Методология AlphaNAS

Идея AlphaNAS: данный метод осуществляет автоматический поиск архитектуры нейросети с помощью **эволюционного алгоритма** и **абстрактного представления модели**. В отличие от перебора всех вариантов архитектуры напрямую (привет GridSearch / RandomSearch), AlphaNAS оперирует абстрактными свойствами архитектуры и постепенно эволюционирует их, чтобы удовлетворить критериям качества.

При классическом NAS пространство поиска может быть огромным (разные количества слоёв, типов слоёв, размеров etc). AlphaNAS вводит понятие **абстрактных** свойств – характеристик архитектуры, через которые задаётся конкретная модель. Вместо детального описания каждого соединения сети используется набор свойств, например: глубина сети (число слоёв), размерность скрытых представлений, структура соединений etc. В контексте оптимизации BERT под задачу токсичности главным свойством выступает глубина энкодер части берта, , которые будут задействованы. Другими словами, хочется, чтобы можно было удалить часть слоёв BERT, я использую тривиальный вариант - описываю возможные архитектуры маской, где каждый бит указывает, используется данный слой или нет. Такая бинарная маска длины 12 является абстрактным описанием конкретной подархитектуры BERT: 1 - сохранение соответствующего слоя, 0 – удаление. Дополнительные свойства, связанные с архитектурой (например, количество голов self-attention или размер внутренних слоёв), в данной работе не изменялись, фокусируя поиск только на выборе слоёв, хотя можно было и включить, но решил отказаться в виду малого влияния на ключевую задачу (сжатие модели). Таким образом, абстрактные свойства здесь сведены к шаблону отключения/включения слоёв – это компактное генотип-представление архитектуры для эволюционного алгоритма.

Как вы поняли по тексту выше, необходимо уметь преобразовать абстрактное описание (маску слоёв) обратно в конкретную модель, чтобы оценить её качество. Шаг синтеза архитектуры заключается в порождении подмодели на основе заданных свойств. В нашем случае синтез – это построение урезанной версии берта по заданной маске слоев. Конкретно, из предобученной модели BERT-base для классификации мы выкидываем те слои, которые помечены как 0 в маске, и оставляем лишь слои, отмеченные 1. Полученная нейросеть имеет ту же структуру входов-выходов, что и исходный BERT, но содержит меньше последовательных слоев в энкдоере. При этом мы сохраняем предобученные веса тех слоёв, которые остались. По сути, синтез подграфа BERT реализует заданное свойство глубины: если в маске включено, например, 6 слоёв из 12, то глубина результирующей модели составляет 6.

AlphaNAS использует идеи эволюционных алгоритмов, рассматривая архитектуры как особей (ну или кандидатов), эволюционирующих от поколения к поколению. Алгоритм поддерживает популяцию архитектур (набор различных масок) и итеративно улучшает их через операции мутации и отбора. Процесс можно описать следующим образом:

- **Инициализация популяции**: стартовый набор архитектур генерируется случайно либо на основе исходной полной модели. Мы включаем в начальную популяцию маску полного BERT (все 12 слоев активны) как сильную отправную точку по точности, а также несколько случайно прореженных масок для диверсификации. Размер популяции выбирается относительно небольшим.

- **Оценка качества**: для каждой архитектуры из популяции выполняется синтез модели (создаётся подмодель BERT согласно маске) и вычисляется её качество

на заданной задаче. Оценка включает быструю дообучение на моих imdb данных и измерение метрик на валидационном наборе.

- **Отбор лучших**: из оцененных архитектур текущего поколения отбираются наиболее успешные по моей кастомной метрике качества. Худшие удаляются, лучшие остаются и будут основой для порождения нового поколения. Таким образом, лучшие свойства сохраняются.

- Мутация и воспроизведение: чтобы пополнить популяцию до исходного размера, создаются новые архитектуры-потомки на основе оставшихся кандидатов. Для этого к сохраненным лучшим маскам применяются случайные мутации – небольшие изменения в абстрактных свойствах. Мутация в нашем случае означает случайно переключить несколько битов в маске. Вр еализованном мною подходе задействована только мутация от одного родителя для простоты. Каждая новая маска наследует большинство свойств родителя, но с некоторыми изменениями, вносящими разнообразие. После генерации потомков популяция нового поколения снова оценивается, и процесс повторяется заданное число поколений.

- После нескольких поколений алгоритм останавливается. В результате сохраняется глобально лучшая найденная архитектура – маска, давшая максимальное значение целевой метрики качества за все поколения. Эта маска и есть оптимизированная архитектура BERT для нашей задачи, найденная методом AlphaNAS

Метрика качества и критерии отбора

Оценка наилучшей архитектуры реализована с помощью оценки качества по моей кастомной метрике:

$$Q = A - \lambda N, \tag{1}$$

где:

- A – accuracy на валидационной выборке для данной модели
- N – число обучаемых параметров модели
- λ – коэффициент, задающий степень штрафа за сложность

Кроме того, устанавливаются пороговые ограничения: мы отбрасываем архитектуры, если их точность A падает ниже некоторого минимально допустимого уровня или если число параметров N превышает определённый максимум (например, ограничение по памяти). В реализации мы задали минимальную приемлемую accuracy 0.75 – если модель хуже, дальнейший поиск с ней не имеет смысла, – а максимальное число параметров ограничили 100 млн. Последнее значение чуть ниже размеров базового берта. Эти ограничения ускоряют поиск, отсекая заведомо неприемлемые варианты.

Итак, метод AlphaNAS теоретически опирается на:

- представление архитектуры через абстрактные свойства (в нашем случае – маска слоёв)

- процедуру синтеза модели из этих свойств (вырезание слоёв BERT)
- эволюционный цикл поиска оптимальных свойств (итерации мутаций и отбора)
- целевую функцию качества, учитывающую одновременно точность решения задачи и эффективность модели

Результаты

После запуска алгоритма была найдена более простая архитектура берта - 4 слоя, сжатая модель продемонстрировала почти такую же точность, как и полная. На тестовой выборке 4-слойная модель дала **91.37%** accuracy, в то время как BERT-base достиг около **93.99%**.

По числу параметров выигрыш очевиден: примерно 52.8 миллионов против 109 миллионов, то есть модель меньше более чем в два раза.

Таблица 1: Сравнение полной и оптимизированной моделей

Модель	**Активные слои**	**Параметры, млн**	**Точность на тесте**
BERT-base (полный)	12	109.5	$\sim$93.99%
AlphaNAS-BERT	4	52.8	91.37%