Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук имени профессора Н.И. Червякова
Кафедра математического моделирования

Утверждена распоряжением по факультету от 18.03.2025 № 127-р-25.00

Допущена к защите
«____» _____ 20____ г.

Зав. кафедрой математического моделирования, к.ф.-м.н., доцент Ляхов П.А.

_____

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ХИМИИ

Рецензент:
Самойленко Ирина Владимировна,
кандидат технических наук, доцент, доцент кафедры информационных систем СтГАУ

Выполнил:
Бобров Анатолий Анатольевич
студент 2 курса,
группы ПМИ-м-о-23-1
направления подготовки 01.04.02 Прикладная математика и информатика
очной формы обучения

_____

Нормоконтролер:
Ляхов Павел Алексеевич
кандидат физико-математических наук, заведующий кафедрой математического моделирования факультета математики и компьютерных наук имени профессора Н.И. Червякова

_____

Научный руководитель:
Ляхов Павел Алексеевич
кандидат физико-математических наук, заведующий кафедрой математического моделирования факультета математики и компьютерных наук имени профессора Н.И. Червякова

_____

Дата защиты
«____» _____ 2025 г.

Оценка _____

Ставрополь, 2025

Теперь решим более сложную задачу. Допустим, имеются ^{1}H и ^{13}C ЯМР спектры вещества, также известны условия их записи (частота в МГц, растворитель). Данные спектры представлены в виде вектора (на интервалах 0-15 ppm и 0-250 ppm для ^{1}H и ^{13}C ЯМР соответственно, частота дискретизации 500 точек / 1 ppm). Необходимо определить структуру вещества.

Для начала необходимо найти подходящий набор данных для обучения нейронной сети. NP-MRD наиболее подходящий вариант, поскольку в этой базе в небольшое количество файлов собраны все необходимые спектры и структурные формулы к ним.

Далее нам необходимо описать алгоритм для преобразования данных в вид, удобный для обучения на них нашей модели. Имеются два файла – один из них содержит информацию о строении всех веществ, другой – файл архива, в котором спектры записаны в отдельные файлы для каждого соединения.

Файл со структурами веществ – обычный csv-файл, средства его чтения мы описывали выше. Сложность представляют спектры веществ – сигналы ЯМР необходимо преобразовать в одномерный вектор.

Форма сигналов похожи на распределение Коши:

$$f_X(x) = \frac{A}{\pi}\left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right], \tag{3.1}$$

где A – площадь пика (количество атомов), γ – параметр масштаба, для нашей задачи $\gamma = 0.005$, x_0 – центр пика, x – координата, в которой необходимо найти величину сигнала.

Чтобы построить сигнал, необходимо знать обозначения, которые вкладываются в символьные обозначения в базе данных. Как и заведено, это общепринятые обозначения:

- s – сиглет (один сигнал);

- d – дублет (два сигнала);

- t – триплет (утроенный сигнал);

- q – квартет (четыре сигнала);

- p – квинтет (пять сигналов);

- m – мультиплет (шесть и более сигналов, обычно беспорядочных).

Сколько бы ни было сигналов, для одного протона площадь под графиком такого сигнала будет равна единице. При этом высота каждого пика будет зависеть от количества сигналов протона в пропорциях, которые можно найти, воспользовавшись треугольником Паскаля, либо формулой:

$$C_n^1 : C_n^2 : \ldots : C_n^n, \tag{3.2}$$

$$C_n^m = \frac{n!}{m!\,(n-m)!},$$

где n – число, равное количеству сигналов, m = 1, 2 … n. Так, для триплета распределение высот сигнала будет 1:2:1.

Особого внимание требует мультиплет. Под этим понятием подразумевается самые разные сигналы, для определенности будем считать, что это достаточно сильно уширенный синглет (γ = 0.1).

Расстояние между сигналами определяется константами расщепления. Константы расщепления указываются в Гц, поэтому для наших целей эти величины необходимо разделить на рабочую частоту спектрометра (500 МГц):

$$J_{ppm} = \frac{J_{\text{МГц}}}{500}\,. \tag{3.3}$$

Сложные сигналы, когда расщепление пика протона происходит из-за соседних протонов, обозначаются несколькими буквами. Например, триплет-дуплет с константами 20 и 6 Гц обозначается как “td” и выглядит как на рисунке 3.1.

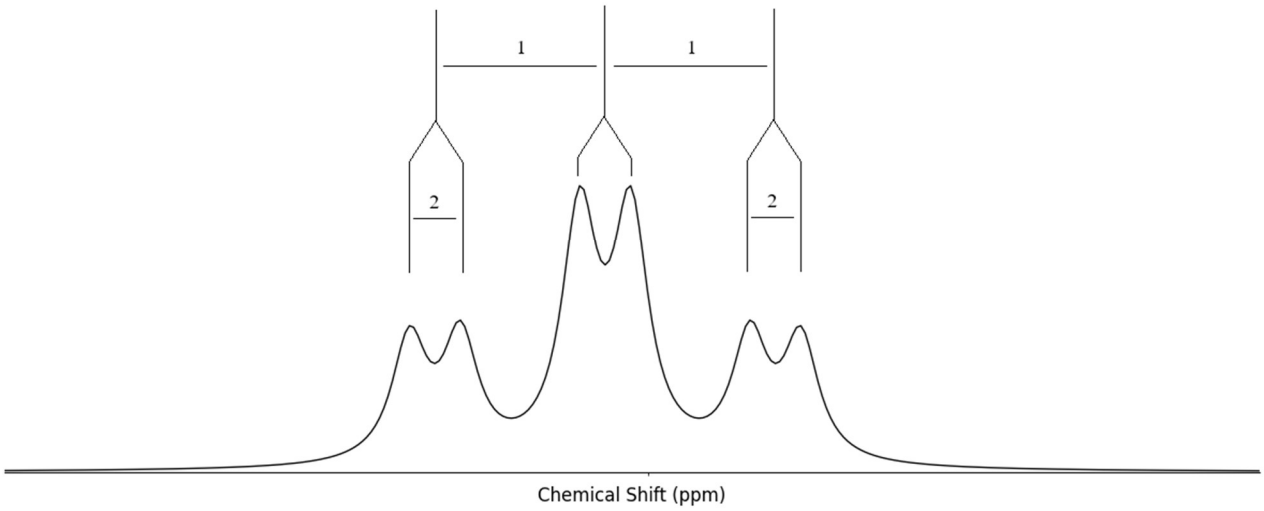

Рисунок 3.1 – Расщепление сигнала, расстояние 1 – 20 Гц, 2 – 6 Гц

При этом мы можем увидеть, что средний сигнал выше двух других в два раза, при этом он раздвоен на два одинаковых по высоте пика.

Итак, теперь, когда алгоритм приведения исходных данных в необходимый вид готов, можно приступать к построению спектров. Пример спектра вещества приведен ниже на рисунке 3.2.

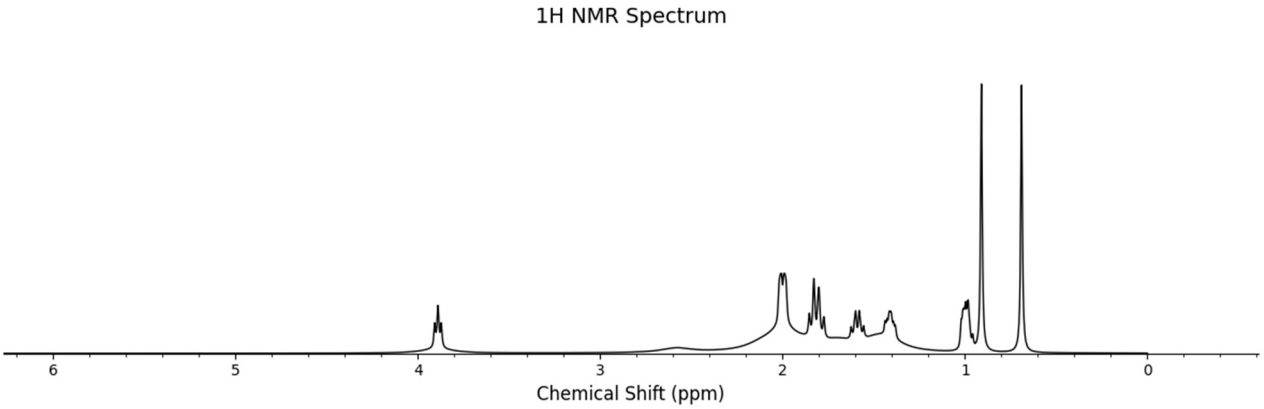

Рисунок 3.2 – Протонный спектр ЯМР вещества NP0104369

Выше приведены правила для протонного спектра ЯМР. Большинство спектров умещаются в интервале 0 – 15 ppm, хотя для гидридов металлов значения химических сдвигов может быть отрицательным. Таких веществ нет в наборе данных, поэтому эти значения не войдут в спектр.

Необходимо сказать, что не все органические вещества имеют в своем составе водород, поэтому их идентификация по протонному спектру невозможна. Более того, тестирование обученной только на протонных ЯМР выявило ошибку в 0.00447 против 0.00397 на обоих спектрах. На схеме 3.1 приведены примеры таких веществ.

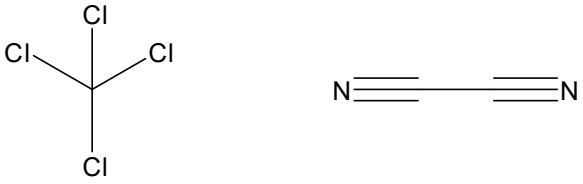

Схема 3.1. Тетрахлорметан и циан

Для ^{13}C ЯМР правила те же самые, за исключением следующих моментов. Химические сдвиги углеродного спектра имеют большие сдвиги (0 – 250 ppm). При этом формы сигналов менее разнообразны, и преобладают синглеты. Пример спектра ^{13}C ЯМР приведен ниже:

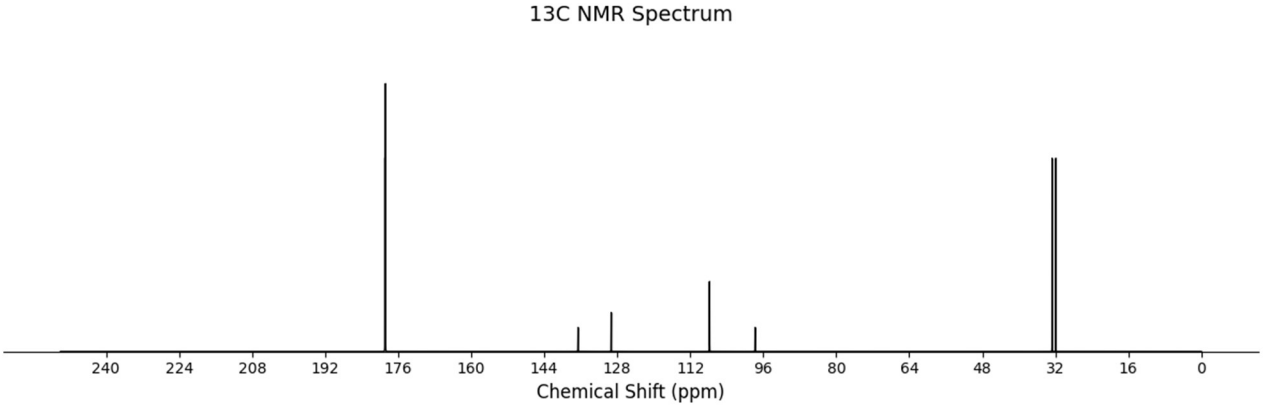

Рисунок 3.3 – Углеродный спектр ЯМР вещества NP0120358

Оба спектра мы представили в виде вектора длиной 7 500, затем нормализовали. Объединив их, получим вектор длиной 15 000.

Теперь, когда удалось представить спектры в виде, удобном для обучения, представим структуры веществ в виде многомерного вектора. Самый простой способ это сделать – ограничить структуру по количеству входящих в нее атомов (без атомов водорода). Для удобства ограничение примем за 80. Каждый атом

описывается рядом свойств, о которых мы говорили ранее, и для кодирования одного атома нам понадобится 136 чисел. Отсюда вывод: количество чисел, необходимых для кодирования всех атомов равно 136 * 80 = 10 880. Количество n всевозможный связей между атомами найдем как сумму первых N_{atoms} членов арифметической прогрессии:

$$n = \frac{1 + N_{atoms} - 1}{2}(N_{atoms} - 1) = \frac{N_{atoms}\,(N_{atoms} - 1)}{2} = \frac{80 * (80 - 1)}{2} = 3160 \quad (3.4)$$

Закодировать эти связи возможно с помощью 3160, при этом значение 0.01 будет обозначать отсутствие этой связи, а значение больше будет указывать на ее наличие и ее кратность.

Однако на наличие и кратность связи возможно выделить 7 самостоятельных числа, каждое из которых будет однозначно указывать на кратность связи (к примеру, (0.01, 0.99, 0.01, 0.01) – двойная связь, (0.01, 0.01, 0.99, 0.01) – тройная связь) и направление связи ((0.01, 0.99, 0.01) – транс-изомер, (0.99, 0.01, 0.01) – цис-изомер). Тесты показали, что эти 7 числа лучше передают информацию о связи (функция потери уменьшилась с 0.00319 до 0.00261).

Тогда количество признаков связей будет 3160 * 7 = 22 120. Всего количество чисел, кодирующих молекулу равно 22 120 + 10 880 = 33 000.

В наборе данных сведения о структуре хранятся в виде SMILES. Модуль RDKit позволяет преобразовать такую запись структуры в виде таблиц свойств атомов, таблицы связей и таблицы свойств этих связей. Чтобы закодировать сведения о связи (a, b) (нумерация атомов начинается с 0) для связи необходимо производить запись в вектор, начиная с:

$$t = 4 * \left(\frac{2(N_{atoms} - 1) + (a - 1)(-1)}{2} * a + b - a - 1\right) = 4 * \left(\frac{2N_{atoms} - 1 - a}{2} * a + b - a - 1\right) \quad (3.5)$$

Следующим этапом является построение модели. Изначально данные подготовлены так, что их удобно использовать для обучения перцептрона. Так и поступим. Архитектура сети будет следующей:

- Полносвязный слой Linear(15 000, hc, dtype=DTYPE);
- Слой активации ReLU;
- Полносвязный слой Linear(hc, 33 000, dtype=DTYPE);
- Слой активации Sigmoid.

Сигмоидная функция активации работает медленнее ReLU, однако она улучшает точность предсказания. Это связано с тем, что вектор ответов ограничен интервалом [0.01, 0.99]. Тесты с заменой первого слоя активации на сигмоидный не показали улучшения качества сети, к тому же это замедлило работу нейронной сети.

Обсудим, что такое DTYPE? Под DTYPE мы обозначили тип данных для расчетов. Мы можем использовать таким образом управлять точностью, объемом памяти и временем обучения. Мы провели обучение нейронной сети и получили следующие результаты:

- при использовании torch.float32 нейронная сеть ускорилась в 15-20 раз, занимаемый объем памяти уменьшился в 2 раза по сравнению с torch.float64, при этом потери в точности не наблюдалось;

Другое обозначение hc подразумевает количество нейронов в скрытом слое. На небольшом наборе данных были проверены различное количество нейронов:

- 4 000 – значение функции потерь 0.003663, время обучения 11 с / 1 молекулу, вес сети 750 Мб;
- 3 000 – значение функции потерь 0.003656, время обучения 8 с / 1 молекулу, вес сети 563 Мб;
- 2 000 – значение функции потерь 0.003560, время обучения 5 с / 1 молекулу, вес сети 375 Мб;
- 1 000 – значение функции потерь 0.003965, время обучения 3 с / 1 молекулу, вес сети 188 Мб;

Как видим, оптимальным вариантом является 2 000 нейронов. Далее предпринята попытка добавления еще одного скрытого слоя нейронов. Однако такой подход не увенчался успехом – значение функции потерь возросло до 0.005902 (время обучения составило 10 с / 1 молекулу).

Мы до сих пор принимали как должное отсутствие атомов водорода в графовом представлении молекул. Хотя из такого представления возможно однозначное восстановление всех атомов водорода, нейронная сеть, судя по точности предсказания, не умеет это делать наверняка. Поэтому было принято решение использовать полную информацию о структуре вещества с помощью метода AddHs модуля RDKit.

Далее стоит подвергнуть сомнению ограничение в 80 атомов. Полный набор данных содержит 13 779 веществ. При использовании ограничений уменьшаются наборы данных и время обучения:

- 13 598 веществ до 150 атомов (суммарное время обучения 1200 с / 1 молекулу);

- 12 404 веществ до 100 атомов (суммарное время обучения 14 с / 1 молекулу);

- 10 696 веществ до 80 атомов (суммарное время обучения 7 с / 1 молекулу).

Как видно, ограничение в 80 атомов охватывают большую часть молекул в наборе и при этом время обучения наименьшее. Поэтому было принято оставить это ограничение без изменений.

Принято дополнительное ограничение – в молекуле должны отсутствовать заряженные атомы. Это необходимо, чтобы исключить соли из набора данных. К сожалению, это ограничение исключает из набора данных цвиттер-ионы. После наложения такого ограничения, осталось 10 499 вещества (исключены 197 веществ). Это увеличило точность предсказания.

Для одной и той же молекулы возможны различные формы записи SMILES, например:

Вещество с кодом NP0136338 имеет следующие записи:

- «C[C@H](N)C(=O)N[C@@H](CC1=CC=C(O)C=C1)C(O)=O»;

- «C[C@H](N)C(=O)N[C@@H](Cc1ccc(O)cc1)C(=O)O».

Вещество с кодом NP0136339 имеет следующие записи:

- «CC(C)[C@H](NC(=O)[C@H](C)N)C(O)=O»;

- «CC(C)[C@H](NC(=O)[C@H](C)N)C(=O)O».

Как можем видеть, имеются отличия не только в способе обозначения связей, но и в порядке атомов. Поэтому порядок атомов и связей в массиве может различаться.

Чтобы избежать неоднозначности, необходимо канонизировать записи молекул с помощью функции rdkit.Chem.CanonSmiles. При этом сама молекула будет оставаться неизменной. Этот факт косвенно доказывает то, что значение функции потерь при обучении после применения данной функции изменилось незначительно.

До сих пор мы использовали функцию потерь для оценки точности предсказания структуры вещества. Теперь опишем алгоритм для прямого расчета точности модели.

Для сравнения двух молекул можно использовать различные алгоритмы. Используем метод сравнения молекулярных отпечатков пальцев, который предоставляет RDKit (rdkit.DataStructs.FingerprintSimilarity). Для этого необходимо получить отпечаток с помощью функции rdkit.Chem.RDKFingerprint.

Однако перед нами оказывается нерешенная проблема – для нашего метода кодировки нужен декодировщик, который восстанавливает молекулярный граф из его векторного представления. Для начала необходимо определить количество атомов в молекуле.

Подсчет атомов происходит параллельно с восстановлением сведений об атоме. При определении символа атома определяется максимальное значение в 118-мерном векторе. Если это значение выше определенного порога, считается, что этим кодом закодировано наличие атома. В противном случае считается, что атома нет и алгоритм по восстановлению информации об атомах завершается.

Так как все значения вектора находятся в интервале (0, 1), для начала был проверено пороговое значение 0.5, однако проверка других значений дала неожиданные результаты (см. таблицу ниже).

Далее нужно описать алгоритм декодировки молекулярных связей. При определении свойств связи параллельно определяется ее наличие аналогичным образом: если в 4-мерном векторе максимальное значение меньше порогового, то это указывает на ее отсутствие. В противном случае эта связь существует. Для связей подбор порогового значения производится аналогично (таблица 3.2)

Таблица 3.2. Подбор оптимальных гиперпараметров нейронной сети.

		Порог для атомов								
		0.01001	0.0101	0.0105	0.0110	0.015	0.020	0.05	0.2	0.3
Порог для связей	0.015	-	-	-	-	-	-	0.4098	-	-
	0.020	-	0.4867	0.4872	0.4757	0.4307	0.4230	0.4123	-	-
	0.025	0.5109	-	0.5019	-	-	-	0.3945	-	-
	0.030	0.5132	0.5120	0.5026	-	-	-	0.3834	-	-
	0.050	0.4964		0.4862	-	-	-	0.3562	-	-
	0.100	-	-	-	-	-	-	0.3198	-	-
	0.200	-	-	-	-	-	-	0.2941	-	-
	0.300	-	-	-	-	-	-	0.2803	-	-
	0.450	-	-	-	-	-	-	0.2425	-	-
	0.500	-	-	-	-	-	-	0.2299	-	-
	0.550	-	-	-	-	-	0.2196	0.2206	0.1876	0.1666
	0.600	-	-	-	-	-	-	0.2113	-	-

Судя по этим результатам, лучшим выбором будет 0.01001 (меньше 0.01 значения недопустимы) для атомов и 0.03 для связей.

Следующим параметром для проверки будет скорость обучения. До сих пор мы использовали в тестах значение 0.001. Для различных скоростей получены следующие результаты (таблица 3.3):

Таблица 3.3. Подбор оптимальной скорости обучения.

Скорость обучения, 10^{-3}	Функция потерь, 10^{-3}
0.5	3.861
1.0	3.560
2.5	3.678
5.0	5.198

Как можем убедиться, выбор скорости обучения (0.001) является удачным, поэтому будем придерживаться и далее этого значения.

Теперь, когда подготовлен фундамент для проведения обучения, необходимо ускорить процесс обучения. При использовании расчетов на CPU скорость обучения составляет 5 с / молекулу при количестве эпох обучения, равном 10. При обучении модели на полном наборе данных (10 696 молекул) это время составит примерно:

$$t = \frac{100}{10} * 5 * 10\,696 \text{ с} \approx 6.2 \text{ д.} \quad (3.6)$$

Это очень большой промежуток времени. Решением является использование технологии графического ускорения параллельных вычислений CUDA, которую поддерживает PyTorch. Обычно объем видеопамяти меньше оперативной, поэтому необходимо оценить объем, который нам понадобится:

- вес нейронной сети 375 Мб (определено эмпирически);

- вес одной записи в наборе данных: 15 000 байт для спектра + 33 000 байт для структуры вещества = 48 000 байт ≈ 47 Кб; в наборе данных 10 696 веществ, весь набор данный займет примерно 47 Кб * 10 696 ≈ 491 Мб.

Суммарно нам понадобится 491 + 375 = 866 Мб. При объеме видеопамяти в 8 Гб, такой вес допустим.

Для стохастического градиентного спуск, ускорить возможно лишь выполнение отдельной итерации. При тестировании такого подхода скорость обучения

увеличилась в ~13 раз, даже не смотря на то обстоятельство, что для анализа результатов тестирования необходимо выгружать данные с графической памяти видеокарты в оперативную.

Для обучения «мини-пакетами» по 32-256 обучение ускоряется в 2-3 раза без потери точности предсказания.

На тестовом наборе данных точность предсказания составила 67.83%. Это хорошая точность для такой простой модели. На момент написания практической части нет попыток решить эту задачу (литература по этой теме отсутствует).

Более сложные алгоритмы предполагают использование воссоздания структуры, последовательно подбирая атомы и связи, и требуют предварительного обучения декодировщиков. Существуют и другие методы. Это требует больших вычислительных мощностей и времени.

3.3. Осуждение результатов экспериментов

Для определения структуры вещества по комбинации ^{1}H и ^{13}C ЯМР-спектров был выбран перцептрон, основной целью которого является классификация атомов и связей между ними. Для обучения использована база NP-MRD. Записи в этой базе уступают по качеству реальным спектрам, однако обучение на реальных спектрах требует ручного сбора данных, что заняло бы большое количество времени и объема памяти.

Спектры моделировались распределением Коши с учётом мультиплетности (синглеты, дублеты и т.д.) и констант расщепления. Серьезным недостатком модели являются жёсткие ограничения на размер молекул и отсутствие зарядов. Предложенный метод демонстрирует принципиальную возможность реконструкции молекулярных структур по ЯМР-спектрам с помощью нейронных сетей, но требует значительной доработки для практического применения. Работа закладывает фундамент для новых подходов в автоматической расшифровке спектров.