Flamingo: a Visual Language Model for Few-Shot Learning

Лоптев Сергей

Рецензия

Основной вклад

Статья предлагает визуальную языковую модель (Visual LM) Flamingo, способную решать широкий спектр мультимодальных (видео/картинка + текст) задач в few-shot постановке. Модель, обученная на few-shot, ставит рекорд на 6 из 16 анализируемых в статье задач, при этом если дообучить (fine-tune), то будут побиты ещё 5 рекордов.

Контекст

Статья выпущена 29 апреля 2022 года компанией DeepMind. Она также будет представлена на конференции NIPS 2022 года. В самой статье упомянуты 27 авторов, но всего два старших соавтора: Жан-Батист Алейрак (Jean-Baptiste Alayrac) и Карен Симоньян (Karen Simonyan). Обсудим их немного подробнее.
Алейрак достаточно молодой учёный, работает в DeepMind, пока имеет 2636 цитирований и индекс Хирша 22. Пожалуй, самая релевантная данной статье работа — Perceiver IO. В ней Алейрак добавлял модели Perceiver возможность выдавать произвольный (arbitrary) вывод, как и принимать произвольный вход. Модель Perceiver используется в статье для подготовки визуальных токенов, подающихся в языковую модель.
Симоньян — сильно более известный учёный, приложивший руку к большому количеству сильных статей. Он является сооснователем стартапа Inflection AI, занимающегося исследованиями в области AI. У него более 150000 цитирований, из которых более 90000 — цитирования его самой известной работы — модели VGG. Релевантные этой статье работы, соавтором которых является Симоньян — NFNET, которая используется как визуальный кодировщик (visual encoder), и Chinchilla, которая используется как языковая модель.
Я думаю, что статья получилась просто решением авторов объединить сильные модели в одну, чтобы создать визуальную языковую модель, эдакую GPT-3, но с картинками. На мой взгляд, крайне идейных решений тут нет, по сути авторы действительно просто связали несколько моделей адаптерами (Perceiver + Cross-Attention).

Предшественники

Сложно сказать, что у модели есть какие-то непосредственные предшественники. До этого в области Visual LM в целом были какие-то подходы, но я бы не сказал, что Flamingo вдохновилась какой-то одной из них. Тем не менее, авторы перечисляют модели в этой области, которые используют похожие идеи на те, что используются в статье. Это MAGMA (Eichenberg et al., 2021), ClipCap (Mokady et al., 2021), VC-GPT (Luo et al., 2022), PICA (Yang et al., 2021), Socratic (Zeng et al., 2022). Конкретно общие идеи:

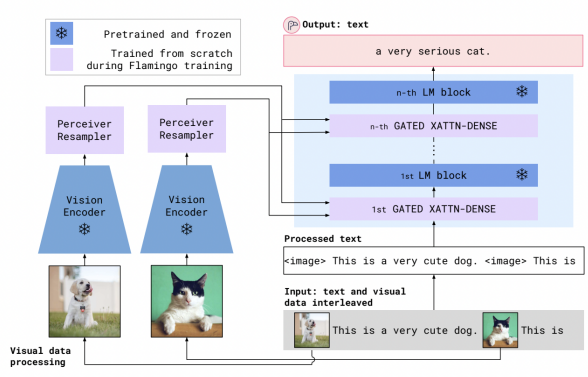
- Используется большая замороженная (frozen) языковая модель;
- Между языковой моделью и визуальным кодировщиком используется модель-адаптер (mapper), основанная на архитектуре трансформеров;
- Между слоями замороженной языковой модели обучаются слои перекрёстного внимания (cross-attention).

Продолжения

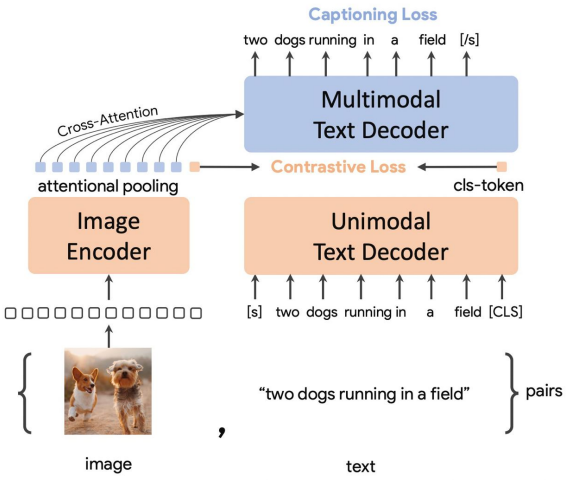
Модель довольно новая, была выпущена около полугода назад. Соответственно, больших прямых статей-продолжений ещё не вышло. Из того, что я смог найти, из имён, которые на слуху, процитировала Flamingo статья про выборку LAION-5B. Flamingo упоминается там как первая визуальная языковая модель в том контексте, что для обучения таких моделей требуются выборки, содержащие миллиарды объектов, какой и является LAION-5B.

Конкуренты

В момент выпуска Flamingo у неё особенно не было конкурентов, я бы сказал, что это была единственная настолько успешная и притом универсальная в few-shot постановке модель. Сейчас, спустя полгода, вышла статья CoCa: Contrastive Captioners are Image-Text Foundation Models. Она, помимо становления state-of-the-art моделью на выборке ImageNet, позиционируется как модель, которая может успешно справляться с различными мультимодальными задачами. В самой статье CoCa Flamingo не цитируется, но сравнить их было бы интересно. Думаю, CoCa и Flamingo отличаются фундаментально: если Flamingo обучает только свой визуальный кодировщик (архитектурно довольно небольшая часть модели) на ContrastiveLoss, то внутри CoCa весь кодировщик изображений (image encoder) обучается на ContrastiveLoss. Архитектуры приведены на картинке 1.



(a) Архитектура Flamingo



(b) Архитектура CoCa

Рис. 1: Сравнение архитектур

Сильные стороны работы

В статье есть отдельная секция Discussion, в которой обсуждаются плюсы и минусы работы, а также направления дальнейшей работы. Обсудим это, начнём с плюсов:

- Модель получилась действительно успешной: обученная на few-shot модель бьёт рекорд в 6 из 16 анализируемых задач, а если дообучать её, то она побьёт ещё 5. Качественные примеры показывают, что в большинстве случаев она также работает хорошо. Это говорит о значимости вклада данной работы.

- Также модель получилась универсальной. Она обучается без каких-то ограничений, то есть авторы не пытаются подогнать её под какую-то задачу. Она может отвечать на открытые вопросы, и в том числе её можно использовать через интерфейс в виде чата. А возможность использовать её в постановке few-shot позволяет использовать Flamingo в задачах, в которых недостаточно данных для дообучения модели. Я думаю, такое универсальное применение приносит актуальность работы Flamingo во всей области Computer Vision в целом.

- На мой взгляд, сама статья хорошо написана. Она большая (66 страниц), все идеи объяснены подробно, я не заметил некорректных утверждений. Отдельно хочется заметить, что в модели достаточно обширный ablation study, на мой взгляд, там рассмотрены почти все моменты. Что важно для воспроизводимости, достаточно подробно рассказаны все технические подробности обучения, как и приведены все гиперпараметры. Также хотелось бы отметить аккуратное тестирование модели. К примеру, авторы говорят, что подбор запросов (prompt engineering) — это просто ещё один вид few-shot learning'а, поэтому почти во всех экспериментах эти запросы стандартизированы. Наконец, хотя исследование и ставит перед собой первостепенную задачу создать мощную модель

в постановке few-shot, анализируется также и дообученная модель, и показывается, что Flamingo способна улучшаться при дообучении.

- Ещё один плюс здесь — это экологичность работы. В работе используется полностью замороженная языковая модель. Авторы отмечают с одной стороны практичность данного решения, а с другой — экологичность, так как обучать такие модели (на 70 миллиардов параметров) крайне энергозатратно.

Слабые стороны работы

Сразу приступим к перечислению минусов работы.

- Несмотря на общую производительность модели, нельзя не заметить большое отставание от моделей, обучающихся на ContrastiveLoss, в качестве классификации изображений. По мнению авторов, такое отставание связано с особенностями обучения моделей: если ContrastiveLoss учит модель непосредственно text-image retrieval, то Flamingo пытается учиться для более широкого спектра задач. Более того, чистая NFNet6 (визуальный кодировщик внутри Flamingo) работает для классификации изображений лучше, чем вся Flamingo.

- Так как модель использует языковые модели, она берёт из них и их минусы. Во-первых, Flamingo однонаправленно подаёт визуальные токены в языковую модель, хотя другими исследованиями доказано, что двунаправленное использование улучшает качество модели, в том числе для широкого спектра задач. Затем, языковые модели, основанные на трансформерах, плохо генерализуются для текстов длиннее, чем во время обучения. Это также доказывается экспериментами. Наконец, авторы подозревают, что замороженные веса модели могут вести к галлюцинациям модели, где она выдает ответ, в целом вероятный, если только смотреть на текст, но не соответствующий картинке.

- Также модель использует in-context learning в качестве метода обучения для few-shot learning. Данный метод обладает рядом преимуществ перед методами, основанными на градиентах, но также обладает и некоторыми недостатками. Во-первых, в зависимости от реализации, время на применении растёт линейно или квадратично в зависимости от числа примеров (shots). Во-вторых, данный метод достаточно чувствителен к запросу (prompt), а именно к самому набору примеров и к их порядку. Наконец, качество метода выходит на плато после 32 примеров. Авторы приводят исследования, изучающие in-context learning, в которых говорится, что данный метод на самом деле не учит модель напрямую, а пытается понять формат задачи (task location). Поэтому как раз после 32 примеров метод уже понимает формат задачи и перестаёт извлекать новую информацию из примеров.

- Также я хотел бы отметить следующий минус: все результаты работы останутся приватными, авторы пишут, что они будут использоваться только для последующей работы внутри DeepMind. Очевидно, что это плохо влияет на используемость и цитируемость работы.

Предложения по улучшению

- В работе нет ablation study на выбор языковой модели. Модель Chinchilla используется безоговорочно, и авторы не рассматривают какие-то другие модели типа BERT или T5-XXL.
- Также не рассматриваются различные способы few-shot learning, кроме in-context learning. На мой взгляд, такое исследование бы могло пролить свет на трейд-оффы между разными методами few-shot обучения для данной модели.
- Во время формирования выборки M3W авторы собирают веб-страницы с включенными фотографиями, но почему-то не собирают веб-страницы с включенными видео. Я думаю, это было бы несложно сделать и могло бы улучшить качество на задачах, связанных с видео.

Предложения по следующему исследованию

Вообще, в секции Discussion авторы сами говорят о нескольких идеях следующих исследований. Я перечислю их тут, и также приведу какие-то свои мысли.

- Изначально Flamingo задумывалась именно как универсальная модель. С этой точки зрения хорошо было бы попробовать расширить спектр задач, с которыми она может справляться. Это может быть либо добавление другой модальности (например, ничего не мешает добавить кодировщик, подобный визуальному, для аудио), либо подумать над тем, как научить Flamingo распознавать пространственно-временные отношения между объектами (например, такие задачи, как bounding box или predicting optical flow).
- Исследователи уже выявили законы масштабирования для языковых моделей, то есть, мы умеем эффективно обучать большие языковые модели (на десятки миллиардов параметров). Хотелось бы научиться делать такое же для визуальных моделей, кажется, что это может принести практическую пользу.
- Тема, наверное, недостойная отдельного исследования, но мне хотелось бы детально сравнить разницу между Flamingo и CoCa на том спектре задач, на котором анализируется Flamingo. Кажется, обе модели достаточно разные, со своими плюсами и минусами, но обе достаточно универсальные, чтобы такое сравнение устроить.
- С точки зрения применения в индустрии, я думаю, что отдельную пользу приносит именно few-shot постановка, потому что это открывает среднему и малому бизнесу потенциальную возможность эффективно использовать Flamingo для каких-то своих задач. Пример — проверка качества ткани для произведения футболок. Пользователю достаточно добавить 32 размеченных изображения такой ткани, и затем можно использовать Flamingo как классификатор изображений такой ткани. Также, учитывая, что дообученный Flamingo ставит рекорд на задаче HatefulMemes, можно было

бы использовать её как модель для фильтрации токсичных и оскорбляющих объектов в какой-то выборке.

Ещё что-то любопытное

Из того, что пришло в голову — модель может понимать шутки. Вот пример:

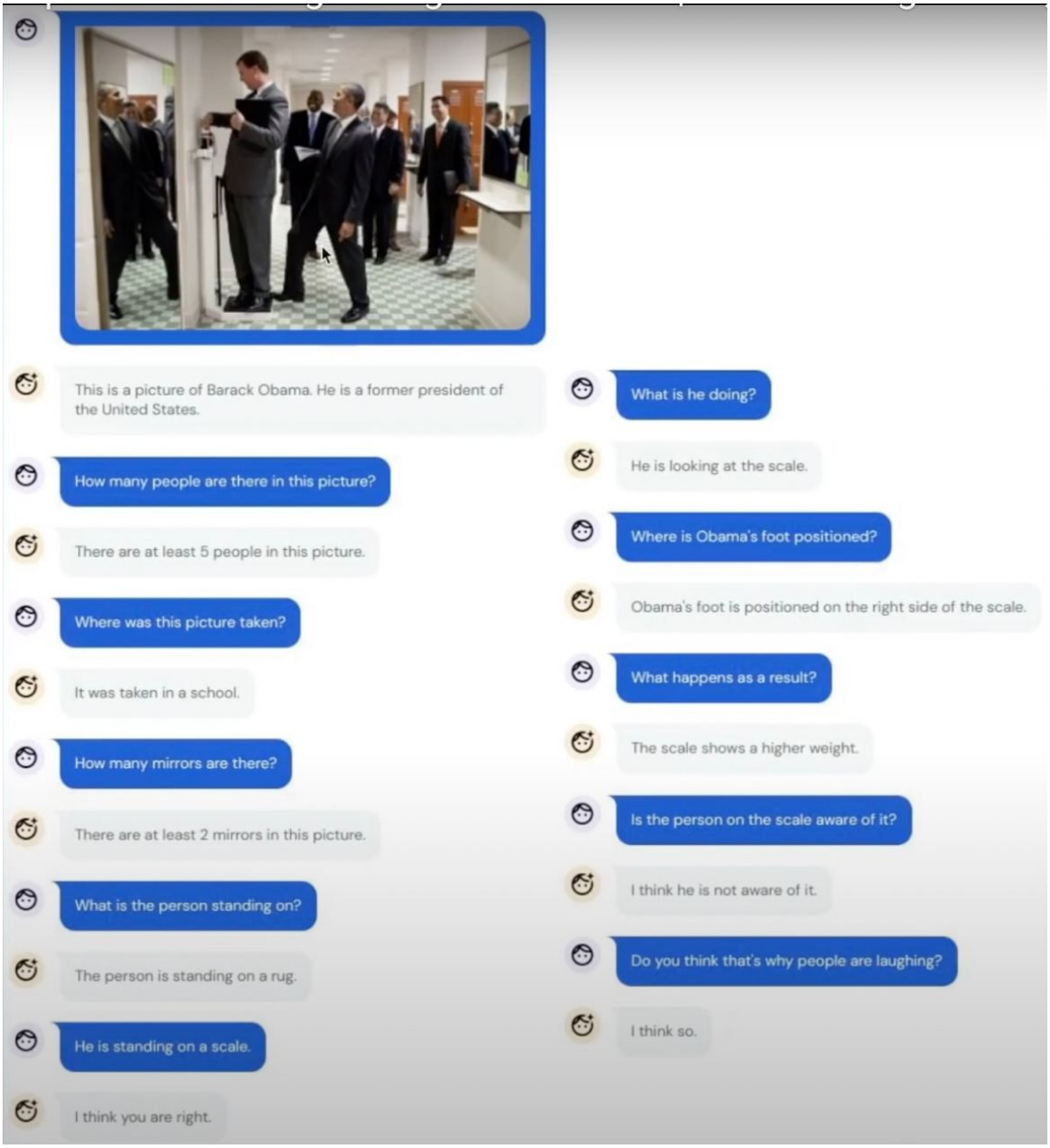


Рис. 2: Сравнение архитектур