

## ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

**Тема занятия:** Сверточные нейронные сети.

**Аннотация к занятию:** обучающиеся постараются решить проблему слишком большого количества параметров в сверточной нейросети. Узнаем, что такое stride. Познакомятся с операцией pooling. Узнают, как pooling помогает эффективно решить проблему большого количества параметров. Познакомятся с разными видами пулинга и рассмотрят два самых популярных: MaxPooling, AveragePooling. Познакомятся с операцией padding.

**Цель занятия:** рассмотреть несколько вариантов решения проблемы слишком большого количества параметров в сверточной нейросети. Выяснить недостатки многих подходов и ввести операцию пулинга. Сформировать представление, как пулинг помогает решить проблему большого количества параметров в сверточной нейросети.

**Задачи занятия:**

- решить проблему слишком большого количества параметров в сверточной нейросети;
- узнаем, что такое stride;
- познакомятся с операцией pooling.

## Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Здравствуйте, ребята! Рада вас приветствовать на занятии. Как ваше настроение? Готовы начинать?	
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	<p><b>Вопросы для обсуждения:</b> Давайте вспомним, что мы проходили на прошлом занятии?</p> <p><b>Возможные ответы обучающихся:</b> На прошлом занятии мы разобрали устройство сверточной нейросети и разобрались с тем, как она работает. Мы поняли, что нейросети в таком виде, как мы их построили, все еще страдают от проблемы переобучения, потому что в них получается слишком много параметров.</p> <p>На этом уроке мы поговорим о решении этой проблемы и познакомимся с операцией пулинга.</p>	

<b>Изучение нового материала</b>	50 мин.	<p>Напомню о проблеме, с которой мы столкнулись.</p> <p>Мы хотим обучить CNN классифицировать картинки размером сто на сто. И пусть в нейросети пять сверточных слоев с фильтрами размером три на три, а в последнем сверточном слое пять карт активации. Тогда размеры карт активаций последнего слоя будут девяносто на девяносто. После операции flatten мы получим вектор размером девяносто на девяносто на пять, что равно сорок тысяч пятьсот. Получается, в первом слое полносвязной части должно быть сорок тысяч пятьсот нейронов, что жутко много. Сеть, конечно, переобучится.</p> <p>Вопросы для обсуждения: Как решить эту проблему?</p> <p>Есть несколько вариантов. Ребята, ответы на объемные вопросы лучше искать, работая в группах. Сейчас мы разделимся на группы и ответим на вопрос.</p> <ul style="list-style-type: none"> <li>Первый. Сделать размеры фильтров в слоях нейросети больше. Заменяем, к примеру, во всех слоях нашей сети фильтры три на три на фильтры размером пять на пять. Тогда размеры карт активаций каждого слоя будут меньше, и в последнем, пятом сверточном слое, карты активации будут размерностью восемьдесят на восемьдесят. Так как в последнем слое пять карт активаций, то вектор после flatten будет размером</li> </ul>	<p>Дети работают в группах. Изучают материал, готовят ответы.</p> <p>Подготовка обучающихся к выполнению групповой работы: распределение по группам, ознакомление с последовательностью выполнения, текущий инструктаж, установка на сотрудничество.</p>
----------------------------------	------------	---	--

		<p>восемьдесят на восемьдесят на пять, то есть тридцать две тысячи. Намного лучше! Мы уменьшили размер вектора на восемь тысяч. Увеличив размеры фильтров, мы могли бы уменьшить размеры карт активаций и, соответственно, уменьшить размер итогового вектора.</p> <p>Однако этот способ работает плохо. Смотрите: увеличивая размер фильтра, мы увеличиваем receptive field пикселей карты активации: теперь в каждом пикселе карты активации первого слоя содержится информация о том, что происходит в куске размером пять на пять изначальной картинки. И если на картинке очень много мелких деталей, то информация о них просто потеряется при свертке такими большими фильтрами: она не сможет уместиться в один пиксель карты активации. А терять информацию не хочется: даже мелкие детали на изображении могут играть важную роль для решения поставленной задачи.</p> <p>Представьте себе, что вы стоите перед картиной, на которой огромное число деталей. Чем дальше вы от нее отходите, то есть, чем на большую часть картины одновременно вы смотрите, тем меньше отдельных мелких деталей вы можете различить. Чтобы их рассмотреть, вам нужно подойти к картине</p>	
--	--	--	--

		<p>поближе и сконцентрировать взгляд на небольшой части картины.</p> <p>Поэтому использовать фильтры больших размеров не хочется: хочется дать нейросети возможность получить из изображения всю возможную информацию. Поэтому в CNN практически всегда используются фильтры размером три на три: по сути, минимально возможный размер. Мы увидим это в следующем модуле, когда будем рассматривать популярные архитектуры CNN. Вопросы для обсуждения:</p> <p>Здесь у вас мог возникнуть вопрос: почему три на три — это минимальный возможный размер фильтра? А как же два на два или один на один?</p> <p>во-первых, фильтры размера один на один не имеют особого смысла. Смысл фильтров в том, что они как бы смотрят на локальные части картинки и в карте активации обобщают информацию о том, что находится в этой локальной части, какие паттерны там присутствуют. И каждый следующий сверточный слой смотрит на все большую часть картинки и находит все более крупные паттерны. Любой паттерн состоит из более чем одного пикселя, поэтому странно было бы фильтром размером один на один пытаться понять, что за паттерн находится в одном пикселе изображения.</p>	
--	--	--	--

		<p>Мы говорили о том, что свертку части картинки три на три фильтром три на три можно рассматривать как получение информации о том, что происходит вокруг центрального пикселя этой части три на три. Если у вас фильтр размером один на один, вы вообще не смотрите вокруг центрального пикселя. Вы смотрите только на него, и никакой дополнительной информации получить не можете.</p> <p>Ну и, опять же, пример с человеком и картиной: рассматривая картину, даже сложную, вы вряд ли рассматриваете каждый пиксель в отдельности.</p> <p>Это касается фильтров размером один на один. С фильтрами размером два на два все проще: они, конечно, имеют смысл, но обычно фильтры четного размера не используются. Вы, наверное, заметили, что, разговаривая о сверточной сети и фильтрах, мы брали в пример фильтры нечетных размеров: три на три или пять на пять. Это связано с удобством вычислений. Не будем на этом останавливаться, но имейте в виду, что в CNN обычно используются фильтры нечетных размеров.</p>	
--	--	---	--

		<p>Таким образом, три на три — это наименьший размер фильтра, который удобно и логично использовать в CNN.</p> <p>Мы только что поняли, что увеличение размера фильтров в слоях CNN — не лучшее решение этой проблемы. Это ухудшает эффективность сверточной сети. Что ж, обсудим другие варианты решения проблемы.</p> <ul style="list-style-type: none"> <li>Следующая идея — увеличить количество сверточных слоев. Вернемся к модели, у которой все фильтры во всех слоях размером три на три. Добавим к ней еще один сверточный слой, в котором фильтры также будут размером три на три. Карты активации этого слоя будут уже меньше: размером восемьдесят восемь на восемьдесят восемь. Добавим еще несколько сверточных слоев — получим карты активации еще меньшего размера. Как следствие, после flatten размер вектора тоже уменьшится.</li> </ul> <p>Эта идея, конечно, уменьшает размер вектора и, как следствие, первого слоя полносвязной части сети. Но при этом добавление каждого нового сверточного слоя увеличивает количество параметров сети. Не будем производить сложные вычисления и оценки. Скажу только, что таким образом приемлемое количество параметров нейросети получить</p>	
--	--	--	--

		<p>не выйдет: при любом количестве сверточных слоев оно будет все равно довольно большим.</p> <ul style="list-style-type: none"> <li>Идея номер три. Давайте уменьшим количество карт активации последнего сверточного слоя. Сразу понятно, что идея плохая. Мы говорили, что разные карты активации реагируют на разные по семантике паттерны. Получается, чем больше карт активации у слоя сети, тем больше разных паттернов и информации сверточная часть может выделить из картинки. А это хорошо. И жертвовать выразительностью нейросети в обмен на количество параметров мы бы не хотели.</li> <li>Идея номер четыре. Познакомимся с такой штукой, как <code>stride</code>. Идея состоит вот в чем: мы все еще будем использовать фильтры размером три на три. Но при свертке картинки или карты активации таким фильтром мы будем двигать фильтр по картинке каждый раз не на один пиксель вправо или вниз, а на два или три. На слайде вы видите анимацию свертки картинки фильтром размера три на три, когда каждый шаг фильтра по картинке происходит на два пикселя вправо или вниз. Количество шагов называется <code>stride</code>. Поэтому на слайде показана свертка картинки фильтром со страйдом два.</li> </ul>	<p>Для справки: <a href="https://python-school.ru/wiki/convolutional-neural-network/">https://python-school.ru/wiki/convolutional-neural-network/</a></p> <p>Для справки: <a href="https://computersciencewiki.org/index.php/Max-pooling/_Pooling">https://computersciencewiki.org/index.php/Max-pooling/_Pooling</a></p>
--	--	--	---



		<p>При такой свертке карта активации получится намного меньше. Действительно, мы как бы пропускаем каждый второй ход. Сворачивая картинку размером семь на семь фильтром три на три, к примеру, мы получим карту активации размером всего три на три. Это позволит сильно уменьшить размеры карт активации слоев сети. Карты активации последнего слоя будут намного меньше, и после операции flatten вектор получится меньшего размера.</p> <p>Эту технику действительно иногда используют, но и у нее есть явный недостаток. Он похож на тот, что мы обсуждали, когда пробовали увеличить размеры фильтров. Это потеря информации. Пропуская ходы фильтра по картинке, мы теряем информацию, которую фильтр мог бы выделить из картинки на пропущенных ходах. Это тоже не очень приятно.</p> <p>Итак, мы рассмотрели четыре варианта уменьшения количества параметров сверточной нейросети, и у всех них были значимые недостатки. Сейчас мы рассмотрим пятый, последний вариант. Это вариант-компромисс: он, конечно, тоже не идеален, но все же работает лучше, чем все те, что мы только что обсудили. Это подтверждается и на практике.</p> <p>Познакомимся с pooling.</p>	
--	--	--	--

		<p>Pooling — это операция уменьшения карт активации.</p> <p>Делается она очень просто. Берем карту активации. Пусть она будет размером четыре на четыре. Делим ее на квадраты размером два на два.</p> <p>Теперь из каждого квадрата два на два берем только одно число: максимальное число в этом квадрате. Получаем четыре числа.</p> <p>И теперь записываем эти числа вместо каждого квадрата два на два. Так мы из карты активации размером четыре на четыре получаем карту активации размером два на два.</p> <p>Такая операция называется MaxPooling с ядром размером два. Max — потому что мы из каждого квадратика размером два на два оставляем максимальный элемент. Ядро размером два значит, что мы делили нашу карту активации на квадратики размером два на два.</p> <p>MaxPooling с ядром размером два можно применять к любой карте активации, которую можно поделить на квадратики размером два на два. То есть, к любой карте активации, у которой размеры сторон четные. К картам активации, у которых нечетный размер, пулинг с ядром два применить нельзя: мы просто не сможем поделить такую карту на квадратики размером два на два.</p>	
--	--	---	--

		<p>На этом слайде пример применения MaxPooling с ядром два к карте активации размером шесть на шесть. Ее мы так же делим на кусочки размером два на два и из каждого кусочка берем максимальный элемент. На выходе получаем карту активации размером три на три. Как видим, пулинг с ядром размером два на два уменьшает размер карты активации в два раза.</p> <p>Пулинг можно делать и с другим размером ядра. Например, три на три. Такой пулинг можно применить к любой карте активации, сторона которой делится на три. На слайде вы видите пример применения MaxPooling'a с ядром три к той же карте активации размером шесть на шесть. Пулинг ядром размером три уменьшает размер карты активации уже в три раза.</p> <p>В целом, MaxPooling с ядром размером <math>n</math> делит карту активации на квадраты размером <math>n</math> на <math>n</math> и оставляет из каждого максимальный элемент. Таким образом он уменьшает размер карты активации в <math>n</math> раз. И может применяться только к тем картам активации, размер которых делится на <math>n</math>.</p> <p>Однако на практике практически всегда применяется пулинг с ядром размером два на два.</p> <p>Мы рассмотрели несколько вариантов решения проблемы слишком большого количества</p>	
--	--	---	--

		<p>параметров в сверточной нейросети. Выяснили недостатки многих подходов и ввели операцию пулинга. В этом видео вы узнаете, как пулинг помогает решить нашу проблему.</p> <p>Вернемся к нейросети с пятью сверточными слоями и фильтрами размером три на три. Она принимала на вход картинку размером сто на сто и в последнем сверточном слое у карт активации был размер девяносто на девяносто, а размер вектора после операции flatten был сорок тысяч пятьсот.</p> <p>Добавим в эту нейросеть операцию пулинга после второго и четвертого сверточных слоев. После операции пулинга карты активации второго сверточного слоя уменьшатся в два раза: станут размером сорок восемь на сорок восемь. Третий сверточный слой будет принимать на вход уже эти, уменьшенные карты активации. Получается, после третьего слоя мы получим карты размером сорок шесть на сорок шесть, а после четвертого — сорок четыре на сорок четыре. После четвертого слоя у нас опять стоит операция пулинга, поэтому карты активации снова уменьшатся вдвое: станут размером двадцать два на двадцать два. После пятого, последнего слоя сверток, итоговые карты активации будут размером двадцать на двадцать.</p>	
--	--	---	--

		<p>Получается, после операции flatten итоговый вектор будет иметь размер двадцать на двадцать на пять, что равно две тысячи. Смотрите: добавив две операции пулинга в сеть, мы сократили размер вектора в двадцать раз. Две тысячи — совсем не много, и такая сеть, скорее всего, переобучаться не будет. Вот так простая идея помогает уменьшить количество параметров!</p> <p>Заметим, что у пулинга нет обучаемых параметров: это просто функция, которая берет на вход карту активации и уменьшает ее в два раза определенным образом. Поэтому пулинг не делает количество обучаемых параметров сети больше.</p> <p>Операцию пулинга можно добавлять после любых сверточных слоев сети. Ну, если размеры карт активаций позволяют, конечно. Хорошо, мы поняли, как работает пулинг и как использовать его в сверточной сети. Скажу пару слов о том, почему использование пулинга, в отличие от наших прошлых идей, не так сильно ухудшает работу сверточной сети.</p> <p>Посмотрим еще раз на то, как работает MaxPooling. Пусть ядро будет размера два. Он делит карту активации на кусочки два на два и из каждого кусочка берет максимальный элемент.</p>	<p>Для справки: Проследив принцип работы биологического мозга, учёные смогли разработать математический аппарат для извлечения свойств. Но оценив общее количество слоёв и свойств, которые нужно проанализировать для отслеживания сложных геометрических форм, учёные поняли, что для хранения всех данных компьютерам не хватит памяти. Более того, количество необходимых вычислительных ресурсов растёт экспоненциально</p>
--	--	--	--

		<p>Вспомним, что большие числа на карте активации — это места, где фильтр активировался сильнее всего. То есть, большие числа на карте активации отражают наличие на исходном изображении явно выраженных паттернов. И чем число больше, тем больше информации оно несет о том, что на картинке есть некоторый паттерн.</p> <p>Получается, MaxPooling, выбирая из каждого кусочка максимальный элемент, выделяет самую важную информацию из каждого кусочка карты активации. Но размер самой карты при этом уменьшается в два раза.</p> <p>Таким образом, MaxPooling — это такой метод сжатия карт активаций: он сохраняет большую часть важной информации в картах активации, но при этом помогает сильно уменьшить количество параметров модели. Это и делает его таким эффективным. Как мы увидим в следующем модуле, во всех современных архитектурах нейросетей используется пулинг.</p> <p>В завершение разговора про пулинг упомяну, что пулинг бывает разный. Мы разбирали MaxPooling, когда из каждого кусочка карты активации выбирается максимальный элемент. Но можно выбирать не максимальный элемент, а, например, брать среднее всех элементов. Такой пулинг называется AveragePooling, от слова Average — среднее. Результат применения</p>	<p>вместе с ростом количества свойств. Для решения этой проблемы была разработана методика пулинга (pooling). Её идея очень проста: если некая область содержит ярко выраженные свойства, то мы можем отказаться от поиска других свойств в этой области.</p>
--	--	---	---

		<p>AveragePooling с ядром размера два к карте активации размером четыре на четыре вы видите на слайде. Вы можете остановить видео и убедиться, что все числа в полученной уменьшенной карте активации — это средние чисел в соответствующих квадратах два на два.</p> <p>Кроме MaxPooling и AveragePooling есть еще несколько вариантов пулинга, но эти два используют чаще всего.</p> <p>Итак, мы разобрали операцию пулинга — идею, которая помогает уменьшить число параметров сверточной нейросети, при этом не сильно ухудшая ее качество работы. Прежде чем мы перейдем к практике, упомяну о еще одной полезной вещи — padding.</p> <p>Посмотрим на нейросеть для классификации картинок. На слайде вы видите ее схему без пулинга. Несколько слайдов назад мы добавили в нее пулинг после второго и четвертого сверточных слоев. Теперь я хочу добавить пулинг с ядром размера два после второго и третьего сверточных слоев.</p> <p>После первого слоя сверток карты активации будут размером девятнадцать на девятнадцать. Добавим пулинг, получим на выходе карты размером сорок девять на сорок девять. Далее — второй сверточный слой. После него</p>	
--	--	--	--

		<p>карты активации станут размером сорок семь на сорок семь.</p> <p>Видно, что я не могу далее добавить пулинг с ядром размера два: сорок семь на два не делится. Более того, я не могу добавить дальше вообще никакой пулинг, потому что 47 — это простое число, не делится вообще ни на что. Что же делать?</p> <p><b>Решение простое:</b> использовать padding.</p> <p>Padding — это добавление к карте активации рамки из нулей. На слайде вы видите пример добавления паддинга размером один к карте активации размером четыре на четыре. Добавляем рамку из нулей толщиной один, но получаем, что длина и ширина карты активации увеличивается на два: получилось шесть на шесть.</p> <p><b>Вопросы для обсуждения:</b> Как это нам поможет?</p> <p>Вернемся к нашей нейросети. Добавим к исходной картинке котика перед подачей в сеть паддинг размером один. Да, к картинке паддинг тоже можно добавлять. Картинка станет размером сто два на сто два. Тогда после первого слоя сверток фильтрами размером три на три получим карты активации размером сто на сто, после первого пулинга — пятьдесят на</p>	
--	--	---	--



		<p>пятьдесят, после следующего слоя сверток — сорок восемь на сорок восемь. Здесь мы можем применить еще один пулинг: сорок восемь — это четное число.</p> <p>Таким образом паддинг помог нам чуть изменить размеры карт активации, чтобы применить пулинг после третьего слоя стало возможно.</p> <p>Очень часто при построении сверточных сетей делают так: задают изначальный размер изображения, равный степени двойки. Например, шестьдесят четыре на шестьдесят четыре. Далее идут несколько слоев сверток с фильтрами размером три на три. В каждом слое сверток добавляется паддинг размером один. Из-за этого получается, что карты активации на выходе каждого слоя будут размером шестьдесят четыре на шестьдесят четыре. Слоев при это может быть сколько угодно.</p> <p>Далее применяется pooling с ядром размером два. Карты активации становятся размером тридцать два на тридцать два. И далее снова несколько слоев сверток с фильтрами три на три с паддингом размером один. Все карты активации в этой части сети будут одинакового размера тридцать два на тридцать два.</p> <p>Можно так продолжать и далее: снова добавить пулинг, уменьшить карты активации и добавить еще слоев сверток.</p>	
--	--	--	--

		<p>Понятно, что изначально картинку брали размером степени двойки, чтобы каждый раз при очередном пулинге размер карты активации все еще был четным.</p> <p>Ну и последнее про паддинг. У него есть второй смысл, кроме манипулирования размерами карт активации. Рассмотрим свертку картинку размером шесть на шесть фильтром размером три на три. Наложим фильтр на левый верхний угол картинку. Мы говорили о том, что свертку этой части картинку фильтром можно рассматривать как получение информации о том, что происходит на картинке вокруг среднего пикселя, который выделен красным.</p> <p>Проходясь по картинке фильтром, мы получаем информацию о том, что происходит вокруг всех внутренних пикселей изображения. На слайде они выделены красной рамкой. А что происходит вокруг пикселей границы, мы не знаем: они никогда не бывают центральными пикселями при свертке.</p> <p>Это объясняет, почему карта активации после свертки становится меньшего размера, чем начальная картинка. При свертке этой картинку шесть на шесть фильтром размером три на три мы получим карту активации четыре на четыре. Можно сказать, что карта активации содержит информацию о паттернах, которые присутствуют на внутренней части нашего изображения</p>	
--	--	---	--

		<p>размером четыре на четыре. Каждый следующий слой свертки уменьшает размеры карт активации и еще немного теряет информацию с границ изображения.</p> <p>Если мы применим padding к картинке, получим вот что. Теперь при свертке каждый пиксель исходной картинки будет в какой-то момент в центре фильтра. Мы получим карту активации такого же размера, как и начальная картинка. На ней будет вся информация о том, что происходит вокруг каждого из пикселей изначального изображения.</p> <p>Кстати, когда мы применяем к картам активации пулинг, информация со всех частей картинки сохраняется, хоть размер карты активации и уменьшается в два раза. Просто теперь каждый пиксель карты активации содержит информацию о том, что происходит в окрестности четырех пикселей изначальной картинки. Да, часть информации теряется, но большая часть все же остается. Другими словами, receptive field пикселей карты активации увеличивается. Вы можете самостоятельно подумать, как меняется receptive field каждого пикселя карты активации при пулинге.</p>	
--	--	--	--

### Подведем итоги.

Мы начали с того, что захотели решить проблему переобучения сверточных нейросетей. Проблема состояла в том, что в сверточной нейросети, а именно в ее полносвязной части, получалось слишком много параметров.

Мы перебрали несколько вариантов решения этой проблемы: увеличение количества сверточных слоев сети, увеличение размеров фильтров в слоях, уменьшение количества карт активаций. Также попробовали увеличить страйд: двигаться фильтром не на один пиксель вправо или вниз, а на два или больше. Но все это было не очень эффективно.

Тогда мы перешли к последнему варианту — пулингу. В следующем видео мы поймем, как пулинг поможет решить проблему слишком большого количества параметров в CNN.

Мы поняли, как применять пулинг для решения проблемы большого количества параметров. Поговорили о том, почему пулинг — эффективное решение проблемы. Узнали, что бывают разные виды пулинга, но самые популярные из них — MaxPooling и AveragePooling.

Наконец, мы познакомились с операцией паддинга. Узнали, что с ее помощью можно изменять размеры карт активации в слоях

		<p>нейросети, что бывает очень удобно. Поговорили о ее втором смысле: сохранении информации с границ изображения.</p> <p>На этом теоретические видео о сверточных нейросетях подошли к концу. Самое время заняться практикой: построить и обучить свою сверточную нейросеть.</p>	
<b>Закрепление изученного материала</b>	15 мин.	<p><b>Вопросы для обсуждения:</b></p> <ul style="list-style-type: none"> <li>• Как можно решить проблему слишком большого количества параметров в сверточной нейросети?</li> <li>• Что такое stride?</li> <li>• Расскажите об операции pooling.</li> <li>• Какие виды пулинга мы рассмотрели (MaxPooling, AveragePooling);</li> <li>• Расскажите об операции padding</li> </ul>	Педагог организует беседу по вопросам.
<b>Этап подведения итогов занятия (рефлексия)</b>	8 мин.	<p><b>Вопросы для обсуждения:</b></p> <ul style="list-style-type: none"> <li>• Чему я научился?</li> <li>• С какими трудностями я столкнулся?</li> <li>• Каких знаний мне не хватает для более глубокого понимания изученного материала?</li> <li>• Достиг ли я поставленных целей и задач?</li> </ul>	Педагог способствует размышлению обучающихся над вопросами.

Информация о домашнем задании, инструктаж по его применению	5 мин.	-	
---	--------	---	--

### Рекомендуемые ресурсы для дополнительного изучения:

1. Что такое нейронные сети. [Электронный ресурс] – Режим доступа: <https://python-school.ru/wiki/convolutional-neural-network/> ;
2. Как работает сверточная нейронная сеть. [Электронный ресурс] – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/> ;
3. Свёрточные нейросети. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/nix/blog/443236/> ;
4. Pooling. MaxPooling [Электронный ресурс] – Режим доступа <https://computersciencewiki.org/index.php/Max-pooling> / [Pooling](https://computersciencewiki.org/index.php/Pooling)