

## **Домашнее задание по дисциплине «Методы машинного обучения»**

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 репозитория 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.

Отчет по домашнему заданию должен содержать:

1. Титульный лист.
2. Постановку выбранной задачи машинного обучения, соответствующую этапу выбора задачи.
3. Теоретическую часть отчета.
4. Практическую часть отчета.
5. Выводы обучающегося по результатам выполнения теоретической и практической частей.
6. Список использованных источников.

## Теоретическую часть

Классификация изображений с помощью набора данных MNIST

### MNIST

Начнём с того, что разберёмся с датасетом MNIST, на котором мы потом будем сравнивать обычные и свёрточные сети.

MNIST - это один из классических датасетов на котором принято пробовать всевозможные подходы к классификации (и не только) изображений. Набор

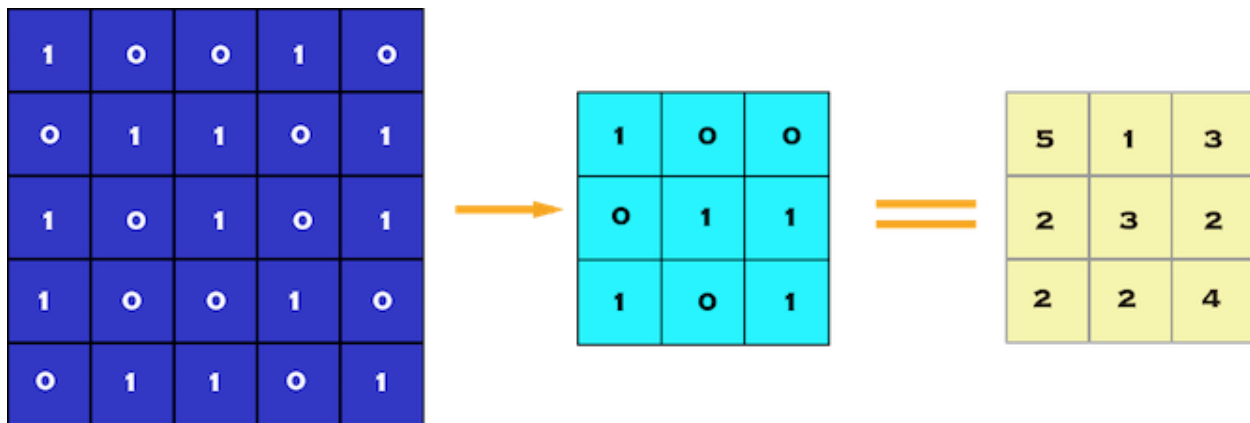
содержит 60'000 (тренировочная часть) и 10'000 (тестовая часть) черно-белых изображений размера  $28 \times 28 \times 28$  пикселей рукописных цифр от 0 до 9. В [TensorFlow](#) есть стандартный скрипт для скачивания и развёртывания этого датасета и соответственно подгрузки данных в тензоры, что весьма удобно. Обычно для тренировки выбирается 55'000 изображений, а еще 5'000 откладывается для уточнения гиперпараметров. Таким образом для тренировки мы имеем  $28 \cdot 28 \cdot 55'000 = 43'120'000$  пикселей.

Можно ознакомиться со [списком классификаторов](#) и результатами их работы на MNIST. Видно, что лучший на данный момент результат ошибается всего на 21 изображении из 10'000. Мы этого результата в данном рассказе не достигнем, поскольку нас интересует немного другое.

Главной структурной особенностью RegularNets является то, что все нейроны связаны друг с другом. Например, когда у нас есть изображения размером 28 на 28 пикселей в оттенках серого, у нас будет 784 ( $28 \times 28 \times 1$ ) нейронов в слое, который кажется управляемым. Однако у большинства изображений гораздо больше пикселей, и они не имеют оттенков серого. Следовательно, если предположить, что у нас есть набор цветных изображений в формате 4K Ultra HD, у нас будет 26 542 080 ( $4096 \times 2160 \times 3$ ) различных нейронов, связанных друг с другом в первом слое, что на самом деле неуправляемо. Следовательно, мы можем сказать, что обычные сети не масштабируются для классификации изображений. Однако, особенно когда речь идет об изображениях, корреляция или взаимосвязь между двумя отдельными пикселями незначительна, если только они не расположены близко друг к другу. Это приводит к идее сверточных слоев и объединенных слоев.

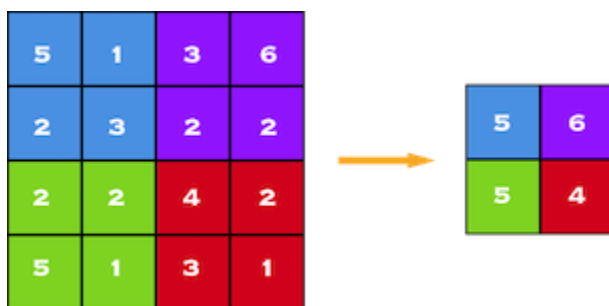
## Сверточные слои

Сверточный слой — это самый первый слой, на котором мы извлекаем признаки из изображений в наших наборах данных. Благодаря тому, что пиксели связаны только с соседними и близкими пикселями, свертка позволяет нам сохранить взаимосвязь между различными частями изображения. Свертка в основном фильтрует изображение с меньшим пиксельным фильтром, чтобы уменьшить размер изображения без потери взаимосвязи между пикселями. Когда мы применяем свертку к изображению  $5 \times 5$ , используя фильтр  $3 \times 3$  с шагом  $1 \times 1$  (сдвиг на 1 пиксель на каждом шаге). В конечном итоге мы получим вывод  $3 \times 3$  (уменьшение сложности на 64%).



### Объединение слоев

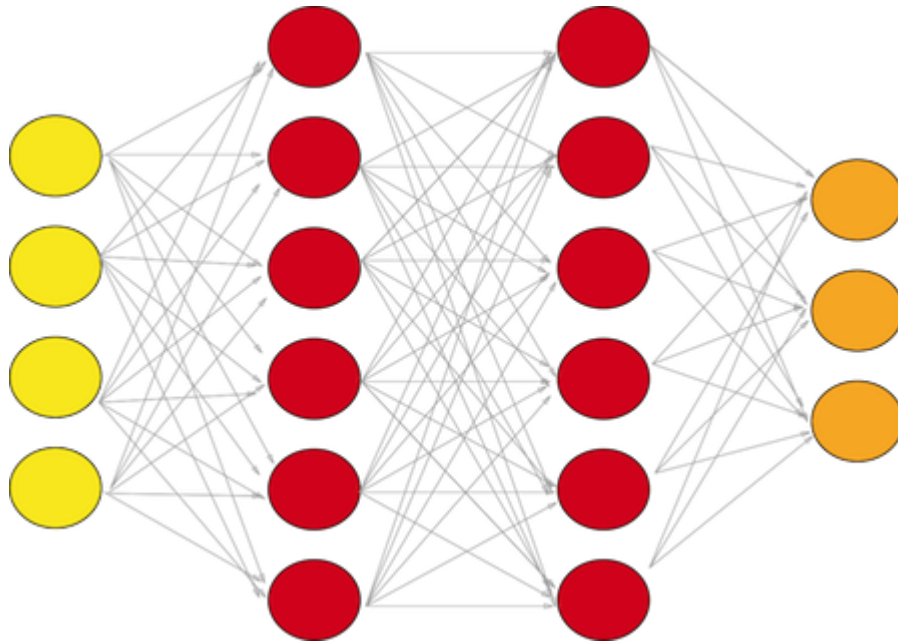
При построении CNN обычно вставляют слои объединения после каждого слоя свертки, чтобы уменьшить пространственный размер представления, чтобы уменьшить количество параметров, что снижает вычислительную сложность. Кроме того, объединение слоев также помогает решить проблему переобучения. По сути, мы выбираем размер пула, чтобы уменьшить количество параметров, выбирая максимальное, среднее или суммарное значение внутри этих пикселей. Max Pooling, один из наиболее распространенных методов объединения, можно продемонстрировать следующим образом:



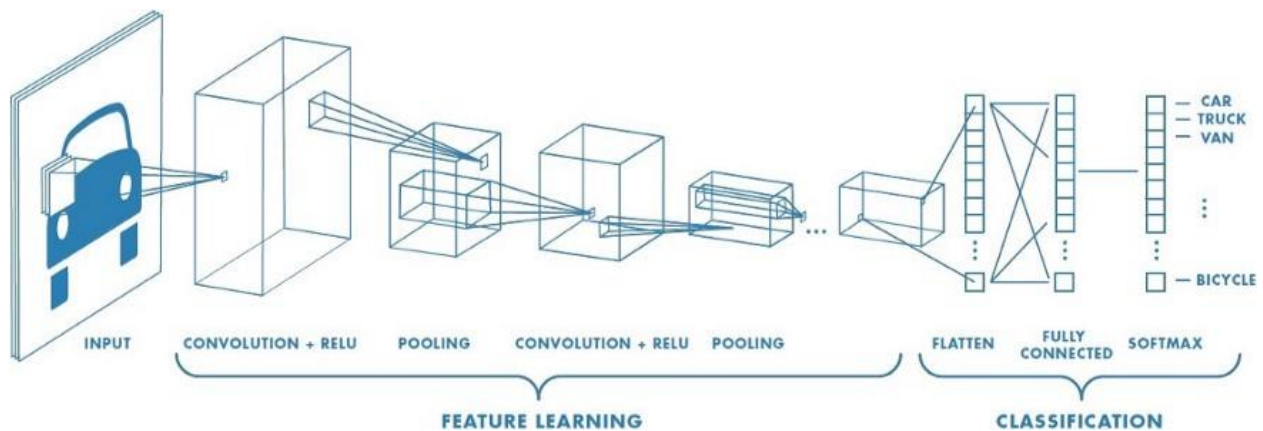
### Набор полностью связанных слоев

Полносвязная сеть — это наша регулярная сеть, в которой каждый параметр связан друг с другом, чтобы определить истинное отношение и влияние каждого параметра на метки. Поскольку наша пространственно-временная сложность значительно снижается благодаря свертке и слоям объединения, мы можем в конце концов построить полностью связанную сеть для

классификации наших изображений. Набор полносвязных слоев выглядит так:



A fully connected layer with two hidden layers



A Convolutional Neural Network Example by [Mathworks](https://www.mathworks.com/help/deeplearning/ug/convnet-architecture.html)

И теперь, когда у вас есть представление о том, как построить свёрточную нейронную сеть, которую вы можете построить для классификации изображений, мы можем получить самый банальный набор данных для классификации: набор данных MNIST, что означает модифицированную базу данных Национального института стандартов и технологий. Это большая база данных рукописных цифр, которая обычно используется для обучения различных систем обработки изображений.

3. Практическая часть выложена в Github.