Введение в нейронные сети

Урок 8. GAN

Задание.

Сделайте краткий обзор научной работы, посвящённой алгоритму нейронных сетей, не рассматриваемому ранее на курсе.
Проведите анализ:
Чем отличается выбранная архитектура от других?
В чём плюсы и минусы данной архитектуры?
Какие могут возникнуть трудности при её применении на практике?

Решение:

Нейронные сети типа трансформер (Transformer). Статья "Attention is All You Need" (Vaswani et al., 2017)

Один из важнейших инструментов машинного обучения — трансформеры. Популярность трансформеров взлетела до небес в связи с появлением больших языковых моделей вроде ChatGPT, GPT-4 и LLama. Эти модели созданы на основе трансформерной архитектуры и демонстрируют отличную производительность в понимании и синтезе естественных языков.

Трансформер – это такой вид нейросетевой архитектуры, который хорошо подходит для обработки последовательностей данных. Пожалуй, самый популярный пример таких данных это предложение, которое можно считать упорядоченным набором слов.

Трансформеры создают цифровое представление каждого элемента последовательности, инкапсулируют важную информацию о нём и окружающем его контексте. Получившиеся представления затем можно передать в другие нейросети, которые воспользуются этой информацией для решения разных задач, в том числе для синтеза и классификации. Создавая такие информативные представления, трансформеры помогают последующим нейросетям лучше понять скрытые паттерны и взаимосвязи во входных данных. И поэтому они лучше синтезируют последовательные и взаимосвязанные результаты.

Главное преимущество трансформеров заключается в их способности обрабатывать длительные зависимости в последовательностях. Кроме того, они очень производительны, могут обрабатывать последовательности параллельно. Это особенно полезно в задачах вроде машинного перевода, анализа настроений и синтеза текста.

Вообще почти все нашумевшие достижения в глубинном обучении последних лет так или иначе опираются на эту архитектуру. Что же в ней такого особенного и почему трансформеры успешно применяются в самых разных задачах?

Для начала вспомним, что основным подходом для работы с последовательностями до 2017 года (выхода оригинальной статьи про архитектуру Трансформер) было использование рекуррентных нейронных сетей, или RNN. Однако у такого подхода есть несколько известных минусов:
− во-первых, RNN содержат всю информацию о последовательности в скрытом состоянии, которое обновляется с каждым шагом. Если модели необходимо «вспомнить» что-то, что было сотни шагов назад, то эту информацию необходимо хранить внутри скрытого состояния и не заменять чем-то новым. Следовательно, придется иметь либо очень большое скрытое состояние, либо мириться с потерей информации.

− во-вторых, обучение рекуррентных сетей сложно распараллелить: чтобы получить скрытое состояние RNN-слоя для шага i+1, вам необходимо вычислить состояние для шага i. Таким образом, обработка батча примеров длиной 1000 должна потребовать 1000 последовательных операций, что занимает много времени и не очень эффективно работает на GPU, созданных для параллельных вычислений.

Обе этих проблемы затрудняют применение RNN к по-настоящему длинным последовательностям: даже если вы дождетесь конца обучения, ваша модель по своей конструкции будет так или иначе терять информацию о том, что было в начале текста. Хочется иметь способ «читать» последовательность так, чтобы в каждый момент времени можно было обратиться к произвольному моменту из прошлого за константное время и без потерь информации. Таким способом и является лежащий в основе трансформеров механизм внимания self-attention. Благодаря своей универсальности и масштабируемости этот механизм оказался применим к множеству задач помимо обработки естественного языка.

Ниже приведено устройство архитектуры Трансформер из оригинальной статьи [http://arxiv.org/pdf/1706.03762]:

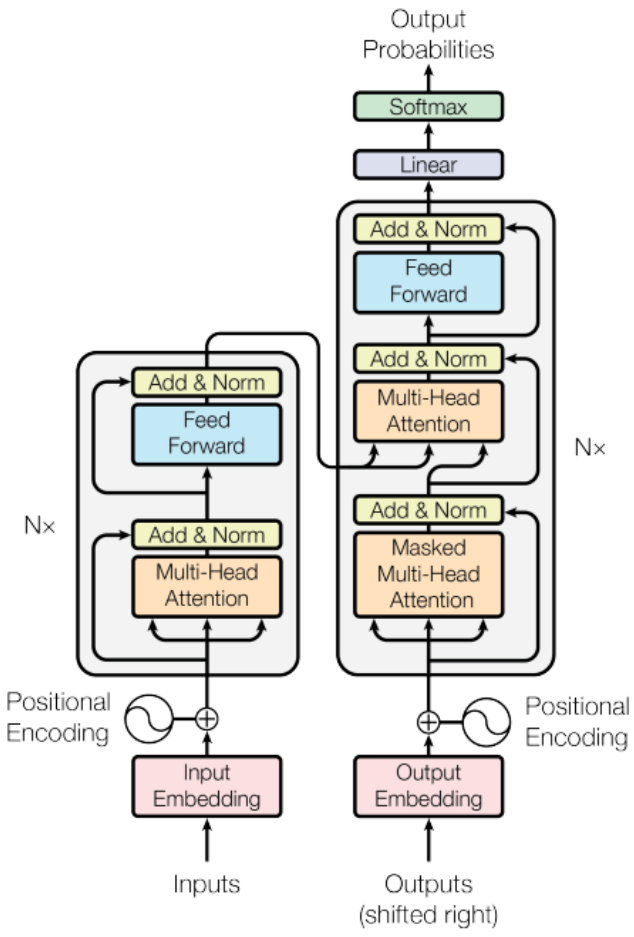

Figure 1: The Transformer - model architecture.

В целом, основное нововведение — это использование механизма внимания self-attention, чтобы взаимодействовать с другими словами в предложении вместо механизмов RNN или CNN.
В статье теоретизируют, что это помогает, потому что сеть может с одинаковой легкостью обратиться к любой информации вне зависимости от длины контекста - обратиться к прошлому слову или к слову за 10 шагов назад одинаково просто.
От этого и проще обучаться, и можно проводить вычисления параллельно, в отличие от RNN, где нужно каждый шаг делать по очереди.

Описание архитектуры.
Transformer — это архитектура нейронных сетей, представленная в статье "Attention is All You Need" (Vaswani et al., 2017). Она была разработана в Google для обработки последовательностей данных и стала основой для многих современных моделей в области обработки естественного языка (NLP), таких как BERT и GPT.

Ключевые особенности:
- механизм внимания (self-attention): основная инновация заключается в использовании механизма внимания, который позволяет модели фокусироваться на различных частях входной последовательности независимо от их положения. Это позволяет эффективно обрабатывать длинные зависимости.
- отсутствие рекурсии: в отличие от LSTM и GRU, Transformer не использует рекуррентные слои, что позволяет значительно ускорить обучение и обработку данных.

Сравнение с другими архитектурами:

1. Отличия от RNN/ LSTM:
   - Transformers не зависят от последовательной обработки данных, что позволяет параллелизовать обучение.
   - механизм внимания позволяет учитывать все элементы последовательности одновременно, в то время как RNN обрабатывают данные последовательно.

2. Отличия от CNN:
   - CNN в основном используются для обработки изображений и локальных зависимостей, тогда как Transformers лучше справляются с глобальными зависимостями в последовательностях.

Плюсы и минусы:

Плюсы:
- параллелизация: возможность параллельной обработки данных значительно ускоряет обучение.
- гибкость: эффективно работает с длинными последовательностями и различными типами данных (текст, изображения).
- мощный механизм внимания (self-attention): позволяет модели фокусироваться на наиболее релевантных частях входных данных.

Минусы:
- высокие требования к ресурсам: Transformers требуют значительных вычислительных мощностей и памяти, особенно на больших наборах данных.
- сложность настройки: требуется тщательная настройка гиперпараметров для достижения оптимальных результатов.

Трудности при применении:

1. Объем данных: для достижения хороших результатов требуется большое количество обучающих данных.
2. Обучение: долгое время обучения и необходимость использования специализированного оборудования (GPU/TPU).
3. Интерпретация: сложность понимания работы модели из-за большого количества параметров и отсутствия явных признаков объяснимости.

В целом, Transformer стал революционным шагом в области нейронных сетей, обеспечив значительные улучшения в задачах NLP и за его пределами. Однако его применение требует внимательного подхода к ресурсам и данным.