm{Si}(x) = \int_0^x \frac{\sin t}{t} \, dt$ – интегральный синус;
- $\mathrm{Cin}(x) = \int_0^x \frac{1-\cos t}{t} \, dt$ – интегральный косинус.

Если в (4.4) имеем дело с разложением функции $g(t)$ по ортонормированному базису из функций $2 \sin \pi i t$, то в случае с (4.6) последовательность функций $\sin \frac{\pi i}{1+e^{-x}}$ не является ни нормированной, ни попарно ортогональной. Рассчитаем основные интегралы, связанные с (4.6):

$$\begin{aligned}
&\int_{-\infty}^{+\infty} \sin \frac{\pi i}{1+e^{-x}} \, dx = \int_0^1 \sin \pi i t \, d \ln \frac{t}{1-t} = \int_0^1 \frac{\sin \pi i t}{t(1-t)} \, dt = \\
&= \int_0^1 \frac{\sin \pi i t}{t} \, dt + \int_0^1 \frac{\sin \pi i t}{1-t} \, dt = \int_0^1 \frac{\sin \pi i t}{t} \, dt - \int_0^1 \frac{\sin(\pi - \pi i \xi)}{\xi} \, d\xi = \\
&= \int_0^1 \frac{\sin \pi i t}{t} \, dt - (-1)^i \int_0^1 \frac{\sin \pi i \xi}{\xi} \, d\xi = (1 + (-1)^{i+1}) \, \mathrm{Si}(\pi i).
\end{aligned} \tag{4.8}$$

Формула (4.8) позволяет при наличии коэффициентов (4.5) посчитать интеграл функции $f(x)$ по всей числовой прямой. Пусть имеются две функции $f^1(x)$ и $f^2(x)$, которые разлагаются по (4.6) с коэффициентами a_i^1 и a_j^2. Крайне полезно уметь считать интеграл $\int_{-\infty}^{+\infty} f^1(x) f^2(x) \, dx$ (сравните с (3.18) и (3.25)), что было бы крайне легко, если бы последовательность функций $\sin \frac{\pi i}{1+e^{-x}}$ была

ортонормированной, но так как это не так, необходимо рассчитать интегралы попарных произведений функций этой последовательности:

$$\int_{-\infty}^{+\infty} \sin\frac{\pi i}{1+e^{-x}} \sin\frac{\pi j}{1+e^{-x}}\,dx = \int_0^1 \sin\pi it \sin\pi jt\,d\ln\frac{t}{1-t} =$$
$$= \int_0^1 \frac{\cos\pi(i-j)t - \cos\pi(i+j)t}{2}\left(\frac{1}{t}+\frac{1}{1-t}\right)\,dt = \frac{1}{2}(\int_0^1 \frac{1-\cos\pi(i+j)t}{t}\,dt -$$
$$- \int_0^1 \frac{1-\cos\pi(i-j)t}{t}\,dt + \int_0^1 \frac{\cos\pi(i-j)t - \cos\pi(i+j)t}{1-t}\,dt) = \frac{1}{2}(\mathrm{Cin}(\pi(i+j)) -$$
$$- \mathrm{Cin}(\pi(i-j)) + \int_0^1 \frac{(-1)^{i-j}\cos\pi(i-j)\xi - (-1)^{i+j}\cos\pi(i+j)\xi}{\xi}\,d\xi) =$$
$$= \frac{1+(-1)^{i+j}}{2}\left(\mathrm{Cin}(\pi(i+j)) - \mathrm{Cin}(\pi(i-j))\right). \tag{4.9}$$

Из (4.9) при $i = j$ получим:

$$\int_{-\infty}^{+\infty} \sin^2\frac{\pi i}{1+e^{-x}}\,dx = \mathrm{Cin}(2\pi i). \tag{4.10}$$

Так как в рамках исследуемой задачи, необходимо аппроксимировать функции заданные на трёхмерном, а не одномерном пространстве, то следует перейти к трёхмерному ряду Фурье. Для такого ряда формулу (4.6) перепишем как:

$$f(x,y,z) = \sum_{i,j,k=1}^{\infty} A_{i,j,k} \sin\frac{\pi i}{1+e^{-x}} \sin\frac{\pi j}{1+e^{-y}} \sin\frac{\pi k}{1+e^{-z}}, \tag{4.11}$$

где:

$$A_{i,j,k} = 2^3 \iiint_{\mathbb{R}^3} \frac{f(x,y,z)\sin\frac{\pi i}{(1+e^{-x})^2}\sin\frac{\pi j}{(1+e^{-y})^2}\sin\frac{\pi k}{(1+e^{-z})^2} e^{-x-y-z}}{(1+e^{-x})^2(1+e^{-y})^2(1+e^{-z})^2}\,d^3x. \tag{4.12}$$

Легко видеть, что формулы (4.8), (4.9) и (4.10) полезны и для трёхмерного случая.

4.1.2. Вычисление коэффициентов по методу Монте-Карло

При построении аппроксимации, во-первых, приходится ограничиваться некоторым конечным числом слагаемых ряда, а во-вторых, необходимо численно получать значения коэффициентов (4.12). Из-за относительно большой размерности функции, и необходимости считать интегралы по всему пространству, интегрирование по сетке точек не так удобно – проще применить

метод Монте-Карло.

Для метода Монте-Карло необходимо иметь некоторую трёхмерную случайную величину $\xi = [\xi_x, \xi_y, \xi_z]^{\mathrm{T}}$, принимающую все значения из $\mathbb{R}^3$ с распределением вероятности $p(x, y, z)$ ($p > 0$, $\iiint_{\mathbb{R}^3} p(x, y, z)\, dx\, dy\, dz = 1$). На роль такой случайной величины хорошо подходит многомерное нормальное распределение. Хорошо известно, что для любой функции $f(x, y, z)$ можно записать:

$$\iiint_{\mathbb{R}^3} f(x, y, z)\, dx\, dy\, dz = \iiint_{\mathbb{R}^3} \frac{f(x, y, z)}{p(x, y, z)} p(x, y, z)\, dx\, dy\, dz = \mathbb{E}\left[\frac{f(\xi)}{p(\xi)}\right], \tag{4.13}$$

из чего, при наличии N выборочных значений ξ^i, через статистическую оценку среднего получаем:

$$\iiint_{\mathbb{R}^3} f(x, y, z)\, dx\, dy\, dz = \mathbb{E}\left[\frac{f(\xi)}{p(\xi)}\right] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\xi^i)}{p(\xi^i)}. \tag{4.14}$$

Использование формулы (4.14) совместно с (4.12) даёт искомую аппроксимацию.

Достоинством описанного выше метода является, как минимум, то, что он работает, и позволяет с некоторой точностью записать многомерную, заданную на всём бесконечном множестве функцию, используя конечное число параметров. Следует также привести недостатки данного метода:

1. При ограниченном числе членов ряда, точность приближения крайне чувствительна к сдвигам приближаемой функции ($f(x, y, z) \to f(x + c_x, y + c_y, z + c_z)$). Это выражается в том плане, что не следует сразу использовать (4.12), вместо этого следует искать такое преобразование исходных координат – желательно линейное $x = A\hat{x} + b$, для которого отклонение аппроксимации $\hat{f}$ от исходной f будет минимальным:

$$A'_{i,j,k} = 2^3 \iiint_{\mathbb{R}^3} \frac{f(Ax + b) \sin \frac{\pi i}{(1+e^{-(a_{11}x+a_{12}y+a_{13}z+b_1)})^2} \cdot \ldots \cdot e^{-((a_{11}+\ldots)x+\ldots)}}{(1 + e^{-(a_{11}x+a_{12}y+a_{13}z+b_1)})^2 \cdot \ldots} d^3x, \tag{4.15}$$

$$\hat{f}(x, y, z) = \det(A) \sum_{i,j,k=1}^{\infty} A'_{i,j,k} \sin \frac{\pi i}{1 + e^{-(a_{11}x+a_{12}y+a_{13}z+b_1)}} \cdot \ldots, \tag{4.16}$$

и

$$\iiint_{\mathbb{R}^3} \left(f(x) - \hat{f}(x)\right)^2 d^3x \to \min. \tag{4.17}$$

Поиск (4.17) эффективно выполняется с помощью методов градиентного спуска.

2. Точность приближения также сильно зависит от дисперсии распределения ξ. Если дисперсия слишком мала, она может не покрыть всех особенностей функции f.

3. Увеличение количества учитываемых членов ряда Фурье приводит к увеличению точности, но в многомерном случае приводит к очень быстрому росту трудоёмкости вычислений, а формулы (4.8), (4.9) и (4.10) становятся малоприменимыми.

4. Метод Монте-Карло обладает высокой дисперсией, его точность растёт со скоростью $O(\frac{1}{\sqrt{N}})$. Вместе с разложением в ряд Фурье и недостаточно большими значениями N приводят к бвстрому накоплению ошибок.

4.1.3. Графические примеры

Приведём несколько графических примеров описанного выше подхода для аппроксимации одномерных функций и двумерных изображений.

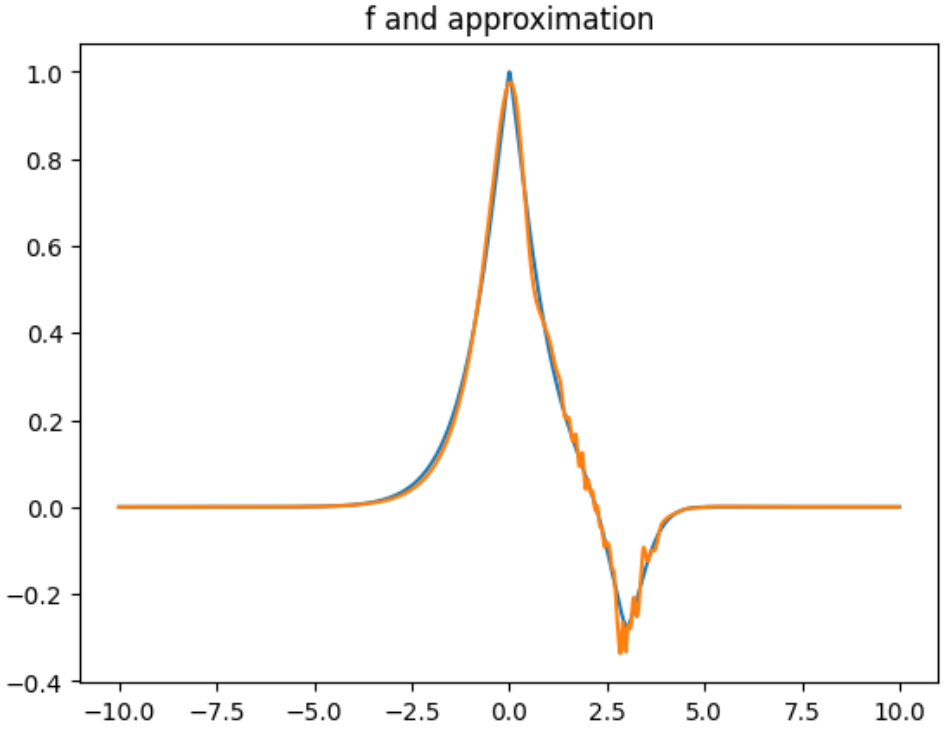

Рис. 4.1. Приближение функции $f(x) = e^{-|x|^{1,2}} - 0,3e^{-2|x-3|^{1,5}}$.

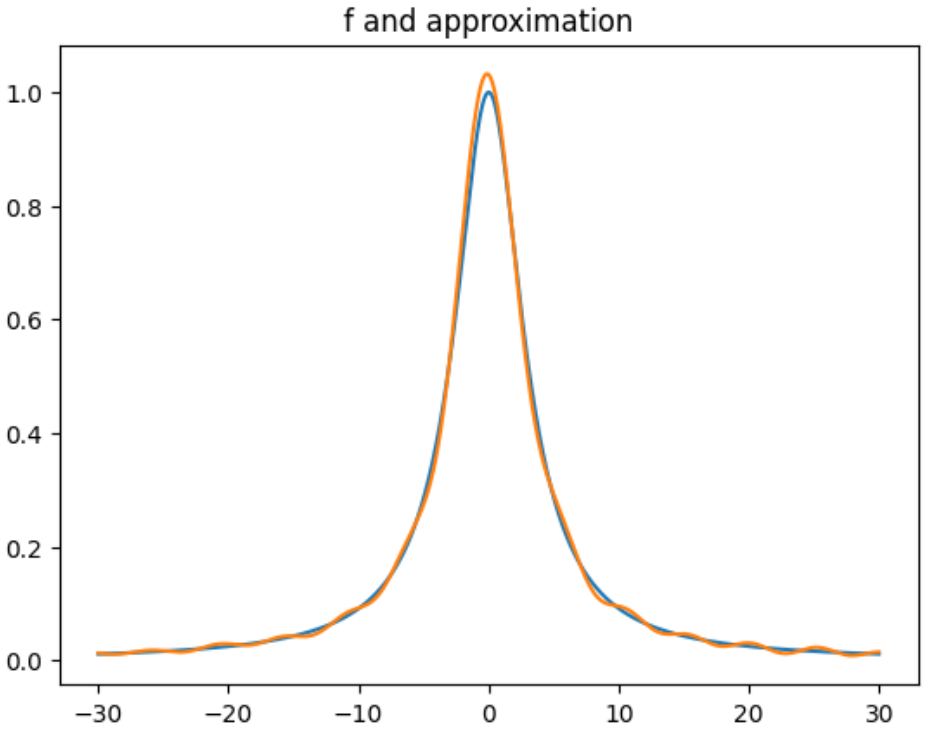

Рис. 4.2. Приближение функции $f(x) = \frac{10}{10+x^2}$.

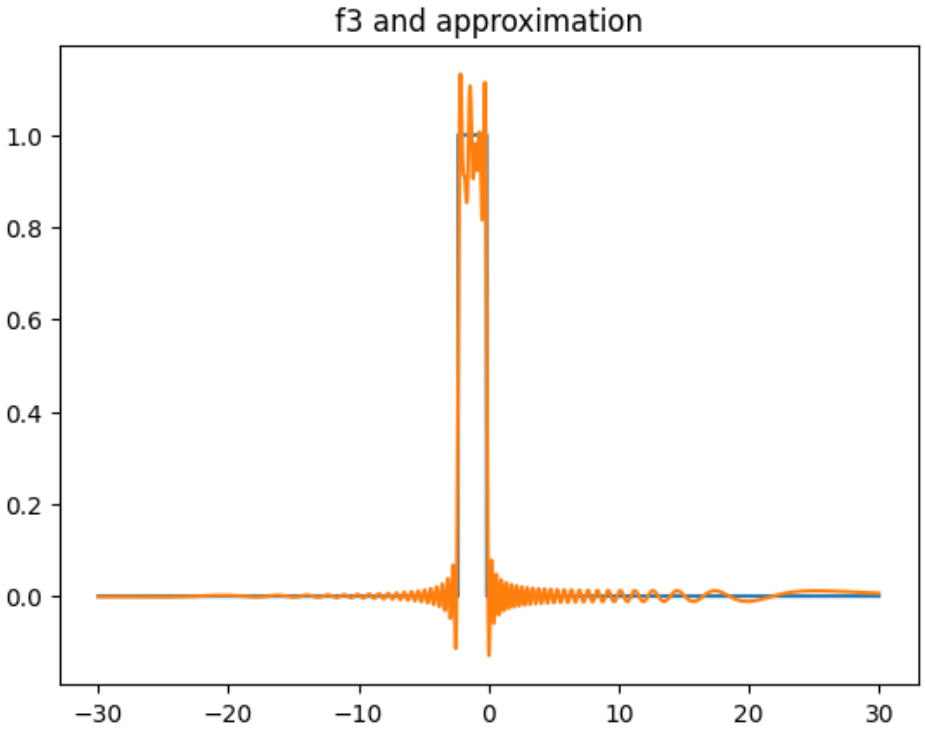

Рис. 4.3. Приближение функции $f(x) = \mathbb{I}[|x + 1,25| < 1,1]$.

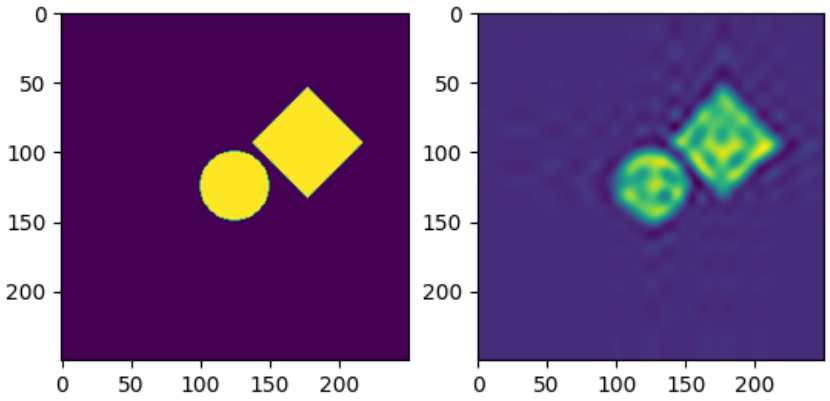

Рис. 4.4. Приближение круга и квадрата.

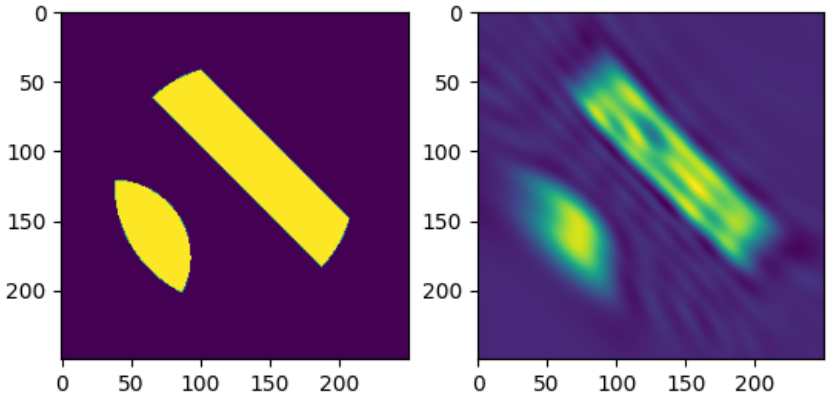

Рис. 4.5. Приближение пересечения круга с объединением круга и прямоугольника.

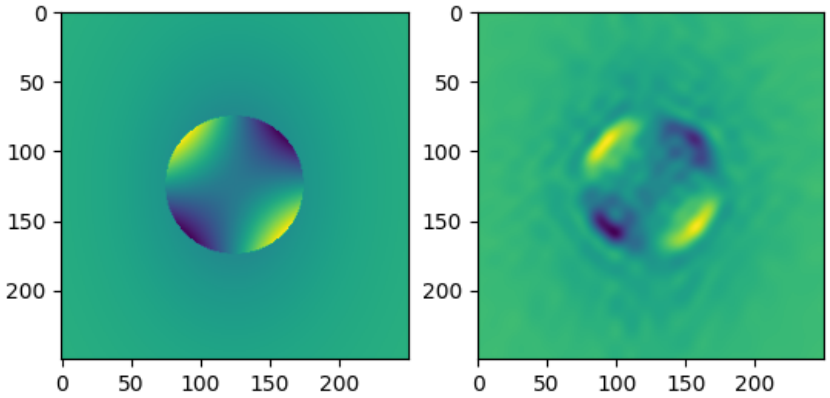

Рис. 4.6. Приближение функции $f(x,y) = \frac{xy}{10}\mathbb{I}[|x^2+y^2|<4] - \frac{1}{10+x^2+y^2}$.

4.2. Параметризация пары взаимодействующих молекул

Взаимодействие двух белковых молекул проще всего моделировать как взаимодействие двух жёстких тел. В общем случае, все конформации получаемого комплекса можно записать с помощью 6 параметров: одна компонента

фиксируется, для второй компоненты выбирается центр (3 параметра) и поворот (3 параметра). В случае, когда компоненты идентичны (гомодимерные комплексы), в силу определённой симметрии, можно обойтись всего 4 параметрами.

Воспользовавшись физическими соображениями, можно попробовать сократить необходимое число параметров на 1. Для этого можно воспользоваться *операцией «сталкивания»* [2]. Она заключается в том, чтобы найти такое минимальное расстояние между молекулами, при котором минимальное расстояние между Cα-атомами не превышает некоторой границы (ван-дер–ваальсовых радиусов атомов углерода – 3,4Å или, например, 3,6Å). Ранее, такая операция выполнялась с помощью итеративного процесса, при котором молекулы раздвигаются с некоторым постоянным шагом, пока условие нарушения границ не будет выполнено. В рамках данной работы операция выполняется с помощью метода дихотомии.

При использовании моделей типа (3.4), необходимо выполнение интегрирования по всему пространству, что может быть выполнено с помощью метода Монте-Карло (4.14). Поиск решения (3.19) заключается в итеративном интегрировании (4.14), изменении параметров комплекса, дальнейшем интегрировании, изменении параметров и т.д. Если при каждом интегрировании (4.14) случайная выборка ξ^i, $i = \overline{1, N}$ генерируется заново, то возникает дополнительная дисперсия, которая мешает процессу оптимизации. Чтобы избежать этой проблемы, можно сгенерировать единое ξ^i, $i = \overline{1, N}$ случайное облако и применять его на всех итерациях. Впрочем, возникает проблема такого рода, что облако такое фиксированное поле может не покрывать всех возможных конформаций комплекса.

Чтобы как-то обойти данную проблему, можно использовать другую параметризацию. В ней центры молекул задаются через параметр r как: $[-r, 0, 0]^{\mathrm{T}}$ и $[r, 0, 0]^{\mathrm{T}}$. Поворот каждой из молекул задаётся 3 параметрами, что приводит к тому, что итоговое количество параметров равно 7 (если использовать операцию сталкивания для поиска параметра r при заданных поворотах, то можно сократить количество параметров до 6). Тогда случайную величину ξ можно генерировать из многомерного нормального распределения, достаточно протяжённого относительно оси $[1, 0, 0]^{\mathrm{T}}$.

4.3. Алгоритмы оптимизации

В рамках данного исследования были использованы различные методы оптимизации. При обучении нейронный сетей, а также при решении (4.17), используется популярный алгорит стохастической оптимизации Adam [20]. Впрочем, при решении (3.19) данный алгоритм оказался малоприменим. Во–первых, при моделировании часто использовались плохо дифференцируемые функции. Во-вторых, из-за наличия малой относительно малого (по сравне-

нию с нейронными сетями) количества параметров, алгоритм не способен достичь глобального минимума. Для решения данных проблем, предлагается использовать эволюционные методы оптимизации [21].

4.3.1. Простой эволюционный алгоритм

Был использован простой эволюционный алгоритм следующего вида. Пусть p – вектор параметров (кодирует форму взаимодействие), $f(p)$ – минимизируемая функция (частичный лагранжиан), $g(p)$ – бинарная функция, сигнализирующая некорректные параметры (функция, сигнализирующая нарушение радиусов ван-дер-ваальса):

$P \leftarrow [p_1, p_2, ..., p_N]$ из многомерного равномерного распределения с заданными границами;
$p_{best} \leftarrow null$;
$f_{best} \leftarrow +\infty$;
$iter \leftarrow 0$;
while $iter < I$ **do**
$P_{rand} \leftarrow [p_1, p_2, ..., p_{N_{rand}}]$ из многомерного равномерного распределения с заданными границами;
$P_{mut} \leftarrow P + [\delta p_1, \delta p_2, ..., \delta p_{N_{mut}}]$, где δp_i из многомерного нормального распределения с заданной дисперсией;
$P_{parent_1} \leftarrow [p_{i_1}, p_{i_2}, ..., p_{i_{N_{parent}}}]$, где i_j – случайная выборка из популяции;
$P_{parent_2} \leftarrow [p_{j_1}, p_{j_2}, ..., p_{j_{N_{parent}}}]$ (аналогично);
$P_{cross} \leftarrow \alpha P_{parent_1} + (1 - \alpha) P_{parent_2}$, где $\alpha = [\alpha_1, ..., \alpha_{N_{parent}}]$ из $\mathbb{U}[0, 1]$;
$P \leftarrow [P|P_{rand}|P_{mut}|P_{cross}]$;
$P \leftarrow \{p|\forall p \in P : g(p) = 0\}$;
$F \leftarrow [f(p_1), f(p_2), ...]$;
Пусть F_{sorted}, P_{sorted} отсортированые массивы по возрастанию F;
$f_{iterbest} \leftarrow (F_{sorted})_1$, $p_{iterbest} \leftarrow (P_{sorted})_1$;
if $f_{iterbest} < f_{best}$ **then**
$p_{best} \leftarrow p_{iterbest}$;
$f_{best} \leftarrow f_{iterbest}$;
end if
if $\mathrm{length}(P_{sorted}) < N$ **then**
$P_{sorted} \leftarrow [P_{sorted}|p_{\mathrm{length}(P_{sorted})+1}, ..., p_N]$ (добавление случайных параметров);
end if
$P \leftarrow [(P_{sorted})_1, (P_{sorted})_2, ..., (P_{sorted})_N]$;
$iter \leftarrow iter + 1$;
end while
Результат: (p_{best}, f_{best}).

4.3.2. Роевая оптимизация

Другим использованным эволюционным алгоритмом является роевая оптимизация. В данной работе использовалась простая версия такого алгоритма, в которой все итерации, следующие за исходной генерацией параметров являются детерминированными. Используя те же обозначения, что и для предыдущего алгоритма, но не используя $g(p)$, запишем:

$P \leftarrow [p_1, p_2, ..., p_N]$ из многомерного равномерного распределения с заданными границами;
$V \leftarrow [\vec{0}, \vec{0}, ..., \vec{0}]$;
$P_{itembest} \leftarrow [null, ..., null]$;
$F_{itembest} \leftarrow [+\infty, ..., +\infty]$;
$p_{best} \leftarrow null$;
$f_{best} \leftarrow +\infty$;
$iter \leftarrow 0$;
while $iter < I$ **do**
    $F = [f(p_1), f(p_2), ..., f(p_N)]$;
    Для $i = \overline{1, N}$, если $F_i < (F_{itembest})_i$: $(F_{itembest})_i \leftarrow F_i$;
    Для $i = \overline{1, N}$, если $F_i < (F_{itembest})_i$: $(P_{itembest})_i \leftarrow P_i$;
    $f_{iterbest} \leftarrow \min F_i$;
    $p_{iterbest} \leftarrow \min P_i$;
    if $f_{iterbest} < f_{best}$ **then**
        $p_{best} \leftarrow p_{iterbest}$;
        $f_{best} \leftarrow f_{iterbest}$;
    end if
    $V \leftarrow \beta(\alpha_{loc}(P_{itembest} - P) + \alpha_{glob}(p_{best} - P)) + (1 - \beta)V$;
    $P \leftarrow P + V$;
    $iter \leftarrow iter + 1$;
end while
Результат: (p_{best}, f_{best}).

ГЛАВА 5
НЕЙРОННЫЕ СЕТИ В ЗАДАЧЕ ПРЕДСКАЗАНИЯ ВЗАИМОДЕЙСТВИЯ БЕЛКОВ

В данной главе описаны нейронные сети, использованные и предлагаемые к использованию для решения задачи предсказания взаимодействия белков, а также подходы к процессу обучения таких сетей. Основные фокусы при проектировании таких сетей:

- Использование матриц косинусов для представления входных/выходных данных;
- Применимость к данным различных размеров.

5.1. Архитектура нейронных сетей

5.1.1. Полносвёрточная нейронная сеть FCN5

Полносвёрточнная нейронная сеть [3] – это свёрточная нейронная сеть, в которой отсутствуют полносвязные слои. Как и все свёрточные сети, полносвёрточные сети позволяют эффективно обрабатывать изображения различных размерностей, наиболее часто - двумерных. Главной особенностью свёрточнных сетей без полносвязных слоёв является то, что они могут преобразовывать входные изображения практически любых размеров без масштабирования. Это крайне важное свойство при работе с матрицами косинусов, так как в отличие от графических изображений, их нельзя масштабировать. В рамках данной работы использовалась сеть FCN5, которая была разработана и использована еще в [2]. Несколько сетей FCN5 можно последовательно объединнить в одну большую, если это необходимо.

Сеть FCN5 состоит из 5 остаточных слоёв [22], 2 слоёв батч-нормализации [23], между слоями присутствуют *пропускные связи*.

Используемые остаточные слои задаются количеством входных и выходных каналов, состоят из двух свёрточных слоёв, и организованы следующим образом:

1. Входной тензор поступает на первый свёрточный слой, который производит промежуточный тензор. Количество каналов в промежуточном тензоре равно количеству выходных канналов.

2. К промежуточный тензору применется функция активации ReLU, а результат поступает на выход второго свёрточного слоя. Второй свёрточный слой сохраняет количесвто каналов тензора. Выход второго свёрточного слоя суммируется с промежуточным тензором (до применения ReLU), а к результату применяется функция активации LeakyReLU с

параметром отрицательного склона равного 0,2 (функия активации не применняется для последнего остаточного блока в сети). Полученный тензор является выходом остаточного слоя.

3. Размер ядра используемых свёрток – 3×3.

4. Для сохранения размеров (ширины и высоты) входных тензоров используется круговой паддинг. Использование других видов паддинга (нулевого, зеркального) не рекомендуется.

Смысл пропускных связей (англ. *skip connection*) заключается в том, чтобы отправлять на вход слоя нейронной сети не просто выход предыдущего слоя, но выход предыдущего слоя соединённый (коннкатенация каналов) с выходами еще более ранних слоёв. Общее устройство сети FCN5 можно увидеть на следующей схеме:

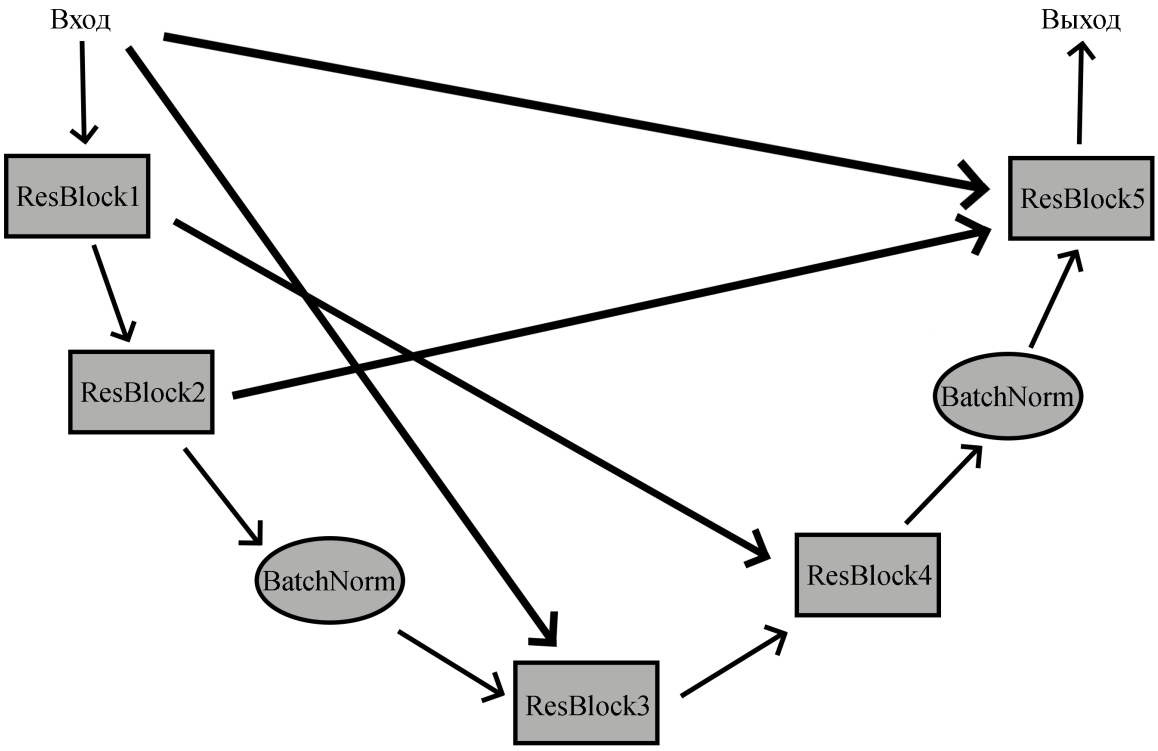

Рис. 5.1. Устройство сети FCN5.

Сеть FCN5 по устройству крайне близка к сетям типа U-Net [24]. Но сеть U-Net известна тем, что количество каналов промежуточных тензоров сначала растёт, а потом убывает. При работе с FCN5, как и в большинстве других свёрточных сетей, количество каналов растёт от начала до конца.

Отметим, что количество каналов входного, а также количество выходных каналов в 5 остаточных слоях могут быть произвольными. В экспериментах часто использовалась следующая комбинация для промежуточных слоёв (входные каналы и пятый слой зависят от типа данных):

- Остаточный слой 1: 128 каналов;
- Остаточный слой 2: 256 каналов;
- Остаточный слой 3: 512 каналов;
- Остаточный слой 4: 256, 384 или 512 каналов.

Для ускорения вычислений, также можно использовать уменьшенный вариант:

- Остаточный слой 1: 32 канала;
- Остаточный слой 2: 64 канала;
- Остаточный слой 3: 128 каналов;
- Остаточный слой 4: 43 канала.

5.1.2. Концепция сети-трансформера

Основным недостатком полносвёрточных нейронных сетей является то, хоть они и могут обрабатывать изображения сколь угодно больших размеров (на сколько то позволяет память компьютера), для изображений крайне малого размера (которые в рамках данного исследования могут возникнуть при работе с пептидами), может оказаться, что применяемых свёрток слишком много (каждая свёртка уменьшает размер изображения, после чего операция паддинга возвращает исходный размер), и результат окажется совершенно ненадёжным. Также отметим, что операция свёртки локальна – за формирования одного пикселя выходнного изображения отвечает ограниченная область вокруг этого пикселя в исходнном изображении. Это значит, что практически отстутствует связь между удалёнными участками изображения.

Наиболее привлекательным решением таких проблем выглядит использование сети-трансформера [25]. Такие сети в настоящее время получили крайне широкое распространение, прежде всего в задачах обработки естественного языка, но также и в задачах биоинформатики. Сети-трансформеры хорошо работают с данными, представленными в виде последовательностей, позволяют эффективно находить связи между удалёнными участками таких последовательностей.

Концептуально, в такой сети можно использовать токен, бинарнно кодирующий 5 аминокислотных остатков и релятивистскую матрицу косинусов размера 4×4. На выходе можно, например, получать токены с матрицами косинусов 4×4, кодирующими матричное поле. Можно использовать и более простые выходные токены.

5.2. Представление данных и процесс обучения нейронной сети

5.2.1. Набор данных

Для создания набора данных использовались записи из банка данных PDB, загруженнные ранее 31 октября 2022. Была составлена программа, которая выявляла среди этих файлов модели белковых комплексов. Было обнаружено:

- 24370 димерных комплексов;
- 4212 тримерных комплексов;
- 7987 4-мерных комплексов;
- 732 5-мерных комплексов;
- 2335 6-мерных комплексов;
- 164 7-мерных комплексов;
- 1270 8-мерных комплексов;
- 1853 комплексов более высокого порядка.

После фильтрации, которая включала в себя выделение всех попарных взаимодействий, исключая содержащие нестандартные аминоксилотные остатки, а также слишком короткие взаимодействующие молекулы, получили наборы данных из:

- 85014 взаимодействий в обучающем наборе;
- 21547 взаимодействий в тестовом наборе.

5.2.2. Процесс обучения

Нейронная сеть обучается в течении нескольких больших итераций, называемых эпохами. Одна эпоха означает один полный обход тестового набора данных, и валидацию на всём тестовом наборе. Каждую эпоху делят на некоторое число итераций, при которых выделяется фиксированный «кусок» тестового набора данных (батч), который обрабатывается нейронной сетью, результат оценивается с помощью некоторой функции потерь, после чего веса нейрронной сети обновляются для минимизации этой самой функции потерь. Валидация тестового набора данных также производится по батчам.

В предлагаемом подходе, типичная итерация обучения выглядит следующим образом:

1. Из набора данных выбирается следующий батч. Для этого в каждой записи о взаимодействии двух молекул выбирают пару взаимодействующих остатков, и производят обрезку обеих молекул под фиксированный размер так, чтобы взаимодействующая пара остатков сохранилась. Теперь, для сравнения, либо выбирают точку, где взаимодействия нет (для (3.24)), либо строят другую, ложную модель взаимодействия. Координаты, закодированные аминокислотные остатки, а также координаты точек взаимодействия собирают в тензоры и отправляют на память видеокарты.

2. На видеокарте из координат атомов и точек взаимодействия строят соответствующую релятивистскую матрицу косинусов. Код аминокислотных остатков дополнительно обрабатывают, и записывают в виде матрицы.

3. Нейросеть высчитывает матрицы косинусов, кодирующие обобщённые переменные поля для истинного и ложного взаимодействия. Из этих матриц извлекают эти переменные и высчитывают частичные лагранжианы L^T_{part} и L^F_{part}.

4. В качестве функции потерь проще всего использовать разницу $L^T_{part} - L^F_{part}$, что будет приводить к тому, что истинным моделям будет сопоставляться меньший частичный лагранжиан, чем ложным. Также желательно, чтобы выполнялось: $L^T_{part} < 0$ и $L^F_{part} > 0$, для чего использовалась функция потерь:

$$l(L^T_{part}, L^F_{part}) = \frac{(L^T_{part} - L^F_{part}) - \mathrm{ReLU}(-L^T_{part}) + \mathrm{ReLU}(-L^F_{part})}{2}. \tag{5.1}$$

   К данной функции также можно добавить значение (с маленьким весом), которое отражающую внутреннюю согласованность предсказанной матрицы косинусов (для этого из матрицы сначала получают обобщённые переменные, а потом по этим переменным опять строят матрицу, после чего считаеют разницу между первым и вторым).

5. Посчитанная $l(L^T_{part}, L^F_{part})$ минимизируется методом оптимизации Adam.

В следующей главе изложены результаты использования описанного выше подхода.

ГЛАВА 6
ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

6.1. Синтетические задачи

6.1.1. Задача №1

6.1.2. Задача №2

6.1.3. Задача №3

6.1.4. Задача №4

6.2. Моделирование полей для задачи взаимодействия белков

ЗАКЛЮЧЕНИЕ

В рамках данного отчёта, приведены черновые варианты первых двух глав диссертационной работы. В них содержатся:

- Укороченное изложение ранее полученных результатов, касающихся матриц косинусов.
- Новые результаты, касающиеся релятивистских матриц косинусов.
- Формальное определение предлагаемой модели взаимодействия белков.
- Описание некоторых моделей полей, изучавшихся в экспериментах.

В рамках научно-исследовательской практики была выполнена следующая работа:

- Проанализированы ранее полученные результаты.
- Составлен новый банк белков.
- Разработаны некоторые программные процедуры для моделирования взаимодействия белков.
- Проведены численные эксперименты, в том числе, с обучением глубоких нейронных сетей.
- Проанализированы полученные результаты. Подготовлен данный отчёт.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Deep Learning for Protein–Protein Interaction Site Prediction / R. Arian [et al.] // Methods Mol Biol. – 2021. – P. 1-26.
2. Novikov, A. A. Prediction of protein-protein interaction with cosine matrices / A. A. Novikov, A. V. Tuzikov, A. V. Batyanovskii // Pattern Recognition and Information Processing (PRIP'2023) : Proceedings of the 16th International Conference, October 17–19, 2023, Minsk, Belarus / United Institute of Informatics Problems of the National Academy of Sciences of Belarus. – Minsk, 2023. – P. 258–263.
3. Shelhamer, E. Fully convolutional networks for semantic segmentation / E. Shelhamer, J. Long // IEEE – 2015. – P. 1-12.
4. Exploring protein symmetry at the RCSB Protein Data Bank / J. Duarte [et al] // Emerging Topics in Life Sciences. – 2022. – P. 1-13.
5. Vreven, T. Integrating atom-based and residue-based scoring functions for protein–protein docking / T.Vreven, H. Hwang, Z. Weng // Protein Sci. – 2011. – P. 1-11.
6. Highly accurate protein structure prediction with AlphaFold / J. Jumper [et al] // Nature. – 2021. – P. 1-16.
7. Protein complex prediction with AlphaFold-Multimer / R. Evans [et al] // DeepMind. – 2021. – P. 1-25.
8. Accurate structure prediction of biomolecular interactions with AlphaFold 3 / J. Abramson [et al] // Nature. – 2024. –P. 493-517.
9. FastFold: Reducing AlphaFold Training Time from 11 Days to 67 Hours / S. Cheng [et al] // arXiv. – 2023. – P. 1-13.
10. Andras, S. Template-based structure modeling of protein-protein interactions / S. Andras, Y. Zhang // Curr Opin Struct Biol. – 2014. – P. 1-24.
11. Hang, Y. Protein surface representation and analysis by dimension reduction / Y. Hang, R. Quershi, A. Sacan // Proteome Science. – 2012. – P. 1-13.
12. Fast end-to-end learning on protein surfaces / F. Sverrisson [et al] // bioRxiv. – 2020. – P. 1-11.
13. Prediction of inter-chain distance maps of protein complexes with 2D attentionbased deep neural networks / Z. Guo [et al] // University of Missouri. – 2022. – P. 1-13.
14. Хадарович, А. Ю. Алгоритмы моделирования димерных белковых комплексов / А. Ю. Хадарович // ОИПИ НАН. – 2020. – P. 1-100.
15. Kufareva, I. Methods of protein structure comparison / I. Kufareva, R. Abagyan // Methods Mol Biol. – 2012. – P. 1-30.
16. Bradley, A. The use of the area under the ROC curve in the evaluation of machine learning algorithms / A. P. Bradley // Pattern Recognition. – 1997. – P. 1145-1159.
17. Basu, S. DockQ: A Quality Measure for Protein-Protein Docking Models / S. Basu, B. Wallner // PLoS ONE. – 2016. – P. 1-9.

18. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes / W. Zhu [et al] // Bioinformatics. – 2023. – P. 1-7.
19. Богуш, А.А. Введение в теорию классических полей / А. А. Богуш, Л. Г. Мороз. – Изд. 2-е. – М.: Едиториал УРСС, 2004. – 384 с.
20. Kingma, D. Adam: A Method for Stochastic Optimization / D. Kingma, J. Ba // arXiv. – 2017. – P. 1-15.
21. Eiben, A. Introduction to Evolutionary Computing / A.E. Eiben, J.E. Smith. – 2nd ed. – Berlin : Springer, 2015. – XII, 287 p.
22. Deep Residual Learning for Image Recognition / H. Kaiming [et al] // arXiv. – 2015. – P. 1-12.
23. Ioffe, S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift / S. Ioffe, C. Szegedy // arXiv. – 2015. – P. 1-11.
24. Ronnenberger, O. U-Net: Convolutional Networks for Biomedical Image Segmentation / O. Ronnenberger, P. Fisher, T. Brox // arXiv. – 2015. – P. 1-8.
25. Attention Is All You Need / A. Vaswani [et al] // arXiv. – 2015. P. 1-15.