

ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Свёрточные нейросети

Аннотация к занятию: обучающиеся познакомятся с устройством свёрточной нейросети с несколькими свёрточными слоями. Узнают, что такое операция свёртки и в чём её смысл, а также построят прототип нейросети на основе свёрток.

Цель занятия: сформировать у обучающихся представление об устройстве свёрточной нейросети. Построить свёрточную нейросеть с несколькими свёрточными слоями. Показать, как слои свёрток помогают модели выделять из картинок более сложные, высокоуровневые паттерны.

Задачи занятия:

- познакомить обучающихся с устройством свёрточной нейросети с несколькими свёрточными слоями;
- сформировать понятие «фильтр»;
- изучить операцию свёртки;
- построить прототип свёрточной нейросети.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Добрый день! Вы уже знаете, как свёрточные нейросети понимают, что изображено на картинке.	Приветствие. Создание в классе атмосферы психологического комфорта
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	<p>Как вы думаете, что такое операция свёртки и какой у неё смысл?</p> <p>Возможные ответы обучающихся:</p> <ul style="list-style-type: none"> • алгоритм; • обработка информации; • не знаю, что такое свёртка. <p>Какая сегодня тема урока?</p> <p>Возможный ответ обучающихся: свёрточная нейросеть.</p> <p>Да, тема урока — «Свёрточная нейросеть».</p> <p>Как вы думаете, какие задачи мы сможем сегодня решить?</p> <p>Возможные ответы обучающихся:</p> <ul style="list-style-type: none"> • узнаем, что такое свёрточная нейросеть; • изучим операцию свёртки; 	Способствовать обсуждению мотивационных вопросов

		<ul style="list-style-type: none"> • построим прототип свёрточной нейросети. 	
<p>Изучение нового материала</p>	<p>45 мин.</p>	<p>Операция свёртки состоит в сворачивании изображения с помощью фильтра.</p> <p>Что же такое фильтр?</p> <p>Возможные ответы обучающихся:</p> <ul style="list-style-type: none"> • устройство для очистки; • устройство для удаления чего-нибудь. <p>Фильтр — это тоже матрица чисел. У него есть и другое название — ядро. Давайте рассмотрим процесс свёртки изображения фильтром на примере.</p> <p>Возьмем чёрно-белое изображение размером 4x4 и двумерный фильтр размером 3x3. Вы видите их на слайде. Значок между ними означает операцию свёртки.</p> <p>Наложим фильтр на левый верхний угол изображения. Каждый пиксель фильтра будет соответствовать одному пикселю на изображении.</p> <p>Теперь произведём свёртку этой части изображения. Для этого поэлементно перемножим числа фильтра на соответствующие им числа на картинке и сложим полученные произведения. В результате получится одно число — двадцать девять.</p>	<p>Для справки: Сайт: https://habr.com/ru/post/309508/ Перед уроком рекомендуется ознакомиться с материалами, представленными на сайте.</p>

Это число будет первым элементом матрицы, которая получится в результате свёртки всего изображения с помощью нашего фильтра.
На этом слайде вы видите небольшой спойлер: результатом свёртки всей картинки в нашем случае будет матрица размером два на два. И мы только что вычислили её первый элемент.

После вычисления первого значения двигаемся дальше: двигаем фильтр по изображению на 1 пиксель вправо. Теперь фильтр покрывает другие 9 пикселей изображения.
Здесь мы делаем то же самое: перемножаем поэлементно элементы фильтра на элементы изображения, которые фильтр покрывает. Получаем новое значение итоговой матрицы: -192 . Это значение записываем в итоговой матрице справа от предыдущего значения 29 . Справа, потому что мы сдвинули фильтр вправо по изображению.

Двигаем фильтр по изображению дальше. Вправо двигаться уже некуда, поэтому возвращаемся к началу строки и смещаем фильтр на 1 пиксель вниз. Продолжаем в этом положении фильтра ту же операцию свёртки и получаем третье значение: -35 . Это значение мы записываем в итоговую матрицу в начало второй строки, так как мы сдвинули фильтр на 1 пиксель вниз на изображении.

Последний шаг. Сдвигаем фильтр вправо на 1 пиксель, выполняем свёртку, получаем последнее значение: -22 . Записываем его в правую нижнюю ячейку итоговой матрицы.

Всё! Мы свернули наше изображение размером 4×4 с помощью фильтра размером 3×3 и получили итоговую матрицу размером 2×2 . Эта итоговая матрица называется картой активации. Почему она так называется, мы узнаем далее.

На слайде вы видите анимацию всего процесса свёртки, который мы только что проделали. Аналогично можно сворачивать изображения любого размера фильтрами любого размера. Нужно только, чтобы размер изображения был не меньше, чем размер фильтра. Иначе мы просто не сможем наложить фильтр на картинку так, чтобы все числа фильтра лежали на каком-то из пикселей картинки. Заметим, что размер карты активации не равен размеру исходного изображения: он стал меньше. Это произошло, потому что мы сворачивали картинку фильтром размером три на три, и из каждого кусочка размером три на три изображения получали лишь один пиксель карты активации.

Точно так же, если бы мы сворачивали фильтром три на три картинку из MNIST размером двадцать восемь на двадцать восемь, то получили бы карту активации размером двадцать шесть на двадцать шесть: на два меньше по высоте и по ширине. А если бы сворачивали её фильтром размером пять на пять, то получили бы карту активации размером ещё меньше: двадцать четыре на двадцать четыре.

На этом слайде вы видите общую формулу расчёта размера карты активации по размеру исходного

изображения и размеру фильтра. Здесь m — размер карты активации, i — размер изображения, f — размер фильтра. Вы можете убедиться, что формула верна: попробовать самостоятельно посчитать карты активации для картинок и фильтров разных размеров.

Заметим здесь вот что: операция свёртки удовлетворяет принципам локальности инвариантности относительно переноса, о которых мы говорили ранее. Смотрите: каждый пиксель карты активации получается в результате воздействия фильтра на небольшую локальную область изображения. Поэтому выполняется принцип локальности. Также выполняется и принцип инвариантности к переносу. Все части изображения обрабатываются одним и тем же фильтром. То есть, результаты свёрток одного и того же объекта в разных частях картинки будут одинаковыми.

Итак, мы научились сворачивать чёрно-белое изображение с помощью фильтров и получать карты активации. Прежде чем мы перейдём к обсуждению того, какой у свёртки смысл, давайте научимся сворачивать и цветные картинки.

Цветная картинка — это уже трёхмерная матрица. Её третья размерность равна трём, по числу цветовых каналов. Такую картинку можно свернуть трёхмерным фильтром, третья размерность которого также равна трём. Например, фильтром размером три на три на три или пять на пять на три. Полученная карта активации при этом будет двумерной.

Второй вариант: представить RGB-картинку как три чёрно-белые картинку: по одной на каждый цветовой канал. Свернуть каждую двумерную картинку фильтром отдельно. Так мы получим три двумерные карты активации.

Чаще всего для свёртки цветных картинок используется первый вариант: трёхмерные фильтры. Далее в этом модуле при построении свёрточной нейросети мы будем пользоваться трёхмерными фильтрами.

Отлично, теперь мы умеем сворачивать фильтрами любые картинки: чёрно-белые и цветные. Давайте теперь поймём, какой в свёртке смысл.

Рассмотрим свёртку картинки с помощью фильтра, который вы видите на слайде. Этот фильтр известен как вертикальный фильтр Собеля. Справа вы видите результат свёртки картинки домика этим фильтром. Видно, что на полученной карте активации подсвечены места, в которых на исходном изображении находились вертикальные линии. «Подсвечены» означает, что на карте активации в этих местах находятся большие по модулю числа: чем больше значение пикселя, тем ярче, белее цвет этого пикселя. Получается, вертикальный фильтр Собеля выделяет из изображения информацию, в каких местах изображения присутствуют вертикальные линии, и подсвечивает эти места на карте активации.

А вот фильтр, который называется горизонтальным фильтром Собеля. Видно, что этот фильтр на карте

активации подсвечивает места, где на изначальном изображении находились горизонтальные линии. Точно так же есть фильтры, которые реагируют на другие виды паттернов: диагональные линии, кривые и т.д. Становится понятно, почему карта активации так называется. На ней «активированы» места, где на изначальном изображении присутствует паттерн, который «ищет» фильтр. Фильтр, проходя по изображению, «активируется» в местах, в которых на изображении есть определённые паттерны, и в этих местах на карте активации появляются большие числа.

Собственно, общая идея работы свёрточной нейросети для задачи классификации такая: с помощью фильтров выделить из изображения информацию о паттернах, которые присутствуют на изображении, и передать эту информацию на вход полносвязной нейросети. По этой информации она поймёт, что за объект изображён на картинке.

Давайте построим прототип свёрточной нейросети. Пусть у нас есть задача классификации картинок четвёрок и восьмёрок.

Какие фильтры помогли бы нам отличить изображения четвёрок от изображений восьмёрок?

Ответы обучающихся

Как мы обсуждали выше, у четвёрок все линии прямые: горизонтальные и вертикальные, а у восьмёрок — плавные, расположенные под углом. Нам отлично подошли бы фильтры, которые реагируют на наличие

вертикальных линий и линий под углом сорок пять градусов. На слайде справа вы видите эти два фильтра.

Тогда наша модель для классификации изображений четвёрок и восьмёрок может быть устроена следующим образом.

Модель состоит из одного слоя свёрток, в котором два фильтра, и полносвязной части. Вот как эта модель обрабатывает входящую картинку.

Сначала картинка сворачивается с помощью двух фильтров — получаются две карты активации.

Затем каждая карта активации растягивается в вектор: точно так же, как мы в начале статьи растягивали картинку в вектор для подачи на вход полносвязной сети. Эта операция «растягивания» называется *flatten*. С английского переводится как «сделать плоским».

Далее полученные два вектора конкатенируются, склеиваются, и получается один вектор в два раза большей длины.

Наконец, этот вектор подаётся на вход обычной полносвязной сети, которая выдаёт ответ на задачу классификации: к какому классу принадлежит картинка.

В этом всём заложен следующий смысл: два фильтра выделяют из картинки информацию о том, есть ли на входящей картинке вертикальные и диагональные линии. На первой карте активации будут присутствовать большие числа, если на картинке есть вертикальные

линии. На второй карте будут большие числа, если на входящей картинке есть диагональные линии. Получается, если мы подадим на вход картинку четвёрки, то на первой карте активации будут большие числа, а на второй — нет. А если подадим на вход картинку восьмёрки, то будет наоборот: второй фильтр «активирует» карту активации, а первый — нет. В итоге после операции flatten и конкатенации полученные векторы для картинки четверки и картинки восьмерки будут отличаться: в векторе, соответствующем четверке, будут большие числа наверху и маленькие внизу, а в векторе восьмёрки наоборот. Вектор мы потом подаём на вход полносвязной сети, которая по вектору должна выдать ответ: к какому классу принадлежит входная картинка. Эту полносвязную сеть мы легко можем обучить с помощью обычного backpropagation. Таким образом, вся наша модель отлично обучится классифицировать картинки четвёрок и восьмёрок.

На самом деле для задачи классификации четвёрок и восьмёрок достаточно даже одного фильтра. Например, того, что реагирует на наличие вертикальных линий. Карты активации для картинок четвёрок и восьмёрок после такого фильтра будут сильно отличаться, и полносвязная сеть отлично обучится их классифицировать. На примере модели с двумя фильтрами видно, что в одном слое свёрточной нейросети могут быть несколько фильтров одновременно. А ещё это иллюстрирует операцию конкатенации векторов после flatten.

Подумайте самостоятельно, какие фильтры были бы полезны для классификации других чисел датасета MNIST.

Заметим, что фильтры в такой модели можно рассматривать как выделяторы признаков из картинок. А полносвязную сеть — как модель-классификатор, которая берёт на вход выделенные признаки и решает задачу классификации. Это напоминает то, как задача классификации картинок решалась до эры нейросетей. Мы говорили об этом в первом видео модуля. Там выделятором признаков был HOG, а классификатором — логистическая регрессия.

Выходит, нейросети не так далеко отошли от классических методов: идеи работы всё же похожи.

В этом виде в модели фильтры фиксированы, задаются заранее, а обучается только полносвязная часть. Однако в настоящей свёрточной нейросети фильтры также обучаются.

Обучение фильтров свёрточных слоёв имеет смысл. При построении свёрточной нейросети не всегда можно сразу понять, какие фильтры подойдут для решения конкретной задачи. В случае с классификацией четвёрок и восьмёрок было понятно, какие фильтры хорошо подойдут для выделения нужных признаков, потому что мы явно видели, чем отличаются формы четвёрок и восьмёрок: у одной цифры линии преимущественно вертикальные, а у второй — расположенные под углом сорок пять градусов.

Для справки:
Проект ImageNet — это большая визуальная база данных, предназначенная для использования в исследованиях программного обеспечения для распознавания визуальных объектов

Допустим, что теперь вы решаете задачу классификации кошек и собак. Можете ли вы сказать, какие фильтры нужно выбрать, чтобы хорошо обучить нейросеть классифицировать животных?

Ответы обучающихся

Кажется, это довольно трудно.

А теперь представьте, что мы решаем задачу классификации картинок ImageNet на тысячу классов. Среди классов есть животные, чашки, машины, самолёты, флаги и т.д. На слайде примеры картинок из ImageNet. Как много сложных форм! Для классификации всех этих картинок подобрать удачные фильтры ещё сложнее. Поэтому гораздо эффективнее дать нейросети в процессе обучения самой понять, какие фильтры лучше всего подходят для решения той или иной задачи. То есть сделать веса фильтров обучаемыми параметрами. Веса фильтров — это все числа матрицы всех фильтров.

Таким образом, обучаемыми параметрами свёрточной нейросети являются все веса всех фильтров и веса полносвязной части сети.

Вы могли подумать: а как обучать фильтры? Мы знаем, как обучается полносвязная сеть: с помощью алгоритма обратного распространения ошибки. А фильтры как? На самом деле, всё очень просто. Фильтры тоже обучаются с помощью алгоритма обратного распространения ошибки. Для обучения весов свёрточных слоёв ничего дополнительно изобретать не нужно. Таким образом, вся наша свёрточная нейросеть

обучается как одно целое с помощью алгоритма backpropagation. На практическом занятии мы в этом убедимся: вы увидите, что построить и обучить свёрточную нейросеть так же просто, как и полносвязную. Нужно просто задать её слои и вызвать тот же самый код обучения сети.

Почему вообще фильтры можно обучать алгоритмом обратного распространения ошибки? Дело в том, что операцию свёртки можно представить в виде перемножения двух матриц: матрицы изображения и некоторым образом модифицированной матрицы фильтра. Формула вычисления результата свёрточного слоя почти такая же, как и формула вычисления результата полносвязного слоя. Только матрицы в этой формуле чуть другие. Тогда понятно, что мы можем точно так же взять производные формул свёрточных слоёв и применить алгоритм backpropagation для их обучения.

Подробно говорить о том, почему операция свёртки представляется в виде перемножения двух матриц, мы не будем. Достаточно понимать, что фильтры можно обучать с помощью backpropagation. Но если вам интересно узнать об этом подробнее, можете пройти по ссылке на слайде.

Посмотрим ещё раз на наш прототип свёрточной нейросети для классификации четвёрок и восьмёрок. Если здесь задать фильтры свёрточного слоя обучаемыми, то это уже будет полноценная свёрточная нейросеть. У неё будет один свёрточный слой, в котором два фильтра.

Однако классификация четвёрок и восьмёрок — это очень простая задача, и здесь не нужна свёрточная нейросеть. С этим справилась бы обычная полносвязная нейросеть. Задачи, которые встречаются в реальном мире, гораздо сложнее.

Пример — та же задача классификации картинок ImageNet. Чтобы решить задачу классификации картинок ImageNet на 1000 классов, одного слоя свёртки с двумя фильтрами недостаточно. Один слой может выделить информацию о том, где и какие линии есть на изображении: вертикальные, диагональные или другие. Но этой информации недостаточно, чтобы передать ее в полносвязную сеть и точно классифицировать изображение из ImageNet в один из тысячи классов. Эти изображения сложнее, чем набор нескольких простых линий. Хочется выделить из картинки более сложную информацию, например, наличие сложных составных элементов: кузова и колёс машины, ручки стакана, глаз, носа и лица в целом. И уже информацию о наличии этих более сложных, высокоуровневых паттернов передавать на вход полносвязной нейросети. Так полносвязной части будет гораздо проще научиться понимать, что за объект изображён на картинке.

Мы научимся это делать. Построим свёрточную сеть, которая сможет выделять информацию о наличии сложных паттернов на картинке. Для этого нам понадобится добавить в модель ещё несколько свёрточных слоёв.

Настоящая свёрточная сеть состоит из нескольких слоёв свёрток и полносвязной части. В слоях свёрток при этом может быть разное количество фильтров. В примере на слайде в первом слое свёрток пять фильтров, во втором — три. Давайте посмотрим, как происходит переход от одного свёрточного слоя к другому. Теперь мы будем подавать в сеть уже цветные картинки, а не чёрно-белые. Итак, первый свёрточный слой. Сворачиваем входящую картинку несколькими, например, пятью фильтрами, получаем пять карт активации. Для свёртки цветной картинки мы используем трёхмерные фильтры. Пусть они будут размером три на три на три. Каждый такой фильтр при свёртке картинки выдаёт одну двумерную карту активации размером тридцать на тридцать.

Далее, второй свёрточный слой. Им мы сворачиваем карты активации, которые получились на выходе первого слоя. Пять двумерных карт активации можно представить как одну трёхмерную матрицу, у которой третья размерность — 5. Это как будто цветная картинка размером тридцать на тридцать, у которой не 3 цветовых канала, а 5. Такую трёхмерную матрицу можно свернуть трёхмерными свёртками, у которых третья размерность тоже 5. Пусть это будут три фильтра размерностью четыре на четыре на пять. Тогда на выходе этого слоя мы получим три двумерные карты активации размером двадцать семь на двадцать семь.

Далее можно добавлять слои свёрток с разным количеством и размерами фильтров. После последнего слоя свёртки нужно растянуть все карты активации

последнего слоя в векторы, конкатенировать их и подать на вход полносвязной части сети, как мы делали выше.

В итоге схематично устройство свёрточной сети можно проиллюстрировать так, как вы видите на слайде. Операция flatten совмещает в себе растягивание всех карт активации последнего свёрточного слоя и их конкатенацию.

Обучается такая нейросеть обычным алгоритмом backpropagation. Все ядра всех свёрточных слоев, а также полносвязные слои, являются обучаемыми. По английски свёрточная нейросеть называется CNN: это сокращение от convolutional neural network. Далее в курсе я часто буду использовать слово CNN как замену длинной фразе «свёрточная нейросеть».

Кстати, мы почти её построили. Остался один нюанс: функция активации. Как и в полносвязной сети, после каждого свёрточного слоя обязательно нужна функция активации. Чаще всего, как и в полносвязных сетях, для свёрток используется функция активации ReLU.

Функция активации действует поэлементно на каждый пиксель карты активации. На слайде вы видите применение ReLU к карте активации: результат — все та же матрица два на два со значениями, которые были пропущены через ReLU.

Мы можем дополнить наше схематичное описание свёрточной нейросети: указать, что после каждого свёрточного слоя должна быть функция активации.

Кстати, для всех карт активации одного слоя обязательно будет применяться одна и та же функция активации. Так же, как и в полносвязной нейросети, все нейроны одного слоя проходят через одну и ту же функцию активации.

Мы сказали, что последовательные слои свёрток помогают модели выделять из картинок более сложные, высокоуровневые паттерны, чем просто горизонтальные или диагональные линии. Как же это получается? Оказывается, дело вот в чём. Каждый последующий слой свёрток сворачивает карты активации, выданные предыдущим слоем. В обученной нейросети получается так, что каждый следующий слой свёрток реагирует на всё более высокоуровневую и сложную информацию, которая присутствует на входном изображении.

Поясню. Пусть свёрточная нейросеть обучена классифицировать картинки из ImageNet, и пусть в нейросети 5 свёрточных слоёв. Среди картинок ImageNet есть картинки кошек. Возьмём картинку кошки и прогоним её через все пять слоёв свёрток. Визуализируем карты активации, полученные после каждого слоя свёрток, по одной на каждый слой.

Мы видим, что каждый последующий слой свёрток на карте активации подсвечивает всё более крупные части изображения: детализация снижается. Первые слои свёрток реагируют на низкоуровневые паттерны (наличие простых линий на изображении) и подсвечивают их на картах активации. Последующие слои свёрток берут информацию из карт активаций предыдущих слоев и реагируют на наличие на них более высокоуровневых паттернов: носа, глаз, ушей. Последний слой почти

полностью подсвечивает область на карте активации. Она соответствует области исходной картинке, в которой находится голова кошки. То есть последний слой свёрток реагирует на наличие на картинке паттерна «голова кошки».

Таким образом, наши свёрточные слои как бы обрабатывают картинку, распознавая на ней сначала низкоуровневые паттерны (горизонтальные/вертикальные/наклонные линии), а затем постепенно собирая эти паттерны вместе в более сложные конструкции, такие как глаза, уши, нос, и, в конце концов, в целую голову кошки.

В последнем свёрточном слое карты активации содержат информацию о наличии или отсутствии на изображении довольно сложных паттернов, и именно эту информацию получает на вход полносвязный слой. Например, разные карты активации последнего свёрточного слоя нейросети для классификации картинок ImageNet могут содержать информацию о том, есть ли на картинке кузов автомобиля, крыло самолёта, голова кошки или собаки. Согласитесь, по таким признакам, выделенным из картинок, полносвязной нейросети будет гораздо легче научиться эти картинки классифицировать. Чем если бы признаки были вида «ну, на картинке есть десять вертикальных линий тут и двадцать наклонных там».

Здесь уместно ввести понятие receptive field. Receptive field — это то, на какую часть изначального изображения «смотрит» тот или иной слой свёртки. Возьмём, к примеру, нейросеть, у которой два свёрточных слоя с одним

фильтром размером 3×3 в каждом. Посмотрим на центральный пиксель карты активации второго слоя, он выделен жёлтым. Этот пиксель получился в результате свёртки участка размером 3×3 предыдущей карты активации. Участок выделен красным. То есть пиксель содержит информацию из участка размером 3×3 предыдущей карты активации. В свою очередь, каждый пиксель красного участка первой карты содержит информацию из участка размером 3×3 изначального изображения. А в сумме пиксели красного участка первой карты содержат информацию из участка размером 5×5 изначального изображения.

Получается, что жёлтый пиксель второй карты активации содержит информацию из участка размером 5×5 изначального изображения. Синий участок 5×5 — receptive field желтого пикселя. Это те пиксели изначальной картинке, на которые «смотрит» желтый пиксель второй карты активации.

Видно, что пиксели карт активации каждого последующего слоя «смотрят» на большую часть входной картинке, чем пиксели карт активации предыдущих слоев. Это отлично согласуется с тем, что свёртки последующих слоёв реагируют на наличие всё более сложных конструкций на изначальном изображении. Первый свёрточный слой «смотрит» на довольно малые, локальные части картинке, и реагирует на локальные паттерны: прямые и изогнутые линии. Дальние слои «смотрят» на большие части картинке и реагируют на сложные, крупные паттерны изображения: морду или тело кота, голову человека или кузов автомобиля.

Именно в этом месте устройство свёрточных нейросетей похоже на то, как мозг человека обрабатывает изображения. Исследования человеческого зрения показали, что при передаче сигнала от глаз к области мозга, которая этот сигнал обрабатывает и осмысливает, происходит примерно то же самое. Сначала детектируются простые линии вроде горизонтальных или диагональных. Затем они собираются в более сложные объекты, которые мы и осознаём. Мы тоже понимаем, что за объект перед нами, по его отдельным кусочкам — линиям, формам. Наше восприятие похоже на то, как последовательные слои нейросети получают информацию из предыдущих слоев и собирают её в более высокоуровневые паттерны.

Итак, мы поговорили о том, что разные слои свёрток реагируют на разные по сложности паттерны. Замечу ещё вот что: разные фильтры реагируют на информацию разного типа. Разные фильтры, даже внутри одного слоя, реагируют на разные по семантике паттерны.

Что я имею в виду: вернёмся к нашей нейросети, которая обучена решать задачу классификации ImageNet. Так как в ImageNet тысяча классов, то не каждый фильтр обучается реагировать на наличие на картинке паттернов кошки. Разные фильтры в разных слоях реагируют на разные паттерны, которые есть на разных изображениях ImageNet. Во время обучения сети один фильтр учится реагировать на паттерн «колесо машины», другой — на паттерн «крыло самолёта», третий — «глаз кошки».

		<p>Чтобы это проиллюстрировать, возьмём нашу сеть, обученную на ImageNet, и подадим ей на вход картинку кошки. Визуализируем пять карт активации каждого слоя сети. Мы видим, что разные ядра разных слоёв сети выделяют различную информацию из картинки: карты активации отличаются. Никакие карты активации первого слоя не пусты, потому что первый слой свёрток реагирует на наличие низкоуровневых паттернов. Они есть почти на любой картинке: это диагональные, вертикальные линии и подобное. На последующих слоях сети всё больше карт активации становятся пустыми или почти пустыми. Фильтры этих слоёв реагируют на высокоуровневые паттерны, и каждый — на свой. Например, на часть головы кошки или крыла самолёта. Большинство фильтров реагируют не на части кошек, а на другие паттерны, присущие другим картинкам из ImageNet. Поэтому мы и видим, что карты активации дальних слоёв пустые, когда мы подали на вход картинку кошки.</p> <p>На какие паттерны будет реагировать тот или иной фильтр, нейросеть понимает в процессе обучения. В итоге фильтры учатся реагировать на те паттерны, которые помогают решить поставленную задачу</p>	
Закрепление изученного материала	20 мин.	<p>Каким образом свёрточные нейросети решают задачу классификации?</p> <p>Во-первых, свёрточные слои выделяют из входного изображения наличие определённых паттернов, характерных для того или иного класса (наличие головы кота, собаки, хвоста самолёта, кузова машины и т.д.). При</p>	Педагог организует беседу по вопросам

подаче на вход нейросети картинки, в нейросети активируются те фильтры, которые обучались реагировать на паттерны того типа, которые присутствуют на изображении.

В итоге карты активации последнего слоя сети содержат информацию о том, какие из этих паттернов на входном изображении есть, а каких нет.

Карты активации последнего слоя подаются на вход полносвязной части сети, которая по информации из карт определяет, какой именно объект изображен на входной картинке.

Напомню о том, что мы уже замечали в предыдущих видео. Из принципа работы свёрточной нейросети понятно, почему свёрточную часть часто называют выделятором признаков, а полносвязную часть — частью-классификатором. Фильтры свёрточных слоев действительно выделяют из картинки признаки — информацию о наличии тех или иных паттернов, на основе которых полносвязная часть картинки классифицирует. Этим устройство модели похоже на то, как решались задачи обработки картинок до эры нейросетей: там тоже из картинки выделялись признаки и подавались на вход классификатору. Главное отличие свёрточной нейросети от классических методов в том, что у нейросети выделение признаков тоже обучаемо, а не жёстко задано заранее. Поэтому неудивительно, что свёрточные нейросети работают намного лучше прошлых алгоритмов: они могут обучать свои части, которые выделяют признаки, под конкретную задачу.

В прошлом видео мы построили настоящую свёрточную нейросеть, у которой несколько свёрточных слоёв, а в этом видео поговорили о том, как эта нейросеть работает. Мы также поняли, почему свёрточные нейросети добились таких успехов в обработке изображений. Давайте кратко пройдемся по тому, какие преимущества есть у CNN перед обычной полносвязной сетью.

Во-первых, вспомним про принципы локальности и инвариантности к переносу. В свёрточной нейросети они прекрасно выполняются. Каждый пиксель карты активации содержит в себе информацию из локальной области изначального изображения. Нейросеть как будто умеет «концентрировать взгляд» на небольших областях картинки и затем собирать общее представление об изображении из этих локальных областей. Также все области картинки обрабатываются фильтрами одинаково: в каком месте изображения бы ни находился объект, результат свёртки его фильтрами слоёв будет одинаковым. Это отражает принцип инвариантности к переносу.

Кроме того, мы упомянули, что принцип работы свёрточных нейросетей похож на то, как работает человеческое зрение. Это тоже большой плюс CNN перед полносвязными сетями.

Эксперименты подтверждают, что эти свойства CNN действительно помогают им лучше решать задачи, связанные с картинками, чем это делают полносвязные сети. На практическом занятии мы с вами в этом убедимся.

		<p>Остался последний вопрос. Когда мы обсуждали недостатки полносвязных нейросетей для обработки картинок, мы говорили, что в них получается очень много параметров и сеть легко переобучается. Вопрос: а у свёрточных нейросетей что, не так? Кажется, что у CNN получается чуть ли не больше параметров, чем у чисто полносвязной сети.</p> <p>Смотрите: пусть я хочу обучить CNN классифицировать картинки размером сто на сто. У меня есть пять свёрточных слоёв с фильтрами размером три на три. И пусть в последнем свёрточном слое пять карт активации. Тогда размеры карт активации последнего слоя будут девяносто на девяносто, и после операции flatten мы получим вектор размером девяносто на девяносто на пять, что равно сорока тысячам пятистам. Получается, в первом слое полносвязной части должно быть сорок тысяч пятьсот нейронов, что жутко много. Сеть, конечно, переобучится.</p> <p>Но не отчаивайтесь: у этой проблемы есть довольно простое решение. Его мы и обсудим на следующем занятии.</p>	
Этап подведения итогов занятия (рефлексия)	8 мин.	Вопросы для обсуждения <ul style="list-style-type: none"> ● Чему я научился? ● С какими трудностями я столкнулся? ● Каких знаний мне не хватает для более глубокого понимания изученного материала? ● Достиг ли я поставленных целей и задач? 	Педагог способствует размышлению обучающихся над вопросами

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

Рекомендуемые ресурсы для дополнительного изучения:

1. Что такое свёрточная нейронная сеть. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/309508/>.
2. Свёрточная нейронная сеть (CNN). [Электронный ресурс] – Режим доступа: <https://www.helenkapatsa.ru/sviortochnaia-nieironnaia-siet/>.
3. Наглядно о том, как работает свёрточная нейронная сеть. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/skillfactory/blog/565232/>.