

ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Архитектуры CNN

Аннотация к занятию: обучающиеся изучат популярные архитектуры свёрточных нейросетей и узнают несколько интересных возможностей, которые помогают CNN лучше решать задачи, связанные с изображениями.

Цель занятия: формирование у обучающихся представления об архитектурах свёрточных нейросетей и их видах, способах работы с ними.

Задачи занятия:

- изучить архитектуры CNN: AlexNet, VGG, ResNet и DenseNet;
- проанализировать виды архитектур CNN, выявить их особенности;
- выявить проблемы, возникающие при работе со свёрточными нейросетями, разобрать методы борьбы с ними.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	<p>Добрый день! Давайте вспомним про ImageNet. Что это такое? В чём её особенность?</p> <p>Возможные ответы обучающихся:</p> <ul style="list-style-type: none"> • ImageNet — это большая база изображений; • у ImageNet есть две основные версии: большая, в которой пятнадцать миллионов картинок, разделённых на двадцать одну тысячу классов; и версия поменьше, где полтора миллиона картинок и тысяча классов. <p>Кроме того, мы помним, что в 2010 году был запущен конкурс по классификации картинок ImageNet. Конкурс проводился на версии с тысячей классов. На слайде таймлайн конкурса. В прошлом модуле мы выяснили, что две первые модели, победившие в конкурсе в 2010 и 2011, были не нейросетевыми. Однако в 2012 году конкурс выиграла свёрточная нейросеть AlexNet, сразу на десять процентов улучшив результат классификации картинок. AlexNet сделала всего пятнадцать процентов ошибок на тестовой выборке. Все модели, которые</p>	<p>Приветствие. Создание в классе атмосферы психологического комфорта</p>

		<p>выигрывали конкурс далее год за годом — тоже свёрточные нейросети.</p> <p>Благодаря чему каждый год удавалось улучшить результат?</p> <p>Возможный ответ обучающихся: нейросеть содержала большее количество слоев.</p> <p>Верно, каждый год нейросеть, которая выигрывала конкурс, была глубже, чем предыдущие модели. Глубже — это значит, что нейросеть содержала большее количество слоёв. Смотрите: AlexNet в 2012 году содержала всего восемь слоёв. VGG в 2014 — уже 19 слоёв. Дальше нейросети становились ещё глубже, а ResNet, выигравшая конкурс в 2015 году — это нейросеть, в которой 152 слоя. Логично, что добавление большего количества слоёв улучшает качество работы нейросетей. Классификация картинок ImageNet — сложная задача, в датасете много классов, и на картинках есть много различных сложных форм. А чем больше у нейросети свёрточных слоев, тем больше различных сложных паттернов они могут выделить из картинки. Однако не всё так просто. Нейросети отличает не только количество слоёв. Более того, без некоторых дополнительных трюков, которые мы обсудим далее, нейросети с большим количеством слоёв не смогли бы показать такой хороший результат.</p>	
--	--	--	--

<p>Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся</p>	<p>7 мин.</p>	<p>Какая сегодня тема урока? Возможный ответ обучающихся: сверточные нейросети.</p> <p>Тема урока — «Архитектуры CNN».</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 архитектуры CNN; • узнаем, какие виды архитектур существуют и чем они отличаются. <p>Мы:</p> <ul style="list-style-type: none"> • узнаем, как устроена сеть AlexNet; • обсудим устройство ещё трёх нейросетей: VGG, ResNet и DenseNet; • узнаем несколько полезных трюков, используемых в архитектуре этих моделей, которые делают их работу ещё более эффективной. <p>Но есть и другая причина. Дело в том, что когда исследователи хотят решить какую-нибудь задачу с помощью нейросети, часто они не пишут свою нейросеть с нуля, а берут какую-то из стандартных архитектур, вроде ResNet или DenseNet, и обучают её под свою задачу. Это удобно: архитектуры доказали, что они эффективно решают задачу классификации. Вместо того, чтобы самому настраивать архитектуру: количество слоев, количество фильтров в слоях, пулинг и прочее, можно взять готовую удачную архитектуру, например, ResNet, и обучить её. То, какую архитектуру из всех выбрать, зависит от типа вашей задачи, оттого,</p>	<p>Способствовать обсуждению мотивационных вопросов</p>
---	---------------	--	---

		<p>насколько задача сложная и сколько у вас данных для обучения. Чем меньше у вас данных и чем проще ваша задача, тем более лёгкую, менее глубокую сеть вам нужно взять, чтобы она не переобучилась. Например, для классификации картинок MNIST или CIFAR подойдёт и самая простая сеть AlexNet. Если же ваша задача сложная, картинки разных форм и множество классов, как в ImageNet, вам понадобится нейросеть с большим количеством слоёв.</p> <p>Более того, классические нейросети типа ResNet или VGG, которые уже были обучены классифицировать картинки ImageNet, можно дообучать под вашу задачу. Дообучение и его роль в работе нейросетей мы обсудим позже.</p>	
<p>Изучение нового материала</p>	<p>50 мин.</p>	<p>Знать устройство классических нейросетей для классификации картинок важно. Давайте переходить к первой нейросети из нашего списка — AlexNet.</p> <p>Это самая обычная свёрточная нейросеть по типу той, что мы с вами построили в предыдущем модуле. Здесь вы видите схему её устройства. В ней всего пять сверточных слоёв и три полносвязных. На вход AlexNet принимает цветную картинку размером двести двадцать семь на двести двадцать семь. На схеме входящая картинка показана первым прямоугольником. Далее пять прямоугольников символизируют свёрточные слои. Под ними подписан размер карт активации, которые они выдают. Например, первый свёрточный слой выдаёт девяносто шесть карт</p>	

активации размером пятьдесят пять на пятьдесят пять каждая, второй — уже двести пятьдесят шесть карт активации размерами двадцать семь на двадцать семь и так далее. Три последних вытянутых прямоугольника символизируют три полносвязных слоя. Количество нейронов на выходе каждого из них подписано внизу. Последний слой выдаёт тысячу значений, потому что ImageNet — это задача классификации на тысячу классов.

Можно заметить, что выбор размеров фильтров в слоях немного странный. Например, после первого свёрточного слоя из картинки размером двести двадцать семь получается карта активации размером пятьдесят пять. Это достигается тем, что в первом свёрточном слое фильтры имеют размер одиннадцать на одиннадцать, и фильтры проходятся по картинке со страйдом четыре. Какие большие фильтры и страйд! В предыдущем модуле мы говорили, что такие большие по размеру фильтры и страйд сейчас уже никто не использует. Но AlexNet — это самая первая успешная свёрточная нейросеть, поэтому неудивительно, что её архитектура не самая эффективная.

На этой схеме не видно, но между некоторыми свёрточными слоями у AlexNet есть MaxPooling и батч-нормализация, а после двух первых полносвязных слоёв используется dropout. Ко всем слоям, кроме самого последнего полносвязного, применяется функция активации ReLU. Получается, AlexNet состоит из известных нам элементов: свёрточных и

полносвязных слоёв, слоёв пулинга и батч-нормализации, а также функции активации ReLU и дропаута. При желании вы могли бы сами реализовать архитектуру этой нейросети в коде. Итак, AlexNet выиграла конкурс ImageNet в 2012 году и стала первой свёрточной нейросетью, которая смогла это сделать. Двигаемся дальше: обратим внимание на VGG.

VGG — это тоже обычная свёрточная нейросеть, которая состоит из известных нам элементов. Здесь вы видите схему её устройства. Элементы, из которых состоит VGG, — те же, что у AlexNet: это свёрточные слои, полносвязные слои, MaxPooling, батч-нормализация, dropout и функция активации ReLU.

Вопрос для обсуждения

Подумаем, что же тогда делает VGG более эффективной, чем AlexNet?

Ответы обучающихся

Во-первых, размеры всех фильтров свёрточных слоев VGG — три на три. В прошлом модуле мы обсуждали, почему использовать фильтры размером три на три лучше, чем фильтры большего размера.

Вопрос для обсуждения

Давайте вспомним, почему?

Ответы обучающихся

Верно, они позволяют нейросети выделить больше информации из картинки и не пропустить мелкие детали.

Во-вторых, у VGG стало больше свёрточных слоёв. Полносвязных слоёв так же, как и у AlexNet, осталось три, а свёрточных слоёв стало больше. На самом деле, есть разные варианты нейросети VGG, у которых разное количество слоёв, но в любом варианте их всё равно больше, чем у AlexNet. Есть варианты VGG, в которых одиннадцать, тринадцать, шестнадцать и девятнадцать слоёв. Большее количество свёрточных слоев помогает нейросети выделять больше разнообразных паттернов на изображении.

Заметим также, что архитектура VGG устроена более логично, не так хаотично, как AlexNet. Посмотрим на схему. Здесь синие параллелепипеды символизируют свёрточные слои, красные — слои MaxPooling. Под свёрточными слоями написано, сколько карт активации и какой размерности они выдают.

VGG принимает на вход картинку размером двести двадцать четыре на двести двадцать четыре. Далее входящая картинка последовательно проходит через несколько блоков. Каждый блок — это несколько слоёв свертков, синих прямоугольников и один MaxPooling с ядром два, красный прямоугольник. Внутри блока у

каждого свёрточного слоя есть фильтры размером три на три и паддинг размером один. Поэтому он выдаёт карты активации того же размера, что и получал на вход: мы видим, что подряд идущие синие прямоугольники имеют один и тот же размер. А каждый MaxPooling, красный прямоугольник, уменьшает размер карт активации в два раза. Получается, после каждого блока размер карт активации уменьшается в два раза.

Таким образом, VGG имеет такую блочную структуру. Это позволяет легко создавать различные варианты VGG и проводить эксперименты. Например, можно добавить ещё один блок свёрток и MaxPooling, можно попробовать увеличить или уменьшить количество свёрточных слоев в каждом из блоков.

В AlexNet же никакой структуры не было. Разные свёрточные слои AlexNet выдавали карты активации совершенно разного размера. Вот такие отличия VGG от AlexNet позволили VGG показать ещё лучший результат в конкурсе ImageNet: всего семь и четыре процента ошибок. После успеха VGG люди задумались: смотрите, у нас был AlexNet, он показывал неплохие результаты на ImageNet. Затем мы добавили в нейросеть больше свёрточных слоев, получили VGG. VGG стал показывать лучший результат. При этом, кроме добавления большего числа слоев и уменьшения размера всех фильтров до три на три, мы ничего не

поменяли. Добавим ещё больше свёрточных слоёв, больше таких блоков, как у VGG, и получим нейросети лучше, которые покажут лучшие результаты на ImageNet.

Исследователи сделали так, и у них ничего не вышло. Почему-то нейросети с большим количеством слоёв начинали показывать даже худший результат, чем VGG: они как будто не обучались. Что это за магия, почему так происходит и как заставить глубокие нейросети обучаться и показывать отличные результаты, мы обсудим далее.

Вспомните, мы уже упоминали, что у нейросети ResNet сто пятьдесят два слоя и всего три процента ошибок на ImageNet. А значит, решение проблемы есть, и люди всё же смогли заставить глубокие нейросети обучаться.

Вопрос для обсуждения
Каким образом?

Ответы обучающихся

Эта проблема называется проблемой затухания градиентов. Её суть состоит в следующем.

Вспомним, мы разобрали устройство нейросетей AlexNet и VGG и поняли, что успех VGG во многом был связан с тем, что в этой нейросети было больше слоев, чем у AlexNet. Появилась идея: что если и дальше увеличивать количество слоёв нейросети? Тогда она

будет ещё лучше решать задачу классификации? Оказалось, это не работает. Если увеличить количество слоёв ещё сильнее, то нейросеть перестает обучаться: качество её классификации остаётся плохим.

Почему так происходит? Оказывается, в очень глубоких нейросетях возникает проблема затухания градиентов. Рассмотрим её подробнее.

Возьмём упрощенную модель нейросети. В ней будет три слоя, в каждом один нейрон. Для простоты давайте скажем, что в нашей нейросети нет функций активации и байеса. То есть, выход первого слоя x_1 вычисляется как вес первого слоя ω_1 умножить на вход x . Выход второго слоя x_2 — это вес второго слоя ω_2 умножить на x_1 , далее x_3 равно ω_3 на x_2 , и, наконец, выход сети подаётся в лосс-функцию, которую мы обозначим как L .

Обучается такая нейросеть с помощью алгоритма back propagation.

Вспомним, как выглядят формулы обновления весов нейросети. Чтобы обновить вес ω_3 , нужно вычислить производную лосса L по ω_3 . Тогда новый вес ω_3 равен старый вес ω_3 минус α умножить на dL по $d\omega_3$. α здесь — это learning rate. По правилу вычисления производной

сложной функции, dL по dw_3 расписывается как dl по dx_3 умножить на dx_3 по dw_3 . Аналогично можно записать формулы обновления для весов w_2 и w_1 . Для w_2 , к примеру, формула обновления такая: w_2 равно w_2 минус альфа умножить на dL по dw_2 . dL по dw_2 расписывается по правилу производной сложной функции уже в три множителя: dL по dx_3 , dx_3 по dx_2 и dx_2 по dw_2 . Вы видите формулу обновления веса w_1 . В этой формуле нужна производная dL по dw_1 расписывается длиннее: в произведение уже четырёх множителей. Если бы наша нейросеть была ещё глубже, с ещё большим количеством слоёв, то производные весов её первых слоёв расписывались бы в ещё большее количество множителей.

Посмотрим на выражение для dl по dw_1 . Это произведение четырёх производных. Чаще всего при обучении нейросетей бывает так, что все эти производные по модулю сильно меньше единицы. Получается, это произведение четырёх чисел, намного меньших единицы, то есть, очень маленькое число. Представьте, что, например, все множители были бы равны одной десятой. Перемножив их вместе, мы получили бы число десять в минус четвёртой степени, то есть одну десятитысячную. Это довольно маленькое число.

Итак, произведение по модулю очень мало. А потом оно умножается на альфу — тоже очень маленькое число. В итоге вес w_1 при обучении меняется на малое значение.

Теперь представьте, что в сети не три слоя, как здесь, а пятьдесят. В процессе обучения такой сети веса первых слоев будут каждый раз изменяться на очень малое число. В формуле изменение веса — это произведение огромного числа малых по модулю значений. Можно сказать, что начальные слои глубокой нейросети практически не будут обучаться: их веса практически не будут меняться. Неудивительно, что глубокая нейросеть будет работать плохо.

С этой проблемой столкнулись исследователи, когда попытались создать нейросеть с большим числом слоёв. Как же эту проблему решить? Один из самых эффективных вариантов — skip connection. Разберёмся, что это такое. Skip connection — это добавление связи между непоследовательными слоями нейросети. Посмотрим снова на нашу модель. Добавим к ней связь между первым и третьим слоем. Теперь к результату работы третьего слоя будет добавляться результат работы первого слоя. Обозначим новый выход третьего слоя через x_3' . x_3' будет равен w_3 умножить на x_2 плюс x_1 .

Эта дополнительная связь и есть skip connection. С английского это так и переводится: «связь, пропускающая несколько элементов», то есть слоёв. Как же такая связь поможет с проблемой затухания градиентов? Посчитаем градиент для обновления веса w_1 для сети со skip connection.

Как и прежде, обновление веса ω один будет происходить по формуле ω_1 новое равно ω_1 старое минус α умножить на dL по d ω один. Давайте распишем dL по d ω один по правилу производной сложной функции. Получится выражение, которое вы видите на экране. Не будем вдаваться в подробности вычисления. Вы можете самостоятельно проделать вычисления и убедиться, что формула верна.

Посмотрим на полученное выражение: оно выделено красной рамкой. Теперь это не произведение четырёх множителей, как было до добавления skip connection. Внутри одного множителя появилось слагаемое — единица. Сравним, как изменилось выражение для dL по d ω один до и после добавления skip connection. То, что было, вы видите слева на слайде. dL по d ω один было равно произведению четырёх множителей.

То, что стало после добавления skip connection, справа на слайде. Представим, что в этих выражениях все производные равны одной десятой. Тогда до skip

		<p>connection значение dL по d ω один было бы равно одной десятичной. А после — уже одной сотой. Добавление skip connection увеличило значение производной в сто раз.</p> <p>Получается, при обучении нейросети градиентным спуском каждый шаг вес ω 1 будет меняться на величину, в сто раз большую, чем это было без skip connection. И нейросеть будет учиться лучше.</p> <p>Вот так skip connection помогает побороть проблему затухания градиентов. Мы разобрали идею skip connection на примере очень простой нейросети, в которой не было функций активации, байеса, и был всего один нейрон в каждом слое. Но всё то, о чём мы говорили, применимо также и к обычным нейросетям.</p> <p>Итак, мы познакомились с идеей skip connection. Построим свёрточные нейросети и обсудим, как skip connection применяется к ним.</p> <p>Рассмотрим нейросеть, у которой четыре свёрточных слоя с одинаковым количеством карт активации. Пусть в каждом слое будет шесть карт активации. Размеры фильтра в каждом слое будут три на три.</p> <p>Пусть мы хотим добавить skip connection между первым и третьим слоями сети.</p>	
--	--	---	--

Что это значит: как и в нашем прошлом примере, третий слой сети должен выдать на выходе не просто свои карты активации, а сумму своих карт активации и карт активации первого слоя. То есть четвёртый свёрточный слой принимает на вход сумму карт активации первого и третьего слоёв.

Сумма считается по каждой карте активации отдельно. Первый слой выдаёт шесть карт активации, и третий слой выдаёт шесть карт активации. Первая карта активации первого слоя поэлементно складывается с первой картой активации третьего слоя, вторая — со второй и так далее. Таким образом, чтобы можно было использовать skip connection между свёрточными слоями, важно, чтобы эти слои выдавали одинаковое количество карт активации.

Есть и ещё один нюанс: так как сложение карт активации первого и третьего слоёв происходит поэлементно, то карты активации должны быть одинакового размера. В нашей нейросети это не так: карты активации каждого следующего слоя становятся чуть меньше, потому что мы используем фильтры размером три на три. Но это легко исправить: тут нам снова поможет padding.

К каждому свёрточному слою добавим padding размером один. Тогда все карты активации всех слоев

будут одинакового размера: в нашем случае тридцать два на тридцать два. Можно будет их складывать. То есть, можно будет применять skip connection.

Итак, мы научились применять skip connection к свёрточным слоям. Пора познакомиться с нейросетью ResNet, архитектура которой основана на skip connection.

На этом слайде вы видите архитектуру сети ResNet-34. У этой версии resNet тридцать четыре слоя. Бывают и другие версии, с другим количеством слоев: ResNet-восемнадцать, сорок девять, пятьдесят, сто, сто пятьдесят два.

На этой схеме все цветные блоки — это свёрточные слои. Внутри блока слоёв одного цвета карты активации этих слоев имеют одинаковый размер. Между блоками разных цветов применяется операция пулинга, и размер карты активации уменьшается в два раза.

Все стрелки между несоседними слоями — это skip connection. Как видно, ResNet — это обычная свёрточная нейросеть с большим количеством слоёв, у которой также есть довольно много skip connections.

Название ResNet образовано от слов Residual Network. Residual — это другое название skip connection. Residual

с английского переводится как «остаточный». Получается, residual connection — это остаточная связь.

Рассмотрим, почему skip connection называют остаточной связью.

У skip connection есть ещё один смысл. Посмотрим снова на нашу свёрточную нейросеть с остаточной связью между первым и третьим слоем. Мы уже поняли, что благодаря skip connection четвёртый свёрточный слой принимает на вход сумму карт активации первого и третьего слоёв.

Получается, что второму и третьему слою нужно в своих картах активации выделить только ту информацию, которой нет на картах активации первого слоя. Потом эта информация сложится с картами активации первого слоя и пойдёт на вход четвёртому слою. Второму и третьему слою не нужно «тащить» всю информацию из карт активации первого слоя через себя, им достаточно выучить остаточную информацию, ту, которой не хватает. А это сделать гораздо проще.

Получается, skip connection, или residual connection, упрощает работу слоям нейросети. И нейросеть начинает обучаться еще эффективнее.

Вот такая элегантная идея skip connection, которая спасает от проблемы затухания градиентов и добавляет эффективности свёрточной нейросети. Как мы видим на таймлайне ImageNet, идея skip connection позволила нейросети ResNet улучшить результат классификации и даже превзойти в этом человека. На сегодняшний день идея skip connection используется во множестве различных архитектур нейросетей, не только в компьютерном зрении.

Поговорим ещё об одной нейросети: она называется DenseNet. Её нет на таймлайне ImageNet, но это тоже одна из популярных свёрточных архитектур. Её идея основана на skip connection, но чуть в другом стиле. Вместо того, чтобы складывать карты активации двух слоёв, в DenseNet карты активации конкатенируются. То есть в нашем примере четвёртый свёрточный слой принимал бы на вход двенадцать карт активации: шесть из первого слоя и шесть из третьего.

Вопрос для обсуждения

Какие проблемы есть у свёрточных нейросетей? Как их решить?

Ответы обучающихся

Это проблема затухания градиентов. Методом борьбы с этой проблемой является skip connection. ResNet и

		DenseNet — это свёрточные нейросети, в разных видах использующие skip connection.	
Закрепление изученного материала	15 мин.	Вопросы для обсуждения <ul style="list-style-type: none"> • Какие архитектуры CNN существуют? • В чём особенности работы с различными свёрточными нейросетями? • Какие проблемы возникают при работе со свёрточными нейросетями? • Какие методы борьбы с этими проблемами существуют? 	Педагог организует беседу по вопросам
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.	Сегодня вам предстоит помочь телекомпании FOX в обработке их контента. Как вы знаете, сериал "Симпсоны" идет на телеэкранах более 25 лет, и за это время скопилось очень много видеоматериала. Персонажи менялись вместе с изменяющимися графическими технологиями, и Гомер Симпсон-2018 не очень похож на Гомера Симпсона-1989. В этом задании	

		<p>вам необходимо классифицировать персонажей, проживающих в Спрингфилде. Думаю, нет смысла представлять каждого из них в отдельности. Конкурс на kaggle: https://www.kaggle.com/competitions/journey-springfield</p> <p>Зачёт задания будет происходить по качеству (F1-score), которого вы достигнете на тестовой выборке (public leaderboard) при сдаче задания в конкурс. Максимальное количество баллов за задание равняется 15. В отличие от обычных конкурсов на kaggle, в этом конкурсе в зачёт идёт ваш результат на публичной части датасета. Критерии оценки следующие: 0.97: 15 баллов 0.96: 14 баллов 0.95: 13 баллов ... 0.84: 2 балла 0.83: 1 балл</p> <p>Используйте все полученные знания, в том числе навыки обучения свёрточных нейросетей, чтобы добиться наилучшего качества. Рекомендуем активно использовать GPU на Google Colab, поскольку так свёрточные нейросети будут учиться во много раз быстрее (скорее всего, у себя на процессоре вы не сможете обучить даже простую CNN).</p>	
--	--	---	--

Рекомендуемые ресурсы для дополнительного изучения:

1. Архитектуры CNN. [Электронный ресурс] – Режим доступа: https://msu.ai/109_cnn_architectures
2. Deep Learning: как это работает? Часть 3 — архитектуры CNN [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/511372/>.
3. Что такое сверточные сети и как они работают. [Электронный ресурс] – Режим доступа: <https://python-school.ru/wiki/convolutional-neural-network/>.