**Текст презентации к проектной работе**

*Слайд 1*

Добрый вечер, уважаемые коллеги! Сегодняшняя презентация будет посвящена тому, каким образом по заданным embedding’ам еще не синтезированных молекул можно предсказывать их свойства и каким образом эти embedding’и можно генерировать.

*Слайд 2*

Первым делом стоит ответить на вопрос: «Что такое embedding молекулы?» Мы рассматриваем молекулу в виде трехмерного графа, в котором узлы являются атомами, а ребра – связи между молекулами, хранящими данные о пространственном расстоянии между атомами. В рамках задачи предсказания свойств граф будет описываться тремя объектами: матрицей смежности, матрицей состояний каждого узла и словарем, содержащим информацию о попарных расстояниях между смежными вершинами. Вторая из указанных матриц строится на основе строкового описания молекулярной структуры SMILES, которое более подробно будет рассматриваться в дальнейшем.

*Слайд 3*

Один прецедент представляет из себя набор следующих данных. Для выделения дополнительных параметров из SMILES’а используется библиотека RDKit, …

*Слайд 4*

… которая парсит строчку и выдает вектор свойств молекулы. На слайде снизу представлен пример SMILES для . Для построения трехмерного графа молекулы используется инструментарий библиотеки NetworkX.

*Слайд 5*

Итак, полный алгоритм обработки исходных данных состоит из 6-и шагов, представленных на слайде.

*Слайд 6*

#код (рассказать про данные)

*Слайд 7*

#код (рассказать про данные)

*Слайд 8*

#код (рассказать про данные)

*Слайд 9*

Перейдем к рассмотрению supervised архитектуры нейронной сети, реализованной в рамках проекта. MPNN была придумана компанией Google как модификация уже имеющих нейросетевых решений, для которых входными данными являются графы и представлена на ICML 2017. Исходная задача ставилась на основе датасета QM9, состоящего из 130.000 молекул с 13-ю свойствами, полученными в результате эталонных лабораторных вычислений и применения вычислительного метода Density Functional Theory и взятыми в качестве таргетов. Представленная архитектура справлялась с аппроксимацией 11-и из 13-и таргетов на основе DFT error. В нашей работе был взят другой датасет, в котором за таргет брались энтальпия и плотность. Это два важных свойства вещества, для вычисления которых необходимы длительные сроки, помимо прочего, плотность не считается в чистом виде, так как квантово-химические расчеты основаны на том, что все вещества находятся в газовом состоянии. Построенная модель на основе известных данных способна с хорошей точностью предсказывать эти свойства.

Итак, на вход сети подается ненаправленный трехмерный граф, описанный ранее, нейронная сеть выдает для каждого прецедента упомянутые свойства. Forward pass разбивается на две части: message passing phase и readout phase. В начале обучения задается гиперпараметр *T*, отвечающий за количество повторений первого этапа перед началом второго, а также выбираются три дифференцируемые функции .

*Слайд 10*

Message passing phase определены в смысле функции сообщений и функции обновления состояния вершин . В ходе этапа новые сообщения вычисляются согласно правилу, представленному на слайде. Здесь – расстояние между вершинами (для сверток молекулярного графа представляет из себя состояние связи, для которого обучаются две матрицы весов), – скрытые состояния каждой вершины. То есть берется сумма значений функции сообщений по всем смежным вершинам. На слайде представлены возможные функции сообщений, относящиеся к разным статьям. # точка с кругом – поэлементное умножение

*Слайд 11*

Скрытые состояние обновляются на основании сообщений по следующему правилу. Опять же, на слайде представлены возможные варианты функции обновления состояния вершин.

*Слайд 12*

Если рассматривать процесс относительно зеленой вершины, на шаге *t* вычисляется функция сообщений для красной и желтой вершины, берется сумма и обновляется состояние вершины, переходя тем самым к шагу *t + 1*. Обновления происходят параллельно до шага *T*.

*Слайд 13*

На readout phase рассматриваются только конечные скрытые состояния вершин. Основная особенность readout функции состоит в том, что она не зависит от порядка аргументов, которые ей подаются. Следовательно, наша модель инварианта относительно порядка вершин в графе, то нумерации и положения молекулы в пространстве.

*Слайд 14*

Итак, наша модель была реализована на основе GG NN, т.е. функции сообщений и readout представляют из себя полносвязные нейронные сети, функция обновления состояний – GRU (модификация LSTM, в котором отсутствует конвейер состояний). Мы взяли большое количество эпох, чтобы ошибка была поменьше. Результаты представлены на слайде.

*Слайд 15*

После решения первой задачи мы задумались над тем, чтобы научиться генерировать новые молекулы по заданным свойствам. Предвещая будущее повествование, скажем сразу, что удалось научиться генерировать только 2d представления молекул. Мотивация этой части работы состоит в том, что для продолжения теоретических вычислений необходимы данные о еще не синтезированных веществах, свойства которых получены в результате лабораторных экспериментов, в силу закона экономии энергии хочется, чтобы за человека вычисления выполнял кто-то другой, например, модель, реализованная в нашем проекте.

Архитектура модели представляет из себя объединение обучаемых раздельно генератора и предиктора. В качестве генератора выбрана архитектура Stacked RNN с ячейкой GRU (представляет из себя несколько скрытых слоев GRU, где каждый слой содержит несколько ячеек памяти, организованных в виде стека и обновляемых на основании вероятностной модели), которая предсказывает следующий символ из алфавита SMILES на основании уже сгенерированного префикса при наличии заданного свойства.

*Слайд 16*

Модель была обучена минимизировать cross-entropy между предсказанным символом и истинным символом.

*Слайд 17*

Forward pass завершается генерацией символа конца строки.

*Слайд 18*

В качестве предиктора была взята архитектура, представленная на слайде. Суть его заключается в определении валидности строки SMILES, которая на embedding layer переводится в скрытое представление, подаваемое на вход RF. Результат его работы подается на вход полносвязной нейронной сети.

*Слайд 19*

После обучения двух моделей происходит их объединение, в некотором роде предиктор играет роль дискриминатора в GAN. Валидация генерируемых сетью смайлов и их преобразование в 2D схему молекулы производится при помощи библиотеки RDKit

Добавляя штрафы, зависящие от свойства, вычисляемого при помощи этой сети, можно генерировать молекулы с нужными значениями этого свойства

*Слайд 20*

Полученные сгенерированные молекулы представлены на слайде.