INCEPTION-V4, INCEPTION-RESNET И ВЛИЯНИЕ ОСТАТОЧНЫХ СВЯЗЕЙ НА ОБУЧЕНИЕ

Кенжаев Шахзодбек Янгибаевич
Национальный университет Узбекистана имени Мирзо Улугбека
https://doi.org/10.5281/zenodo.7841841

Аннотация. Очень глубокие сверточные сети сыграли ключевую роль в самых больших достижениях в области распознавания изображений за последние годы. Одним из примеров является архитектура Inception, которая показала очень хорошую производительность при относительно низких вычислительных затратах. Недавнее введение остаточных соединений в сочетании с более традиционной архитектурой позволило добиться самых современных результатов в испытании ILSVRC 2015 года; его производительность была аналогична сети Inception-v3 последнего поколения. В связи с этим возникает вопрос: есть ли какие-то преимущества в объединении начальных архитектур с остаточными соединениями? Здесь я приводу четкие эмпирические доказательства того, что обучение с остаточными связями значительно ускоряет обучение начальных сетей. Есть также некоторые свидетельства того, что остаточные начальные сети с небольшим отрывом опережают аналогичные дорогие начальные сети без остаточных подключений. Мы также представляем несколько новых оптимизированных архитектур как для остаточных, так и для неостаточных начальных сетей. Эти варианты значительно улучшают производительность распознавания одиночного кадра в задаче классификации ILSVRC 2012. Далее мы демонстрируем, как правильное масштабирование активации стабилизирует обучение очень широких остаточных начальных сетей.

Ключевые слова. Классификация , ImageNet сверточные сети, Inception, Распознавания изображений, Искуственный интеллект, Нейронные сети.

ВВЕДЕНИЕ

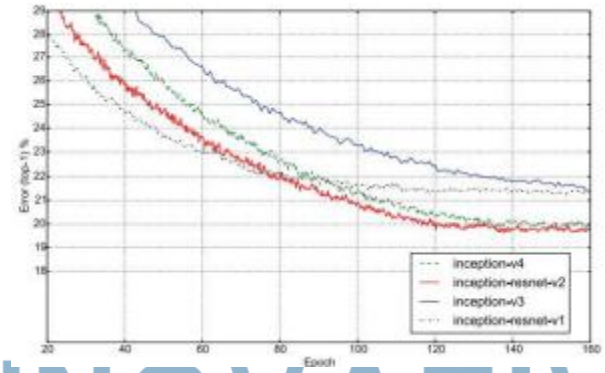
Распознавание объектов является центральной задачей компьютерного зрения и искусственного интеллекта в целом. Модели сильного зрения являются ключевыми компонентами систем ИИ, которые могут обрабатывать визуальные входные данные. Их приложения включают интерфейс пользователя компьютера (распознавание жестов), веб-поиск, системы OCR, автономный транспорт, медицинскую визуализацию, визуализацию местности, робототехнику и обработку изображений. До 2012 года для каждой конкретной области применения требовались специализированные решения. С тех пор для решения этих задач стали широко применяться глубокие сверточные нейронные сети. Сверточные нейронные сети восходят к 1980-м годам (Фукусима, 1980 г.) и (Лекун и др., 1989 г.), но недавние хорошие результаты (Крижевский, Суцкевер и Хинтон, 2012 г.) в крупномасштабном эталонном тесте распознавания изображений ImageNet ILSVRC (Русаковский и др., 2014 г.)) привело к возрождению интереса к их использованию. 1600 Амфитеатр Паркуэй Маунтин-Вью, Калифорния Эта универсальная применимость мотивирует наше внимание к моделям распознавания для широко используемого теста распознавания объектов ILSVRC12 (Русаковский и др., 2014), в котором задача состоит в том, чтобы классифицировать изображения в один (или пять) из тысячи различных классов. Набор данных состоит из 1,2 миллиона обучающих изображений, 50 000 проверочных изображений и 100 000 тестовых

изображений. Все они поделены поровну между 1000 классами. Этот бенчмарк является очень популярной задачей для измерения качества решений по распознаванию объектов с 2010 года. Авторы утверждали, что остаточные связи по своей сути необходимы для обучения очень глубоких сверточных моделей. Наши результаты, кажется, не подтверждают эту точку зрения, по крайней мере, для распознавания изображений. Однако может потребоваться больше экспериментов с еще более глубокими сетями, чтобы полностью понять истинные преимущества остаточных соединений. В экспериментальной части мы показываем, что несложно обучить очень глубокие конкурентные сети без использования остаточных соединений. Однако использование остаточных соединений, по-видимому, значительно повышает скорость обучения, что само по себе является большим аргументом в пользу их использования. др. 2015b), который в этом отчете будет называться Inception-v3. Наконец, мы сообщаем об оценке ансамбля всех описанных моделей. Поскольку было очевидно, что и Inception-v4, и Inception-ResNet-v2 работали одинаково хорошо, каждый из которых превосходил современную производительность одиночного кадра в наборе данных проверки ImageNet, мы хотели увидеть, как комбинация подталкивает состояние. в этом хорошо изученном наборе данных. Полная конфигурация сети Inception-v4 показана на рис. 2, который содержит общую схему и конфигурацию ствола, и на рис. 3, на котором подробно показана конструкция внутренних модулей. Слева исходное остаточное соединение. Справа — оптимизированная версия, которая снижает вычислительные затраты за счет использования свертки 1 × 1. В этой статье мы сравним два чистых варианта Inception, Inception-v3 и v4, с такими же дорогими гибридными версиями Inception-ResNet. По общему признанию, эти модели были выбраны несколько случайным образом с основным ограничением, которое заключалось в том, что параметры и вычислительная сложность моделей должны быть в чем-то схожи со стоимостью неостаточных моделей. Мы протестировали более крупные и широкие варианты Inception-ResNet, и они работали на уровне Inception-ResNet-v2 в наборе данных для задачи классификации ImageNet (Russakovsky et al. 2014). Чистые начальные блоки Наши старые начальные модели обучались раздельным образом, когда каждая реплика была разделена на несколько подсетей, чтобы можно было уместить всю модель в памяти. Тем не менее, начальная архитектура обладает широкими возможностями настройки, а это означает, что существует множество возможных изменений количества фильтров на различных уровнях, которые не влияют на качество полностью обученной сети. Чтобы оптимизировать скорость обучения, мы тщательно настраивали размеры слоев, чтобы сбалансировать вычисления между различными подсетями модели. Напротив, с введением TensorFlow (Абади и др., 2015) наши самые последние модели можно обучать без разделения реплик. Частично это стало возможным благодаря недавней оптимизации памяти для обратного распространения, достигнутой тщательным рассмотрением того, какие тензоры необходимы для вычисления градиента, и структурированием вычислений для уменьшения количества таких тензоров. Исторически сложилось так, что мы были относительно консервативны в отношении изменения архитектуры и ограничивали наши эксперименты различными изолированными сетевыми компонентами, сохраняя при

этом стабильность остальной части сети. Отсутствие упрощения предыдущих вариантов привело к тому, что сети выглядели более сложными, чем должны были быть. В наших новых экспериментах для Inception-v4 мы решили избавиться от этого ненужного багажа и сделали одинаковый выбор для блоков Inception для каждого размера сетки. Остаточные начальные блоки Для остаточных версий начальных сетей мы используем более дешевые начальные блоки, чем исходные начальные. За каждым начальным блоком следует слой расширения фильтра (свертка 1 × 1 без активации), который используется для увеличения размерности банка фильтров перед остаточным добавлением, чтобы соответствовать глубине ввода. Остаточные соединения, представленные в (He et al. необходимость разделения модели для распределенного обучения с использованием DistBelief (Дин и др., 2012). Теперь, после переноса нашей обучающей установки на TensorFlow (Абади и др., 2015), эти ограничения были сняты, что позволило нам значительно упростить архитектуру. Детали этой упрощенной архитектуры описаны в разделе «Выбор архитектуры», начиная со стр. 2. Глубокая сверточная архитектура Inception была представлена как GoogLeNet в (Szegedy et al. 2015a) и названа здесь Inception-v1. Позже архитектура Inception была усовершенствована различными способами, сначала путем введения пакетной нормализации (Иоффе и Сегеди, 2015) (Inception-v2). Позже с помощью дополнительных идей факторизации в третьей итерации (Szegedy et Удивительно, но мы обнаружили, что прирост производительности одиночного кадра не приводит к столь же значительному приросту производительности ансамбля. Тем не менее, это по-прежнему позволяет нам сообщить о 3,1% ошибки топ-5 в проверочном наборе с четырьмя объединенными моделями, устанавливающими новый уровень техники, насколько нам известно. Даже там, где масштабирование не было строго необходимым, оно никогда не снижало конечной точности, но помогало стабилизировать обучение. Аналогичная нестабильность наблюдалась (He et al. 2015) в случае очень глубоких остаточных сетей, и они предложили двухэтапное обучение, при котором первая фаза «разогрева» выполняется с очень низкой скоростью обучения, за которой следует вторая. фаза с высокой скоростью обучения. Мы обнаружили, что если количество фильтров очень велико, то даже очень низкой (0,00001) скорости обучения недостаточно, чтобы справиться с нестабильностью, а обучение с высокой скоростью обучения имело шанс разрушить его эффекты. Экспериментальные результаты. Сначала мы наблюдаем за эволюцией ошибок валидации первых 1 и 5 первых вариантов четырех вариантов во время обучения. После проведения эксперимента мы обнаружили, что наша непрерывная оценка проводилась на подмножестве проверочного набора, в котором было пропущено около 1700 объектов из черного списка из-за плохих ограничивающих рамок. Оказалось, что это упущение должно было быть выполнено только для эталонного теста CLSLOC, но дает несколько несопоставимые (более оптимистичные) цифры по сравнению с другими отчетами, включая некоторые более ранние отчеты нашей команды. Разница составляет около 0,3% для первой ошибки и около 0,15% для ошибки первой пятерки. Однако, поскольку различия постоянны, мы считаем, что сравнение между кривыми является справедливым. Мы попробовали несколько версий остаточной версии Inception. Здесь подробно описаны только два из них. Первая «Inception-ResNet-v1» имеет примерно

такую же вычислительную стоимость, как Inception-v3, а «Inception-ResNet-v2» соответствует необработанной стоимости недавно представленной сети Inception-v4. Полная конфигурация сети Inception-Resnet-v2 использует схему на рисунке 6, основу на рисунке 2 и модули на рисунке 5. Методология обучения Мы обучили наши сети со стохастическим градиентом, используя распределенную систему машинного обучения TensorFlow (Abadi et al. 2015) с использованием 20 реплик, каждая из которых работает под управлением графического процессора NVidia Kepler. В наших более ранних экспериментах использовался импульс (Сутскевер и др., 2013) с затуханием 0,9, в то время как наши лучшие модели были получены с использованием RMSProp (Тилеман и Хинтон) с затуханием 0,9 и = 1,0. Мы использовали скорость обучения 0,045, затухая каждые две эпохи, используя экспоненциальную скорость 0,94. Кроме того, было обнаружено, что отсечение градиента (Pascanu, Mikolov, and Bengio 2012) полезно для стабилизации обучения. Оценки модели выполняются с использованием скользящего среднего значения параметров, рассчитанных с течением времени. Мы обнаружили, что уменьшение остаточных значений перед добавлением их к активации предыдущего слоя, по-видимому, стабилизировало обучение. В общем, мы выбрали несколько коэффициентов масштабирования от 0,1 до 0,3, чтобы масштабировать остатки до того, как они будут добавлены к накопленным активациям слоя (см. рис. 7). компенсировать уменьшение размерности, вызванное начальным блоком. Полная конфигурация сети Inception-Resnet-v1 показана на рис. 6, который содержит общую схему и конфигурацию ствола, и на рис. 4, на котором показана подробная конфигурация внутренних модулей. С другой стороны, мы повторно обработали наши результаты мультикадрирования и ансамбля на полном проверочном наборе, состоящем из 50 000 изображений. Кроме того, окончательный результат ансамбля также был выполнен на тестовом наборе и отправлен на тестовый сервер ILSVRC.Еще одно небольшое техническое различие между нашими остаточными и неостаточными вариантами Inception заключается в том, что в наших экспериментах Inception ResNet мы использовали пакетную нормализацию только поверх традиционных слоев, но не поверх остаточных сумм. Разумно ожидать, что тщательное использование пакетной нормализации должно быть выгодным, но реализация пакетной нормализации в TensorFlow потребляла много памяти, и нам пришлось бы уменьшить общее количество слоев, если бы пакетная нормализация -нормализация применялась повсеместно. Рис. 2. Слева представлена общая схема чистой сети Inception-v4. Справа подробный состав стебля. Обратите внимание, что эта конфигурация основы также использовалась для контуров сети Inception-ResNet-v2 на рисунках 5, 6. V обозначает использование «Действительного» заполнения, в противном случае использовалось «То же самое» заполнение. Размеры сбоку от каждого слоя суммируют форму вывода для этого слоя. Мы обнаружили, что гораздо надежнее масштабировать остатки. Однако на практике время шага Inception-v4 оказалось значительно меньше, вероятно, из-за большего количества слоев. 4280 Масштабирование остатков Мы обнаружили, что если количество фильтров превышало 1000, остаточные варианты начинали демонстрировать нестабильность, и сеть просто «умирала» в начале обучения, а это означало, что последний слой перед объединением средних значений начинал давать только нули. после нескольких десятков тысяч итераций. Этого нельзя было

предотвратить ни снижением скорости обучения, ни добавлением в этот слой дополнительной пакетной нормализации. Оценка измеряется на одной культуре изображений, не входящих в черный список, из проверочного набора ILSVRC-2012. Рисунок 8: Эволюция ошибок Top 1 во время обучения чистой Inception-v3 по сравнению с остаточной сетью с аналогичными вычислительными затратами. Оценка измеряется на одной культуре на изображениях, не включенных в черный список, из набора проверки ILSVRC-2012. Остаточная модель обучалась намного быстрее, но достигла конечной точности немного хуже, чем традиционная Inception-v3. Остаточная версия тренировалась намного быстрее и достигла немного лучшей конечной точности, чем традиционная версия Inception v4. Общая схема масштабирования комбинированных модулей Inception-ResNet. Мы ожидаем, что эта же идея будет полезна и в общем случае ResNet, где вместо блока Inception используется произвольная подсеть. Блок масштабирования просто масштабирует последние линейные активации на подходящую константу, обычно около 0,1, но мы обнаружили, что более глубокие сети требуют меньших констант



. Слева представлена общая схема сети Inception-Resnet-v1 и Inception-Resnet-v2. Хотя схема одинакова для обеих сетей, состав стволовых и внутренних модулей различается. Основа Inception Resnetv1 показана справа, а основа Inception Resnet-v2 такая же, как чистая сеть Inception-v4, обозначает использование «Действительного» заполнения, в противном случае использовалось «То же самое» заполнение. Наконец, мы представляем некоторые сравнения между различными версиями Inception и Inception-ResNet. Модели Inceptionv3 и Inception-v4 представляют собой глубокие сверточные сети, не использующие остаточные соединения, в то время как Inception ResNet-v1 и Inception-ResNet-v2 представляют собой сети в стиле Inception, которые используют остаточные соединения вместо конкатенации фильтров. Размеры сбоку от каждого слоя суммируют форму вывода для этого слоя. для проверки, чтобы убедиться, что наша настройка не привела к перенастройке.

Выводы

Мы подробно представили три новые сетевые архитектуры: • Inception-ResNetv1: гибридная версия Inception с такими же вычислительными затратами, что и Inception-v3 из (Szegedy et al. 2015b). Реализации моделей Inception-ResNet-v2 и Inception-v4 с открытым исходным кодом, описанные в этой статье, а также предварительно обученные веса доступны на странице github моделей TensorFlow: github.com/tensorflow/models

References:

1. Сегеди, К.; Лю, В .; Цзя, Ю .; Серманет, П .; Рид, С.; Ангелов, Д .; Эрхан, Д.; Ванхоук, В.; и Рабинович, А. 2015a.
2. Тошев А. и Сегеди К. 2014. Deeppose: оценка позы человека с помощью глубоких нейронных сетей. В компьютерном зрении и распознавании образов (CVPR), конференция IEEE 2014 г., 1653–1660. IEEE.
3. Ван, Н., и Юнг, Д.-Ю. 2013. Изучение глубокого компактного представления изображения для визуального отслеживания. В достижениях в области систем обработки нейронной информации, 809–817.
4. Карпаты, А .; Тодеричи, Г.; Шетти, С .; Люнг, Т .; Сук Танкар, Р .; и Fei-Fei, L. 2014. Крупномасштабная классификация видео с помощью сверточных нейронных сетей. В книге «Компьютерное зрение и распознавание образов» (CVPR), конференция IEEE 2014 г., 1725–1732 гг. IEEE.