ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Свёрточные нейронные сети.

Аннотация к занятию: обучающиеся узнают, что такое нейросети для задач компьютерного зрения. Познакомятся с историей развития компьютерного зрения, узнают устройство нейросетей для обработки картинок и построят сеть для классификации изображений. Поймут, почему полносвязные сети плохо подходят для работы с картинками. Обсудят идею работы сверточной нейросети.

Цель занятия: сформировать у обучающихся представление о компьютерном зрении, о сверточных сетях, принципах ее работы, показать применение нейронных сетей для задач компьютерного зрения.

Задачи занятия:

- познакомиться с историей развития компьютерного зрения;
- сформировать представление об устройство нейросетей для обработки картинок;
- построить сеть для классификации изображений;
- познакомиться со сверточными нейросетями;
- понять идею работы сверточной нейросети.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Всем добрый день! Очень рада вас приветствовать на занятии. Как ваше настроение? Мы начинаем изучать нейросети для задач компьютерного зрения. Проще говоря, для работы с картинками.	
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	**Вопросы для обсуждения:** Как вы думаете, что вас сегодня ожидает на занятии? **Возможные ответы обучающихся:** • Пройдемся по истории развития компьютерного зрения, узнаем устройство нейросетей для обработки картинок и построим сеть для классификации изображений. • Мы поговорим о развитии компьютерного зрения и узнаем, как люди решали задачи, связанные с картинками, до эры нейросетей. • Начнем разговор о сверточных нейросетях.	Подробнее: https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/

		• Поймем, почему полносвязные сети плохо подходят для работы с картинками. • Обсудим идею работы сверточной нейросети.	
Изучение нового материала	50 мин.	**Вопросы для обсуждения:** Вспомним, что такое ImageNet? Возможные ответы обучающихся: ImageNet — это база изображений, поделенных на большое количество классов. По сути, это датасет, на котором можно решать задачу классификации картинок. **У ImageNet есть две версии:** • в одной, более старой версии, всего полтора миллиона картинок и они поделены на тысячу классов; • во второй версии, которая была собрана не так давно, уже четырнадцать миллионов изображений, и они поделены на двадцать тысяч классов. В прошлом модуле вы узнали, что на основе ImageNet в две тысячи шестом году был запущен конкурс классификации картинок. Участники	Для справки: база данных ImageNet — проект по созданию и сопровождению массивной базы данных аннотированных изображений, предназначенная для отработки и тестирования методов распознавания образов и машинного зрения.

		должны были обучать модели машинного обучения на тренировочной части из ImageNet и получать предсказания на тестовой. Каждый год победитель соревнования определялся по метрике accuracy. Вы видите часть таймлайна конкурса. Здесь показаны модели, которые выиграли конкурс в две тысячи десятом и две тысячи одиннадцатом годах, и их результаты по метрике accuracy. Модель-победитель десятого года имела около 28 процентов ошибок, модель одиннадцатого года решила задачу чуть лучше, сделав около двадцати шести процентов ошибок на тесте. Модели не были нейронными: они были основаны на идеях классического машинного обучения. Поговорим о том, как вообще решались задачи классификации картинок до того, как начали применяться нейросети. Мы рассмотрим алгоритм классификации картинок. Он основан на гистограмме ориентированных градиентов, или, сокращенно, HOG. Кроме этого алгоритма для обработки изображений существуют и другие, но идеи у них похожи.	Для справки: https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

		Вопросы для обсуждения: Что такое гистограмма ориентированных градиентов? **Возможные ответы обучающихся:** Это метод выделения признаков из картинки. Представим, что нам нужно решить задачу классификации картинок кошек и собак. Возьмем картинку. Для простоты картинки будут черно-белыми. Каждый пиксель черно-белой картинки — это число от нуля до двухсот пятидесяти пяти. Чем ярче и белее пиксель, тем число больше. Тогда картинка — это матрица чисел. Возьмем кусочек картинки размером пять на пять пикселей в том месте, где проходит граница между собакой и фоном. Нижние левые пиксели этого кусочка будут маленькими, это — черные пиксели головы собаки, а верхние правые пиксели будут большими, около двухсот пятидесяти пяти, потому что это пиксели белого фона картинки. Посмотрим на центральный пиксель кусочка, он выделен красным.	

		Посчитаем градиенты этого пикселя: вертикальный и горизонтальный. Горизонтальный градиент — это разница пикселя справа и слева от центрального. Он равен двумстам сорока пяти минус десять, то есть, двумстам тридцати пяти. Вертикальный градиент — это разница пикселя сверху и снизу от центрального. Он равен двумстам тридцати. Градиенты получились довольно большими по модулю. Вопросы для обсуждения: Как вы думаете, почему это произошло? Возможные ответы обучающихся: Это произошло как раз потому, что центральный пиксель находится на границе между собакой и фоном. Пиксели справа и сверху от центрального — это пиксели фона, они большие по модулю. Пиксели снизу и слева — это пиксели собаки, они малы по модулю. Поэтому их разности — градиенты центрального пикселя — выходят большими по модулю. Если бы мы взяли кусочек картинки только внутри собаки или внутри фона, градиенты центрального пикселя были бы гораздо меньше по модулю. На слайде вы видите пример подсчета градиентов для пикселя внутри собаки.	

		Так мы можем посчитать два градиента для каждого пикселя картинки. Для всех, кроме крайних пикселей изображения, но ими можно пренебречь. Из картинки мы получим вектор пар градиентов. Если в картинке n пикселей, вектор будет состоять из 2n чисел. Градиенты пикселей можно визуализировать. Если вместо значения каждого пикселя картинки подставить его горизонтальный градиент, получится то, что вы видите на экране в центре. Если подставить значения вертикальных градиентов, то получится картинка справа. С виду они очень похожи: обе показывают, где на исходном изображении находятся границы. Но если приглядеться, между ними будут различия: на картинке в центре сильнее выделены вертикальные линии, а справа — горизонтальные. Вернемся к нашему вектору из градиентов. В нем содержится информация о том, где на картинке находятся границы объектов. Можно рассматривать этот вектор как вектор признаков картинки. Заметим теперь, что информации о том, как выглядят края объектов на	

		изображении, достаточно, чтобы классифицировать эти изображения. Если показать вам контуры кошки и собаки, вы без труда определите, на какой из картинок изображена кошка, а на какой — собака. То есть, в векторе градиентов картинок содержится вся нужная информация, чтобы их классифицировать. Сформируем вектор градиентов для каждой картинки из датасета. И на этих векторах обучим какую-нибудь модель машинного обучения. Например, логистическую регрессию. Когда к нам будет приходить новая картинка, которую нужно классифицировать, нужно будет посчитать градиенты ее пикселей, сформировать из них вектор и подать на вход обученной логистической регрессии. Вот такой нехитрый алгоритм. Конечно, в реальности он используется не совсем в таком виде: после получения вектора градиентов его сжимают, чтобы сделать поменьше. На сжатых векторах алгоритм классификации работает лучше. Кроме того, по картинке вычисляется еще один вектор: вектор направлений градиентов.	Для справки: https://habr.com/ru/post/309508/ Ян Лекун — французский учёный в области информатики, основные сферы деятельности — машинное обучение, компьютерное зрение, мобильная робототехника и вычислительная нейробиология. Известен работами по применению нейросетей к задачам оптического распознавания символов и машинного зрения. AlexNet — сверточная нейронная сеть, которая оказала большое влияние на развитие машинного обучения, в особенности — на алгоритмы компьютерного зрения. Сеть с большим отрывом выиграла конкурс по распознаванию изображений ImageNet LSVRC-2012 в 2012 году (с количеством ошибок 15,3% против 26,2% у второго места).

		Он тоже используется для обучения алгоритма классификации. Но, несмотря на все эти дополнения, смысл алгоритма остается тем же. Как мы сказали ранее, кроме описанного алгоритма есть много других алгоритмов для решения задач, связанных с изображениями. Принцип их работы похож: сперва выделить признаки из картинки, затем обучить на них модель машинного обучения. Популярные способы выделения признаков, помимо HOG, показаны на слайде. Вы можете подробнее почитать о них в интернете. Ну а мы вернемся к конкурсу ImageNet. Как мы сказали ранее, в две тысячи десятом и две тысячи одиннадцатом годах в конкурсе побеждали модели, основанные не на нейросетях, а на идеях, похожих на ту, что мы только что разобрали. Но в две тысячи двенадцатом году конкурс внезапно выиграла сверточная нейросеть. Почему внезапно: как вы знаете, сверточные нейросети для работы с изображениями существовали и ранее. Их придумал Ян Лекун еще в девяносто восьмом году. Но до двенадцатого года ни у кого не	Архитектура AlexNet схожа с созданной Yann LeCum сетью LeNet. Однако у AlexNet больше фильтров на слое и вложенных сверточных слоев. Сеть включает в себя свертки, максимальное объединение, дропаут, аугментацию данных, функции активаций ReLU и стохастический градиентный спуск.

		получалось обучить сверточную нейросеть хорошо решать задачи, связанные с картинками: классические подходы оказывались лучше. Никто не верил, что нейросети когда-либо смогут хорошо обрабатывать картинки. Но в двенадцатом году сверточная сеть AlexNet доказала, что люди были не правы. Смотрите, она улучшила результат предыдущей модели аж на десять процентов: это очень круто. Этот момент принято считать революцией в компьютерном зрении: тут люди поверили, что сверточные нейронные сети — мощный инструмент, и стали их развивать. На графике видно, что это у них получилось. Все модели, побеждавшие в конкурсе ImageNet, были сверточными сетями, и раз за разом улучшали результат. В две тысячи пятнадцатом году нейросеть ResNet даже превзошла человека: сделала меньший процент ошибок при классификации картинок ImageNet, чем это делает человек. Так как же устроены эти ваши сверточные нейросети? Построим нейросеть, у которой в первом слое будет семьсот восемьдесят четыре	Для справки: https://medium.com/analytics-vidhya/understanding-gradients-in-machine-learning-60fff04c6400

		нейрона, а в последнем — десять. Напомним, что в последнем слое обязательно должно быть десять нейронов, потому что мы решаем задачу классификации на десять классов. В скрытых слоях сети может быть любое количество нейронов. Подавать картинку на вход сети мы будем вот как: берем картинку и растягиваем ее в вектор. Это делается очень просто: берем вторую строку картинки и присоединяем справа к первой строке. Получаем вектор из двадцати восьми умножить на два элементов. Точно так же справа присоединяем третью строку, четвертую и так далее. Получаем вектор, в котором будет семьсот восемьдесят четыре числа. Этот вектор мы и будем подавать на вход нейросети. По сути, каждый нейрон первого слоя будет принимать на вход значение одного из пикселей картинки. Обучим полносвязную сеть задаче классификации картинок с помощью алгоритма обратного распространения ошибки. Это мы обсуждали в предыдущем модуле.	

		Такой способ обучения нейросети классификации картинок рабочий, но работает он довольно плохо. **Вопрос для обсуждения:** Давайте обсудим, почему? **Ответы обучающихся:** Заметим, что у нашей нейросети получилось много нейронов в первом слое: семьсот восемьдесят четыре. Тогда у первого слоя сети будет большое количество обучаемых параметров: семьсот восемьдесят четыре умножить на n, где n — это количество нейронов в первом скрытом слое сети. Даже если n довольно мало, например, десять, то количество весов на ребрах первого слоя будет уже семь тысяч восемьсот сорок, что довольно много. А когда у сети много обучаемых параметров, она, скорее всего, переобучится. Представьте теперь, что наши картинки будут размером пятьсот на пятьсот или тысяча на тысяча. Тогда количество параметров нейросети будет во много раз больше, и нейросеть переобучится в два счета. Более того, сеть с таким огромным количеством параметров будет обучаться жутко долго.	

		Можно, конечно, перед обучением сети уменьшать размеры картинок датасета, скажем, до 30 на 30. Но так потеряются маленькие, но важные для решения задачи детали на картинках, и сеть будет работать плохо. • Это первая проблема использования полносвязных нейросетей для работы с картинками. • А вот вторая: хотя, это, скорее, не проблема, а недостаток. Смотрите: пусть перед нами стоит задача классификации картинок животных. Посмотрим на две картинки на слайде: это две картинки одной и той же собаки, просто собака расположена в разных местах изображений. Нам бы хотелось, чтобы нейросеть для классификации обрабатывала эти две картинки одинаково и выдавала на них совершенно одинаковый ответ, ведь с точки зрения задачи классификации совсем не важно, в каком месте картинки находится собака: это все еще собака, причем одна и та же. Но если мы подадим эти две картинки на вход полносвязной нейросети, она обработает их по-разному: пиксели	

		собаки на первой картинке будут обработаны верхними нейронами первого слоя сети, а пиксели собаки на второй картинке — нижними нейронами первого слоя сети. В результате и выходы последнего слоя сети для этих двух картинок будут немного разные. Свойство, когда модель одинаково обрабатывает одинаковые объекты, расположенные в разных местах картинки, называется инвариантностью относительно смещения. Как мы поняли, полносвязная сеть этим свойством не обладает. • Наконец, третий недостаток полносвязных нейросетей. Снова посмотрим на эту картинку собаки. Пиксели, принадлежащие собаке, расположены в небольшой компактной области в левом верхнем углу картинки. Это отражает принцип локальности объектов окружающего мира: чаще всего объект расположен на картинке в определенной области, а не размазан по всему изображению. Чтобы найти нужный объект на картинке, нужно сконцентрировать взгляд на локальных частях картинки, а не смотреть сразу на все изображение целиком. Иными	

		словами принцип локальности можно сформулировать так: чтобы понять, что происходит в отдельном пикселе изображения, нужно посмотреть только на несколько соседних пикселей, не нужно смотреть на все пиксели целого изображения. Но когда мы обрабатываем картинку с помощью полносвязной нейросети, принцип локальности не реализуется. Каждый нейрон второго слоя сети связан со всеми нейронами предыдущими слоя. Получается, каждый нейрон последующего слоя как бы «смотрит» на всю картинку целиком. Вот такие три недостатка есть у полносвязных нейросетей. Заметим еще вот что: человеческое зрение учитывает принципы локальности и инвариантности к переносу. Как мы сказали выше, для человека не важно, в каком месте картинки находится собака: он прекрасно понимает, что это ровно одна и та же собака. Так же и с принципом локальности. Разглядывая картинку в поисках нужных объектов, мы концентрируем взгляд на локальных областях картинки и не всегда воспринимаем все изображение с кучей объектов на нем. Получается, принцип	

		работы полносвязных сетей совсем далек от того, как работает человеческое восприятие объектов. На этом моменте мы переходим к разговору о том, как устроены сверточные нейросети. Сверточные нейросети следуют принципам локальности и инвариантности к переносу, и принцип их работы даже напоминает то, как работает человеческое зрение. Сейчас мы увидим, как именно. Поговорим об идее, на которой основан принцип работы сверточной нейросети. Посмотрим на две картинки из MNIST: четверку и восьмерку. **Вопрос для обсуждения:** Что отличает эти два объекта? **Ответы обучающихся:** Четверка преимущественно состоит из вертикальных и горизонтальных линий, а восьмерка — из изогнутых. Если бы мы строили алгоритм классификации картинок четверок и восьмерок, то признак «наличие горизонтальных и вертикальных линий» отлично разделял бы два этих класса.	

		Сверточные нейросети основаны на похожей идее обработки изображений: они выделяют из картинок информацию о наличии либо отсутствии на картинке неких паттернов, и на основе этой информации понимают, что изображено на картинке. Вертикальные/изогнутые линии — это примеры простых паттернов. Похоже на то, как мозг человека обрабатывает объекты окружающего мира. Это и то, что сверточные нейросети следуют принципам локальности и инвариантности относительно переноса, делает их такими эффективными в работе с картинками. Итак, в этом видео мы поговорили о том, почему полносвязные нейросети плохо справляются с обработкой изображений. Обсудили идею того, как работают сверточные нейросети. В следующих видео мы подробно разберем их устройство.	
Закрепление изученного материала	15 мин.	**Вопросы для обсуждения:** • Расскажите историю развития компьютерного зрения, конкурс ImageNet	Педагог организует беседу по вопросам.

		• Какой была компьютерное зрение до нейросетей? • Почему полносвязные нейросети плохо подходят для работы с изображениями? • Что такое устройство свертки?	
Этап подведения итогов занятия (рефлексия)	8 мин.	**Вопросы для обсуждения:** • 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. HOG. [Электронный ресурс] – Режим доступа: https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f ;
2. Как работает сверточная нейронная сеть. [Электронный ресурс] – Режим доступа: https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/ ;
3. Что такое свёрточная нейронная сеть. [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/309508/ ;
4. Градиентный метод. [Электронный ресурс] – Режим доступа: https://qudata.com/ml/ru/ML_Grad_Method.html