Московский государственный технический университет им. Н.Э.

Баумана

Кафедра «Системы обработки информации и управления»

Домашнее Задание

по дисциплине

«Методы машинного обучения»

Выполнил:

студент группы ИУ5И-23М

Ли Хао

Москва — 2024 г.

1. Требование

Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

- выбор задачи;
- теоретический этап;
- практический этап.

Этап выбора задачи предполагает анализ ресурса paperswithcode. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом. Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;

конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;

- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;

- описание наборов данных, используемых для обучения моделей;

- оценка качества решения задачи, описание метрик качества и их значений;

- предложения обучающегося по улучшению качества решения задачи. Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозиториев с исходным кодом и возможное улучшение обучающимися полученных результатов. Результаты проработки обучающийся излагает в практической части отчета по домашнему заданию, которая может включать:

- исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;

- результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;

- предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

Выбранная задача: «Классификация изображений»

2. Выбор задачи

Классификация изображений – это процесс извлечения классов информации из многоканального растрового изображения. Растр, полученный в результате классификации изображения, можно использовать для создания тематических карт. В зависимости от характера взаимодействия аналитика с компьютером в процессе классификации, различают два типа классификации изображений: классификацию с обучением и классификацию без обучения.

3. Теоретический этап

Часть I

Тема: «Глубокое остаточное обучение для распознавания изображений»

Глубокие конволюционные нейронные сети привели к ряду прорывов в классификации изображений. Глубокие сети естественным образом объединяют признаки низкого/среднего/высокого уровня и классификаторы в сквозном многослойном режиме, а "уровни" признаков могут быть обогащены количеством слоев (глубиной). Последние данные показывают, что глубина сети имеет решающее значение, и ведущие результаты на сложном наборе данных ImageNet все используют "очень глубокие" модели, с глубиной от шестнадцати до тридцати. Многие другие нетривиальные задачи визуального распознавания также значительно выиграли от использования очень глубоких моделей.

Когда более глубокие сети начинают сходиться, обнаруживается проблема деградации: с увеличением глубины сети точность насыщается, а затем быстро деградирует. Неожиданно, но такая деградация не вызвана чрезмерной

подгонкой, а добавление большего количества слоев в достаточно глубокую модель приводит к увеличению ошибки обучения.

В данной работе мы решаем проблему деградации, внедряя систему глубокого остаточного обучения. Вместо того, чтобы надеяться, что каждый из нескольких слоев непосредственно соответствует желаемому базовому отображению, мы явно позволяем этим слоям соответствовать остаточному отображению. Формально, обозначая желаемое базовое отображение как H(x), мы позволяем сложенным нелинейным слоям соответствовать другому отображению F(x) := H(x)-x. Исходное отображение преобразуется в F(x)+x.

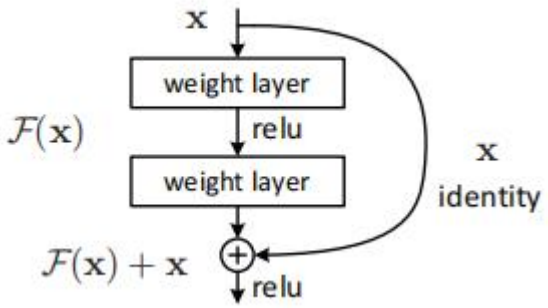


Рис 1. Остаточное обучение: строительный блок

Формула F(x)+x может быть реализована с помощью фидфорвардных нейронных сетей с "короткими связями" (рис. 1). Короткие связи - это связи, пропускающие один или несколько слоев. В нашем случае короткие соединения просто выполняют отображение идентичности, а их выходы добавляются к выходам сложенных слоев (рис. 1). Ярлыковые соединения идентичности не добавляют ни дополнительных параметров, ни вычислительной сложности. Вся сеть по-прежнему может быть обучена сквозным методом SGD с обратным распространением и может быть легко реализована с помощью распространенных библиотек без модификации решателей.

Рассмотрим H(x) как базовое отображение, которое должно быть подогнано несколькими слоями (не обязательно всей сетью), при этом x обозначает входы

первого из этих слоев. Если существует гипотеза, что несколько нелинейных слоев могут асимптотически аппроксимировать сложные функции2 , то эквивалентно гипотезе, что они могут асимптотически аппроксимировать остаточные функции, т.е. H(x) - x (при условии, что вход и выход имеют одинаковые размеры). Поэтому вместо того, чтобы ожидать, что слои будут аппроксимировать H(x), мы явно позволяем этим слоям аппроксимировать остаточную функцию F(x) := H(x) - x. Таким образом, исходная функция становится F(x)+x.

Проблема деградации предполагает, что решатели могут испытывать трудности при аппроксимации тождественных отображений несколькими нелинейными слоями. При использовании переформулировки с остаточным обучением, если тождественные отображения являются оптимальными, решатели могут просто направить веса нескольких нелинейных слоев к нулю, чтобы приблизиться к тождественным отображениям.

Часть II

Тема: «MobileNets: эффективные конволюционные нейронные сети для приложений мобильного зрения»

Конволюционные нейронные сети получили широкое распространение в компьютерном зрении с тех пор, как AlexNet популяризировал глубокие конволюционные нейронные сети, победив в конкурсе ImageNet Challenge: ILSVRC 2012. Общая тенденция заключается в создании более глубоких и сложных сетей для достижения более высокой точности. Однако эти достижения в области повышения точности не обязательно делают сети более эффективными в отношении размера и скорости. Во многих реальных приложениях, таких как робототехника, самоуправляемые автомобили и

дополненная реальность, задачи распознавания должны выполняться своевременно на ограниченной по вычислительным ресурсам платформе.

Mobilenets строятся в основном из разделяемых по глубине сверток, первоначально введенных в моделях Inception и впоследствии использовавшихся в них для уменьшения вычислений в первых нескольких слоях. Flattened networks строят сеть из полностью факторизованных сверток и показали потенциал чрезвычайно факторизованных сетей. Независимо от данной работы, Factorized Networks представляет аналогичную факторизованную свертку, а также использование топологических связей. Впоследствии сеть Xception продемонстрировала, как масштабировать сепарабельные фильтры глубинного типа, чтобы превзойти сети Inception V3. Еще одна небольшая сеть - Squeezenet, которая использует подход узкого места для проектирования очень маленькой сети. Другие сети с уменьшенными вычислениями включают сети структурированных преобразований и глубокие конвентные сети.

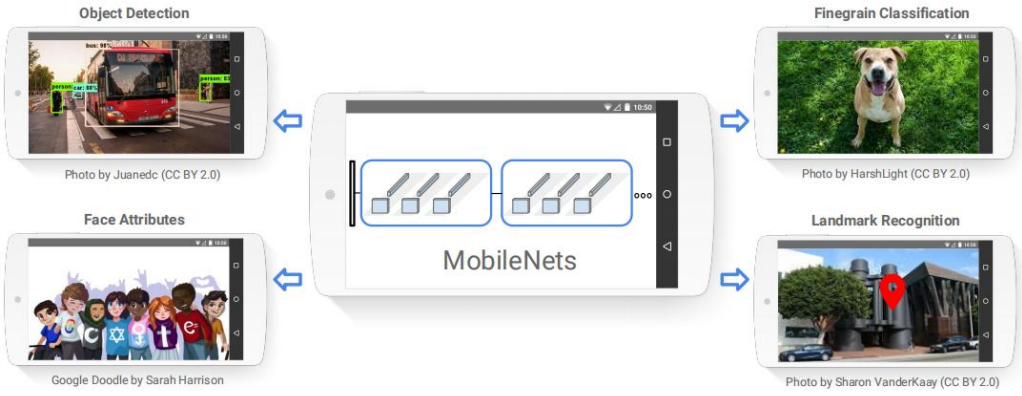
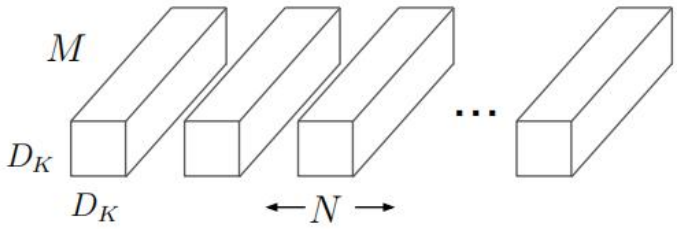


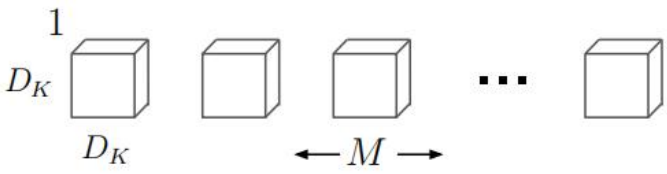
Рис 2. Модели MobileNet можно применять к различным задачам распознавания для эффективной интеллектуальной работы на устройстве

Модель MobileNet основана на глубинных сепарабельных свертках, которые являются формой факторизованных сверток, которые факторизуют стандартную свертку на глубинную свертку и свертку 1×1, называемую точечной сверткой.

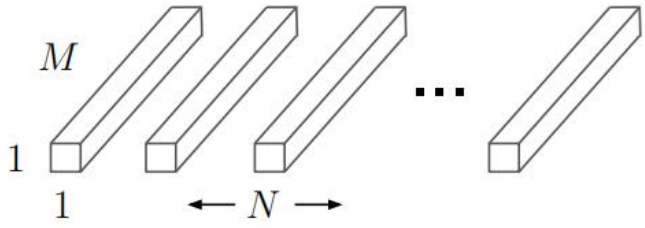
Для мобильных сетей свертка по глубине применяет один фильтр к каждому входному каналу. Точечная свертка затем применяет свертку 1×1 для объединения выходов глубинной свертки. Стандартная свертка одновременно фильтрует и объединяет входы в новый набор выходов за один шаг. Разделяемая по глубине свертка разделяет это на два слоя, отдельный слой для фильтрации и отдельный слой для объединения. Эта факторизация имеет эффект резкого сокращения вычислений и размера модели. На рисунке 6 показано, как стандартная свертка 2(a) факторизуется в глубинную свертку 2(b) и точечную свертку 1 × 1 2(c).



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1 × 1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Рис 3. Стандартные сверточные фильтры в (a) заменены на два слоя: свертка по глубине в (b) и свертка по точке в (c) для построения разделительного фильтра по глубине.

Структура MobileNet построена на глубинных сепарабельных свертках, как упоминалось в предыдущем разделе, за исключением первого слоя, который представляет собой полную свертку. Определяя сеть в таких простых терминах, мы можем легко исследовать топологии сети, чтобы найти хорошую сеть. Архитектура MobileNet представлена в таблице 1. Все слои сопровождаются нелинейностью batchnorm и ReLU, за исключением последнего полностью связанного слоя, который не имеет нелинейности и подается на слой softmax для классификации. На рисунке 3 показано сравнение слоя с обычными свертками, нелинейностью batchnorm и ReLU с факторизованным слоем с глубинной сверткой, точечной сверткой 1 × 1, а также batchnorm и ReLU после каждого сверточного слоя. Нисходящая выборка обрабатывается с помощью свертки по горизонтали в свертках по глубине, а также в первом слое. Окончательное объединение средних снижает пространственное разрешение до 1 перед полностью связного слоя. Учитывая глубинные и точечные свертки как отдельные слои, MobileNet имеет 28 слоев.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Таблица 1. Структура MobileNet

3. Практическая часть

Часть I

Тема: «Глубокое остаточное обучение для распознавания изображений»

ImageNet Classification

Мы оцениваем наш метод на наборе данных классификации ImageNet 2012, состоящем из 1000 классов. Модели обучаются на 1,28 млн. обучающих изображений и оцениваются на 50 тыс. проверочных изображений. Мы также получаем окончательный результат на 100 тыс. тестовых изображений, о которых сообщает тестовый сервер. Мы оцениваем как топ-1, так и топ-5 по количеству ошибок.

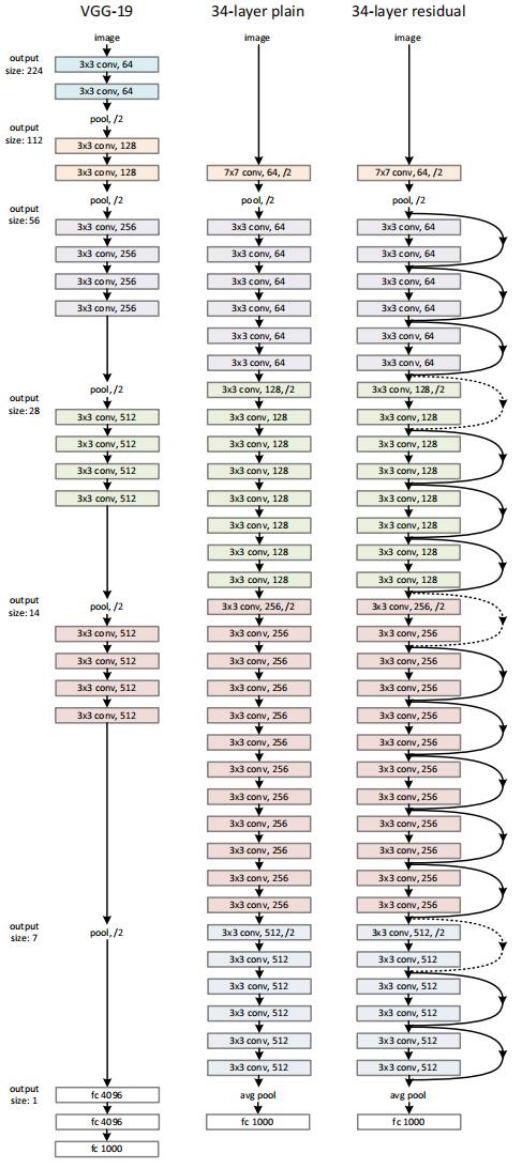


Рис 4. Примеры сетевых архитектур для ImageNet.

Простые сети.

Сначала мы оценим 18- и 34-слойные простые сети. 34-слойная простая сеть показана на рис. 2 (в середине). 18-слойная простая сеть имеет аналогичную форму. Подробную информацию об архитектуре см. в таблице 1.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2 [3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Таблица 2. Структура для ImageNet.

Результаты в таблице 2 показывают, что более глубокая 34-слойная простая сеть имеет большую ошибку валидации, чем более мелкая 18-слойная простая сеть. Чтобы выявить причины, на рис. 3 (слева) мы сравниваем их ошибки обучения/валидации во время процедуры обучения. Мы наблюдаем проблему деградации - 34-слойная простая сеть имеет более высокую ошибку обучения на протяжении всей процедуры обучения, несмотря на то, что пространство решений 18-слойной простой сети является подпространством 34-слойной.

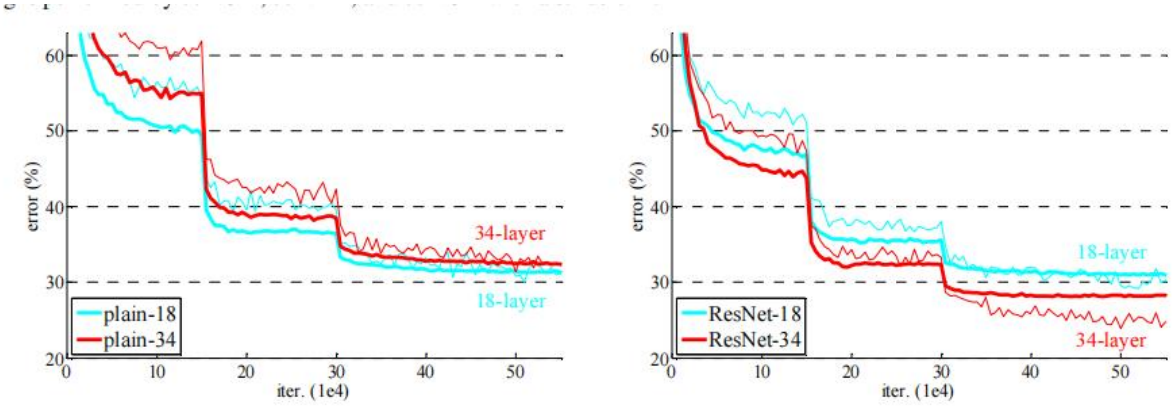


Рис 5. Обучение на ImageNet

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Таблица 3. Ошибка топ-1 (%, 10-кратное тестирование) при проверке ImageNet.

Остаточные сети.

Далее мы оцениваем 18- и 34-слойные остаточные сети (ResNets). Базовые архитектуры такие же, как и у вышеупомянутых простых сетей, за исключением того, что к каждой паре фильтров 3×3 добавляется соединение замыкания, как на рис.2 (справа). В первом сравнении (табл.2 и рис.3 справа) мы используем отображение идентичности для всех замыканий и нулевое добавление для возрастающих размеров. Таким образом, они не имеют дополнительного параметра по сравнению с обычными аналогами.

Из таблицы 2 и рис. 3 можно сделать три основных вывода. Во-первых, ситуация с остаточным обучением обратная - 34-слойная ResNet лучше 18-слойной (на 2,8%). Что более важно, 34-слойная ResNet демонстрирует значительно меньшую ошибку обучения и является обобщенной для данных валидации. Это указывает на то, что проблема деградации хорошо решена в данном случае, и нам удается получить прирост точности за счет увеличения глубины.

Во-вторых, по сравнению со своим простым аналогом, 34-слойнаяВо-вторых, по сравнению со своим простым аналогом, 34-слойная ResNet снижает ошибку top-1 на 3,5% (табл. 2), что является результатом успешного снижения ошибки обучения (рис.3 справа против слева). Это сравнение подтверждает эффективность остаточного обучения на чрезвычайно глубоких системах.

Наконец, мы также отмечаем, что 18-слойные простые/остаточные сети имеют сопоставимую точность (табл. 2), но 18-слойная ResNet сходится быстрее (рис.3 справа и слева). Когда сеть "не слишком глубокая" (здесь 18 слоев), текущий решатель SGD все еще способен находить хорошие решения для простой сети. В этом случае ResNet облегчает оптимизацию, обеспечивая более быструю сходимость на ранней стадии.

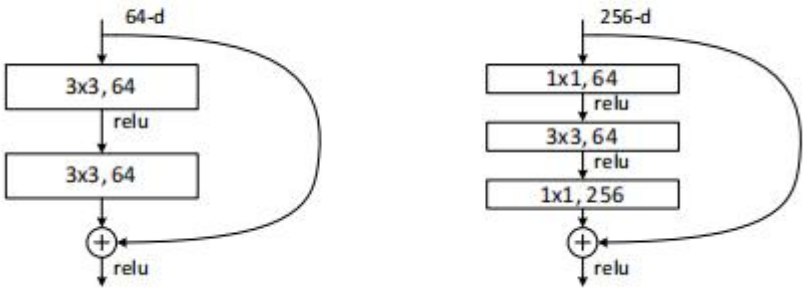


Рис 6. Более глубокая остаточная функция F для ImageNet. Слева:строительный блок (на картах признаков 56×56), как на рис. 3, для ResNet-34. Справа: блок построения "узкого места" для ResNet-50/101/152.

Далее мы описываем наши более глубокие сети для ImageNet. Из-за проблем со временем обучения, которое мы можем себе позволить, мы модифицировали строительный блок как конструкцию узкого места4 . Для каждой остаточной функции F мы используем стек из 3 слоев вместо 2 (рис. 4). Три слоя представляют собой свертки 1×1, 3×3 и 1×1, где слои 1×1 отвечают за уменьшение, а затем увеличение (восстановление) размеров, оставляя слой 3×3 узким местом с меньшими размерами входа/выхода. На рис. 4 показан пример, где обе конструкции имеют схожую временную сложность.

Тождественные ярлыки без параметров особенно важны для архитектур узких мест. Если ярлык идентичности на рис. 4 (справа) заменить проекцией, можно показать, что временная сложность и размер модели удваиваются,

поскольку ярлык связан с двумя высокоразмерными концами. Таким образом, тождественные замыкания приводят к более эффективным моделям для конструкций узких мест.

101-слойные и 152-слойные ResNets: мы создаем 101- слой и 152-слойные ResNets с использованием большего количества 3-слойных блоков (Таблица 1). Замечательно, хотя глубина значительно увеличился, 152-уровневый ResNet (11,3 миллиарда FLOPs) по-прежнему имеет меньшую сложность, чем сети VGG-16/19 (15,3/19,6 млрд FLOP).

102-50/101/152-слойные ResNet более точны, чем 34-слойные со значительным отрывом (табл. 3, 4). Мы не наблюдаем проблемы деградации и, таким образом, наслаждаемся значительным приростом точности за счет значительного увеличения глубина. Преимущества глубины засвидетельствованы для всех оценок метрики (табл. 3 и 4).

Сравнение с современными методами. В таблице 4 мы сравниваем с предыдущими лучшими результатами одной модели. Наши базовые 34-слойные сети ResNet достигли очень конкурентоспособной точности. Наш 152-уровневый ResNet имеет единую модель ошибка проверки топ-5 4,49%. Этот результат одной модели превосходит все предыдущие результаты ансамбля (таблица 5). Мы объединить шесть моделей разной глубины, чтобы сформировать ансамбль (только с двумя 152-слойными на момент подачи). Это приводит к 3,57% ошибки топ-5 в тестовом наборе (таблица 5). Эта работа заняла 1-е место на ILSVRC 2015.

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43[†]
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except [†] reported on the test set).

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

Часть II

Тема: «MobileNets: эффективные конволюционные нейронные сети для приложений мобильного зрения»

Сначала мы покажем результаты для MobileNet с глубинными сепарабельными свертками по сравнению с моделью, построенной с использованием полных сверток. В таблице 4 мы видим, что использование глубинных сепарабельных сверток по сравнению с полными свертками снижает точность только на 1% в ImageNet, при этом значительно экономя на мультидобавках и параметрах.

Далее мы покажем результаты сравнения более тонких моделей с множителем ширины с более мелкими моделями, использующими меньшее количество слоев. Чтобы сделать MobileNet более мелкой, удаляются 5 слоев разделяемых фильтров с размером признаков 14 × 14 × 512 в таблице 1. В таблице 5 показано, что при одинаковых вычислениях и количестве параметров сделать MobileNet более тонкими на 3% лучше, чем сделать их более мелкими.

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

В таблице 6 показаны компромиссы между точностью, вычислениями и размером при уменьшении архитектуры MobileNet с помощью множителя ширины. Точность плавно снижается, пока архитектура не становится слишком маленькой при $\alpha = 0{,}25$.

В таблице 7 показаны компромиссы между точностью, вычислениями и размерами для различных множителей разрешения путем обучения MobileNet с уменьшенным входным разрешением. Точность падает плавно снижается в зависимости от разрешения.

На рисунке 7 показан компромисс между точностью ImageNet и вычислениями для 16 моделей, составленных из перекрестного произведения множителя ширины $\alpha \in \{1, 0.75, 0.5, 0.25\}$ и разрешения {224, 192, 160, 128}. Результаты являются логарифмически линейными со скачком, когда модели становятся очень маленькими при $\alpha = 0{,}25$.

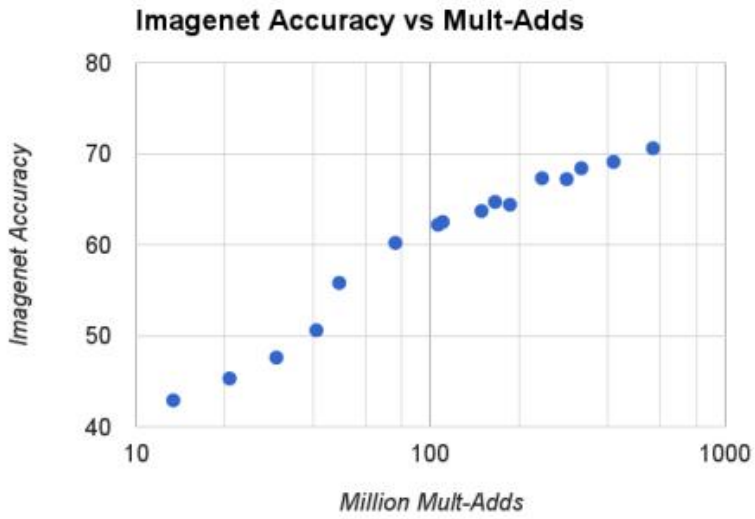


Рис 7. На этом рисунке показан компромисс между вычислениями (Mult-Adds) и точностью на эталоне ImageNet. Обратите внимание на логарифмически линейную зависимость между точностью и вычислениями.

На рисунке 5 показан компромисс между точностью ImageNet и количеством параметров для 16 моделей составленных из перекрестного произведения множителя ширины $\alpha \in \{1, 0.75, 0.5, 0.25\}$ и разрешения {224, 192, 160, 128}.

В таблице 8 приведено сравнение полного MobileNet с оригинальным GoogleNet [30] и VGG16 [27]. MobileNet почти так же точна, как VGG16, при этом она в 32 раза меньше и в 27 раз менее требовательна к вычислениям. Он более точен, чем GoogleNet при меньших размерах и более чем в 2,5 раза меньшем объеме вычислений.

В таблице 9 сравнивается уменьшенная MobileNet с множителем ширины α = 0,5 и уменьшенным разрешением 160 × 160. Уменьшенный MobileNet на 4% лучше, чем AlexNet [19], при этом он на 45× меньше и на 9,4× меньше вычислений, чем AlexNet. Она также на 4% лучше, чем Squeezenet [12] при примерно таком же размере и 22× меньше вычислений.

Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Table 9. Smaller MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.50 MobileNet-160	60.2%	76	1.32
Squeezenet	57.5%	1700	1.25
AlexNet	57.2%	720	60

Table 10. MobileNet for Stanford Dogs

Model	Top-1 Accuracy	Million Mult-Adds	Million Parameters
Inception V3 [18]	84%	5000	23.2
1.0 MobileNet-224	83.3%	569	3.3
0.75 MobileNet-224	81.9%	325	1.9
1.0 MobileNet-192	81.9%	418	3.3
0.75 MobileNet-192	80.5%	239	1.9

Table 11. Performance of PlaNet using the MobileNet architecture. Percentages are the fraction of the Im2GPS test dataset that were localized within a certain distance from the ground truth. The numbers for the original PlaNet model are based on an updated version that has an improved architecture and training dataset.

Scale	Im2GPS [7]	PlaNet [35]	PlaNet MobileNet
Continent (2500 km)	51.9%	77.6%	79.3%
Country (750 km)	35.4%	64.0%	60.3%
Region (200 km)	32.1%	51.1%	45.2%
City (25 km)	21.9%	31.7%	31.7%
Street (1 km)	2.5%	11.0%	11.4%

Мелкозернистое распознавание

Мы обучаем MobileNet для мелкозернистого распознавания на наборе данных Stanford Dogs. Мы расширяем подход и собираем еще больший, но

шумный набор данных для обучения, чем из Интернета. Мы используем зашумленные веб-данные для предварительного обучения мелкозернистой модели распознавания собак, а затем проводим точную настройку модели на обучающем множестве Stanford Dogs. Результаты на тестовом наборе Stanford Dogs приведены в таблице 10. MobileNet может почти достичь современных результатов из при значительно меньших вычислениях и размерах.

4. Заключение

Когда сеть деградирует, мелкая сеть способна достичь лучших результатов обучения, чем глубокая сеть, и в этот момент, если мы передадим признаки с нижних слоев на верхние, результаты должны быть, по крайней мере, такими же хорошими, как у мелкой сети, или если сеть VGG-100 использует точно такие же признаки на слое 98, как VGG-16 на слое 14, то VGG-100 должна иметь тот же эффект, что и VGG-16. такой же эффект. Поэтому мы можем добавить прямое отображение (Identity Mapping) между слоями 98 и 14 VGG-100 для достижения этого эффекта.

С информационно-теоретической точки зрения, из-за неравенства DPI (Data Processing Inequality) карта характеристик содержит меньше информации изображения слой за слоем по мере увеличения количества слоев при прямой передаче, а добавление прямого отображения ResNet гарантирует, что l+1 слой сети должен содержать больше информации изображения, чем l слой. Основываясь на этой идее использования прямого отображения для прямого соединения различных слоев сети, мы, таким образом, приходим к использованию остаточной сети

Мы предложили новую архитектуру модели под названием MobileNets, основанную на глубинных сепарабельных свертках. Мы исследовали некоторые важные проектные решения, ведущие к созданию эффективной модели. Затем мы продемонстрировали, как построить более компактные и быстрые

MobileNets с использованием множителя ширины и множителя разрешения, пожертвовав разумной точностью для уменьшения размера и задержки. Затем мы сравнили различные MobileNets с популярными моделями, демонстрирующими превосходные характеристики размера, скорости и точности.

5. Список использованных источников

[1] X. Glorot and Y. Bengio. Understanding the diffificulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[2] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv:1302.4389*, 2013.

[3] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *CVPR*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectififiers: Surpassing human-level performance on imagenet classifification. In
ICCV, 2015.

[6] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fifine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[7] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fifine-grained recognition. *arXiv preprint arXiv:1511.06789*, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classifification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fifine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*, 2014.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 7

[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classifification using binary convolutional neural networks. *arXiv preprint arXiv:1603.05279*, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.