Московский государственный технический университет им. Н.Э.

Баумана

Кафедра «Системы обработки информации и управления»

Домашнее Задание

по дисциплине

«Методы машинного обучения»

Выполнил:

студент группы ИУ5-23М

Юй Шанчэнь

Москва — 2022 г.

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 по MNIST**»**

1. Выбор задачи

Полное название MNIST - Mixed National Institute of Standards and Technology database, очень большая база данных рукописных чисел.

Эти наборы данных выполняют две функции: первая - предоставление большого количества данных в качестве обучающего и проверочного наборов, обеспечивая богатую выборку информации для некоторых обучающихся. Одна из функций заключается в предоставлении богатого набора данных для обучения и проверки, обеспечивая некоторых обучающихся богатым набором образцов. Еще одной особенностью является возможность формирования эталонного проекта, который является относительно универсальным в отрасли. Поскольку мы все используем один и тот же набор данных, сети, разработанные каждым из нас, можно сравнить друг с другом на этих наборах данных, чтобы проверить, чья сеть имеет более высокую скорость распознавания.

2. Теоретический этап

Часть I

Тема<<АНСАМБЛЬ ПРОСТЫХ МОДЕЛЕЙ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ЦИФР МНЕСТА>>

Набор данных MNIST для распознавания рукописных цифр (рис. 1) является одним из самых основных наборов данных, используемых для тестирования производительности нейросетевых моделей и методов обучения. Используя 60 000 изображений в качестве обучающего набора, можно легко достичь точности 97%-98% на тестовом наборе из 10 000 изображений, используя такие методы обучения, как k-nearest neighbors (KNN), random forests, support vector machines (SVM) и простые нейросетевые модели. Конволюционные нейронные сети (CNN) повышают эту точность до более чем 99% при менее чем 100 неправильно классифицированных изображениях в тестовом наборе.

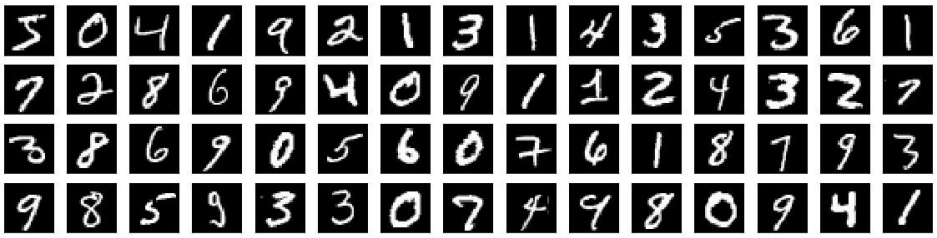

Рисунок 1: Изображения из обучающего набора MNIST.

Последние 100 изображений труднее классифицировать правильно. Чтобы повысить точность после 99%, нам нужны более сложные модели, тщательная настройка гиперпараметров, таких как скорость обучения и размер партии, методы регуляризации, такие как нормализация партии и отсев, а также увеличение объема обучающих данных. Наибольшая точность, достигнутая на тестовом наборе MNIST, составляет примерно 99,7-99,84%, как сообщается в работах [2, 3, 4, 5, 6].

В этой статье мы сообщаем о модели, которая может достичь очень высокой точности на тестовом наборе MNIST без сложных структурных аспектов или методов обучения. В модели используется набор сверточных слоев, за которыми следует полностью связанный слой в конце, что является одной из часто используемых архитектур модели. Мы используем основные схемы дополнения данных,

перевод и вращение. Мы обучаем три модели с похожими архитектурами и используем мажоритарное голосование между моделями для получения окончательного прогноза. Три модели имеют схожую архитектуру, но разные размеры ядер в слоях свертки. Эксперименты показывают, что объединение моделей с разными размерами ядра дает более высокую точность, чем объединение моделей с одинаковым размером ядра.

Проектирование и обучение сети

Наши сетевые модели состоят из нескольких слоев свертки и полностью связанного слоя в конце. В каждом слое свертки выполняется двумерная свертка, затем двумерная пакетная нормализация и активация ReLU. После свертки не используется максимальное или среднее объединение. Вместо этого размер карты признаков уменьшается после каждой свертки, так как не используется прокладка. Например, если мы используем ядро 3×3, ширина и высота изображения уменьшается на два после каждого слоя свертки. Подобный подход используется и в других сетях [6, 2]. Количество каналов увеличивается после каждого слоя, чтобы учесть уменьшение размера карты признаков. Когда размер карты признаков становится достаточно маленьким, слой с полным соединением соединяет карту признаков с конечным выходом. На полностью подключенном слое используется пакетная нормализация 1D, а отсев не используется.

Мы используем три различные сети и объединяем результаты этих сетей. Сети отличаются только размерами ядер сверточных слоев: 3×3, 5×5 и 7×7. Поскольку разный размер ядра приводит к различному уменьшению размеров карт признаков, количество слоев для каждой сети различно. Первая сеть, M3, использует 10 слоев свертки с 16(i+1) каналами в каждом слое свертки. Карта признаков становится 8×8 с 176 каналами после 10-го слоя. Вторая сеть, M5, использует 5 слоев свертки с 32i каналами в каждом слое свертки. Карта характеристик становится 8×8 со 160 каналами после 5-го слоя. Третья сеть, M7, использует 4 слоя свертки с 48i каналами в каждом слое свертки. Карта признаков становится 4×4 со 192 каналами после 4-го слоя. Структура трех сетей показана на рисунке 2.

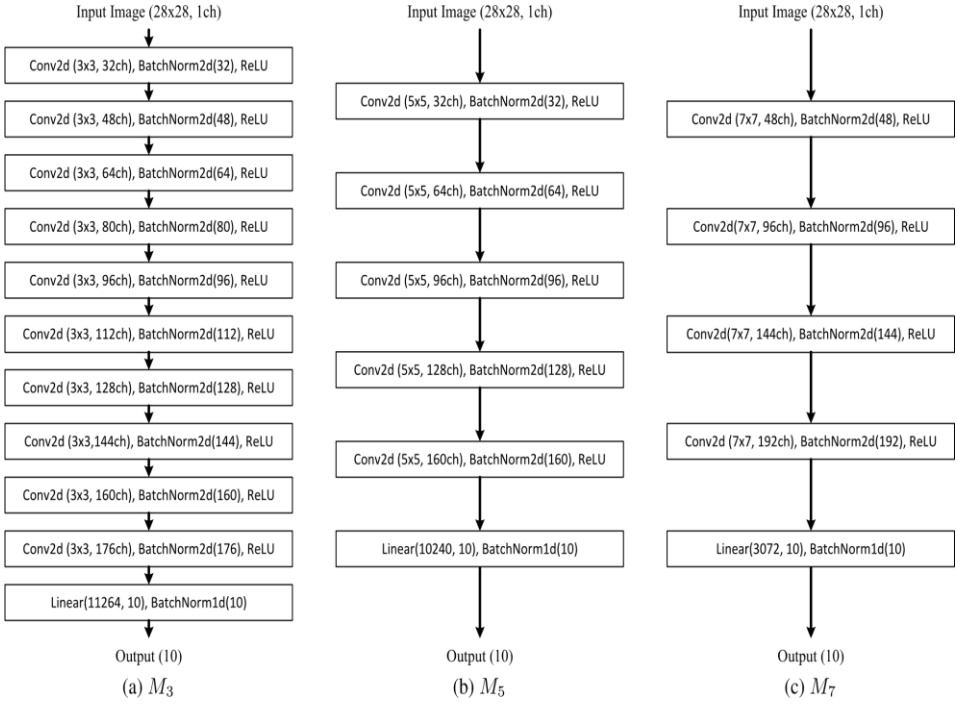

Рисунок 2: Сетевые модели, используемые для классификации цифр MNIST.

При обучении мы применяем преобразование к данным, состоящим из случайного перевода и случайного вращения. При случайном переводе изображение случайным образом сдвигается по горизонтали и вертикали на 20% от размера изображения в каждом направлении. При случайном повороте изображение поворачивается на 20 градусов по часовой или против часовой стрелки. Количество преобразований варьируется для каждого изображения и каждой эпохи, поэтому сеть получает возможность видеть различные версии изображения в обучающем наборе (рис. 3). Для обучения и оценки входные векторы, которые обычно являются целыми числами в диапазоне [0, 255], преобразуются в значения с плавающей точкой в диапазоне [-1.0, 1.0].

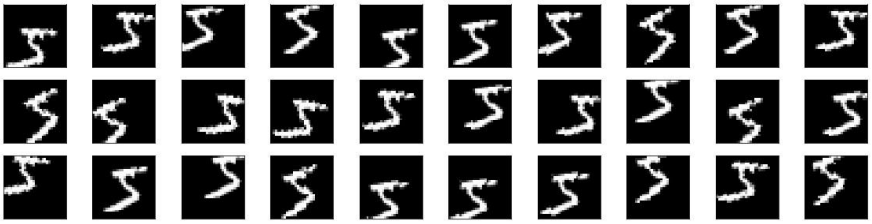

Рисунок 3: Случайный перевод и случайное вращение, примененные к обучающему изображению.

Параметры сети инициализируются с помощью методов инициализации по умолчанию в PyTorch. Для оптимизации параметров мы используем оптимизатор Адама с функцией потерь кросс-энтропии. Скорость обучения начинается с 0.001 и экспоненциально затухает с коэффициентом затухания γ=0.98. Размер

партии равен 120, поэтому за эпоху происходит 500 обновлений параметров. Для оценки мы используем экспоненциальное скользящее среднее весов, что может привести к лучшему обобщению. Экспоненциальный спад, используемый для вычисления скользящего среднего, равен 0,999.

Часть II

Тема<<Полностью конволюционные сети для семантической сегментации >>

Конволюционные нейронные сети (КНС) привлекают к себе большое внимание, поскольку они демонстрируют замечательную производительность в общих задачах распознавания объектов. До сих пор были предложены различные методы для улучшения производительности CNN: предварительная обработка, отсев, пакетная нормализация, ансамблевое обучение и так далее.

В данной работе мы предлагаем новую модель на основе CNN для дальнейшего улучшения производительности в задачах распознавания изображений. Наша модель состоит из одной базовой CNN и нескольких полносвязных подсетей (FCSN). Базовая CNN генерирует набор многоканальных карт признаков после каждого сверточного слоя. Набор признаковых карт, сгенерированных последним сверточным слоем, делится по каналам на несовпадающие подмножества, и каждое подмножество назначается одной из FCSN, которая обучается независимо от других, чтобы она могла предсказывать метку класса по подмножеству назначенных ей признаковых карт. Выход общей модели определяется большинством голосов базовой CNN и FCSN. Таким образом, в предлагаемом методе осуществляется ансамблевое обучение. Поэтому в данной работе мы называем эту модель EnsNet. Известно, что для того, чтобы обучение по ансамблю было эффективным, базовые обучаемые должны представлять определенную степень разнообразия. В предлагаемой модели ожидается, что FCSN обладают этим свойством, поскольку различные подсети обучаются на разных обучающих данных.

Далее мы сначала объясним архитектуру сети EnsNet и способ ее обучения. Затем мы приводим результаты некоторых экспериментов с использованием наборов данных MNIST, Fashion-MNIST и CIFAR-10, которые показывают, что предложенный подход, безусловно, улучшает производительность CNN. В частности, показано, что

EnsNet достигает современного уровня ошибок в 0,16% на MNIST.

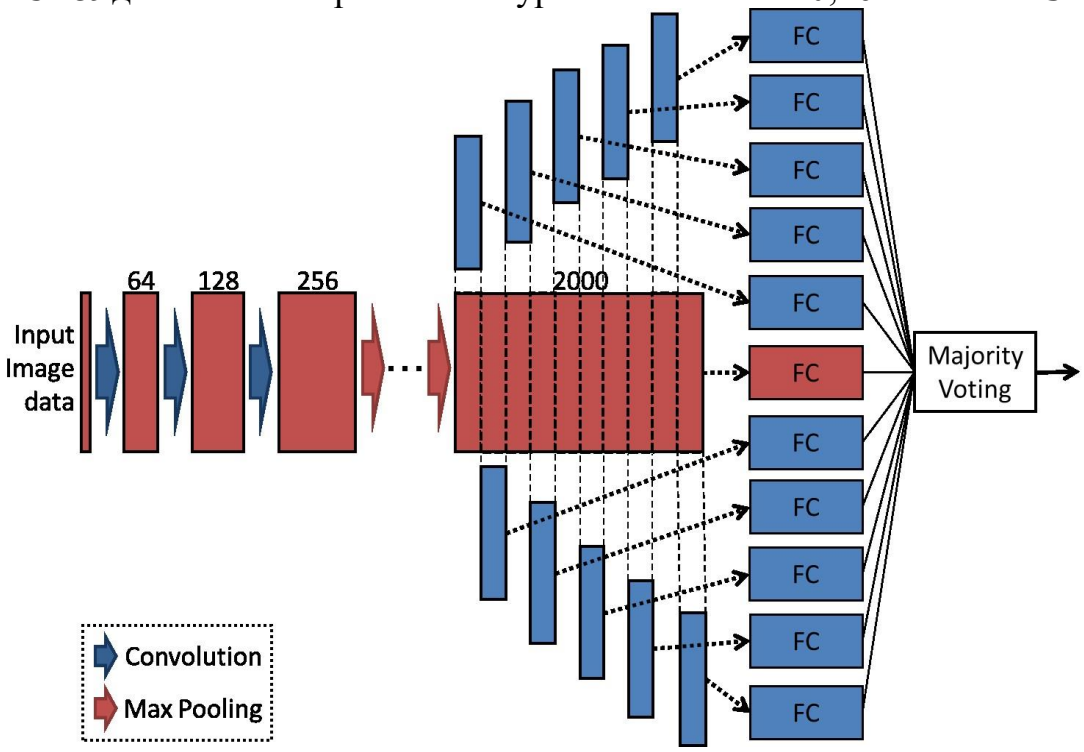

рисунок 4: пример архитектуры EnsNet.

Красные и синие ячейки представляют базовую CNN и подсети, соответственно. Целое число в верхней части каждой красной ячейки - это количество каналов карты признаков EnsNet

Предлагаемая модель под названием EnsNet состоит из одной базовой CNN и нескольких подсетей, как показано на рис. 1. Структура базовой CNN варьируется в зависимости от задач распознавания образов. В таблице 1 приведены две различные структуры, использованные в экспериментах, показанных в разделе 3: одна - для MNIST и Fashion-MNIST, а другая - для CIFAR-10. В предложенной модели используется функция активации ReLU, хотя это не показано в таблице 1. Набор карт признаков, сгенерированных последним сверточным слоем базовой CNN, делится вдоль каналов на несовпадающие подмножества, и каждое подмножество поступает в одну из подсетей. Каждая подсеть представляет собой полносвязную нейронную сеть, состоящую из нескольких весовых слоев. В таблице 2 приведены детали структуры подсетей, использованных в экспериментах: одна из них предназначена для MNIST и Fashion-MNIST, а другая - для CIFAR-10. Выход общей модели определяется большинством голосов базового CNN и подсетей.

EnsNet обучается путем чередования двух этапов: один - этап обучения базового CNN, а другой - этап обучения подсетей. На этапе обучения базовой CNN параметры сверточных слоев и полностью связанных слоев базовой CNN обновляются с помощью некоторого алгоритма оптимизации, в то время как параметры подсетей фиксированы. В экспериментах, показанных в разделе 3, используется оптимизатор Adam. На этапе обучения подсетей параметры базовой CNN фиксированы, и каждая подсеть обучается независимо от других подсетей, используя в качестве обучающих данных соответствующее подмножество карт признаков, сгенерированных последним сверточным слоем базовой CNN, и метки целевых классов. Параметры полностью связанных слоев каждой подсети обновляются тем же алгоритмом оптимизации, что и в базовой CNN.

Таблица 1: Структуры базовой CNN, использованные в экспериментах. Слева - для наборов данных MNIST и Fashion-MNIST, справа - для набора данных CIFAR-10. Обе CNN имеют девять весовых слоев. Размер каждого сверточного слоя обозначается как "Conv<размер рецептивного поля>-<число каналов>", а размер каждого полностью связанного слоя обозначается как "FC-<число узлов>". Функция активации ReLU используется в обеих моделях, но для простоты не показана в этой таблице.

Для того чтобы оценить эффективность EnsNet, мы провели эксперименты по классификации с использованием наборов данных MNIST, Fashion-MNIST и CIFAR-10. Модели, использованные в экспериментах, были реализованы в фреймворке Chainer и обучены оптимизатором Adam. Эксперименты показали, что полностью связанные подсети и мажоритарное голосование могут улучшить производительность CNN. полностью связанные подсети и мажоритарное голосование, безусловно, улучшили производительность CNN.

Input: 28 × 28 MNIST or Fashion-MNIST image
Conv3-64 (zero padding) BatchNormalization Dropout(0.35) Conv3-128 BatchNormalization Dropout(0.35) Conv3-256 (zero padding) BatchNormalization
maxpool(2 × 2)
Dropout(0.35) Conv3-512 (zero padding) BatchNormalization Dropout(0.35) Conv3-1024 BatchNormalization Dropout(0.35) Conv3-2000 (zero padding) BatchNormalization
maxpool(2 × 2)
Dropout(0.35)
Dividing feature-maps (10 divition)
FC-512 BatchNormalization Dropout(0.5)
Dropconnect(0.5) [12, 13] FC-512
FC-10
soft-max

Input: 32 × 32 CIFAR-10 image
Conv3-64 (zero padding) BatchNormalization Dropout(0.25) Conv3-128 BatchNormalization Dropout(0.25) Conv3-256 (zero padding) BatchNormalization
maxpool(2 × 2)
Dropout(0.25) Conv3-512 (zero padding) BatchNormalization Dropout(0.25) Conv3-1024 BatchNormalization Dropout(0.25) Conv3-2048 (zero padding) BatchNormalization
maxpool(2 × 2)
Dropout(0.25) Conv3-3000 (zero padding) BatchNormalization Dropout(0.25) Conv3-3500 (zero padding) BatchNormalization Dropout(0.25) Conv3-4000 (zero padding) BatchNormalization Dropout(0.25)
Dividing feature-maps (10 divition)
FC-512 BatchNormalization Dropout(0.3)
Dropconnect(0.3) FC-512
FC-10
soft-max

3. Практическая часть

Практическая часть выложена в Gitlab.

4 Заключение

Набор данных MNIST по рукописным цифрам часто используется в качестве набора данных начального уровня для обучения и тестирования нейронных сетей. В то время как достичь точности 99% на тестовом наборе довольно легко, правильная классификация последнего 1% изображений является сложной задачей. Люди испробовали множество различных сетевых моделей и методов для повышения точности тестирования, и наилучшая

точность, о которой сообщалось, достигает примерно 99,8%. В данной работе мы показали, что простая модель CNN с пакетной нормализацией и увеличением объема данных может достичь наилучшей точности. Использование ансамбля однородных и разнородных сетевых моделей может повысить производительность, до 99,91% точности теста, что является одним из самых современных показателей. Исследования с различными конфигурациями показывают, что высокая производительность достигается не за счет одной техники или архитектуры модели, а благодаря нескольким техникам, таким как нормализация партии, увеличение объема данных и методы ансамбля.

Предлагается новая модель CNN под названием EnsNet, которая состоит из одной базовой CNN и нескольких полностью связанных подсетей. В этой модели набор карт признаков, сгенерированных последним сверточным слоем базовой CNN, делится на разрозненные подмножества, и каждое подмножество подается на вход одной из подсетей. Обучение EnsNet происходит путем поочередного обновления параметров базовой CNN и подсетей, а предсказание осуществляется по большинству голосов базовой CNN и подсетей. Результаты экспериментов с использованием наборов данных MNIST, FashionMNIST и CIFAR-10 показывают, что EnsNet превосходит базовый CNN. В частности, EnsNet достигает наименьшего коэффициента ошибок среди некоторых современных моделей. Будущая работа заключается в оценке эффективности нашего подхода на других моделях CNN, таких как ResNet.

5 Список использованных источников

[1] Sanghyeon An Minjun Lee Sanglee Park Heerin Yang Jungmin So, “AN ENSEMBLE OF SIMPLE CONVOLUTIONAL NEURAL NETWORK MODELS FOR MNIST DIGIT RECOGNITION,” // Okayama University.

[2] Daiki Hirataand Norikazu Takahashi, “IENSEMBLE LEARNING IN CNN AUGMENTED WITH FULLY CONNECTED SUBNETWORKS,” 2020,// Department of Computer Science and Engineering Sogang University.