

Титульный лист материалов по дисциплине
(заполняется по каждому виду учебного материала)

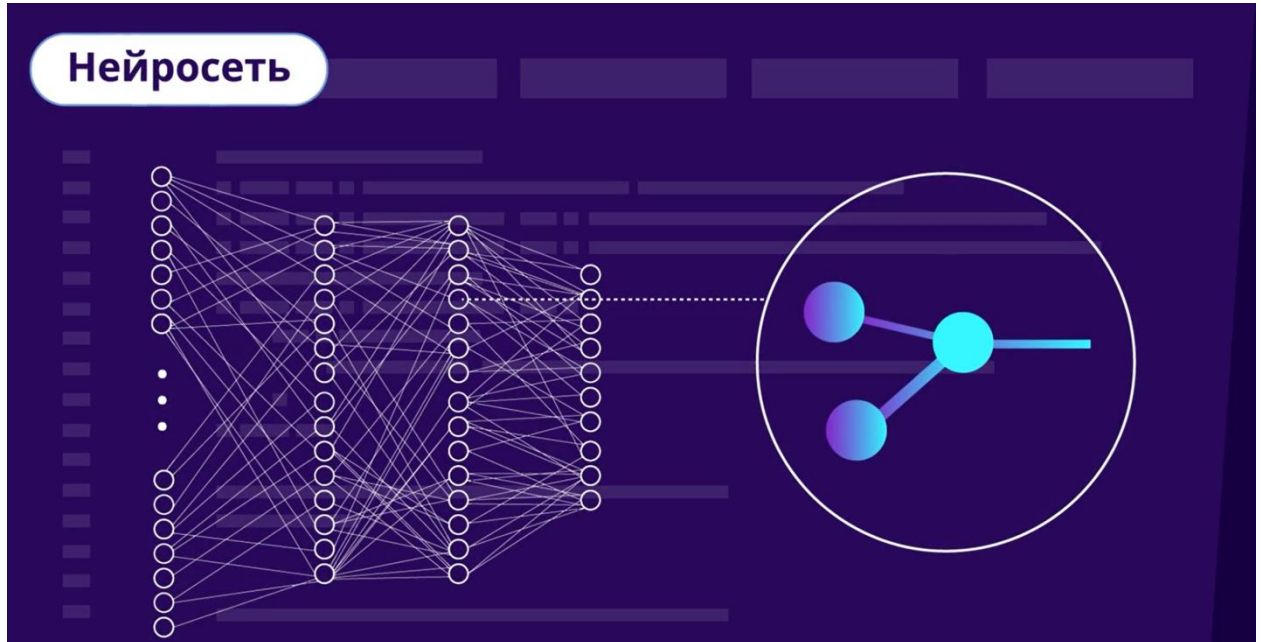
ДИСЦИПЛИНА	Технологии извлечения знаний из больших данных <small>(полное наименование дисциплины без сокращений)</small>
ИНСТИТУТ	ИКБ
КАФЕДРА	Кафедра КБ-14 «Цифровые технологии обработки данных» <small>полное наименование кафедры)</small>
ВИД УЧЕБНОГО МАТЕРИАЛА	Лекция <small>(в соответствии с пп.1-11)</small>
ПРЕПОДАВАТЕЛЬ	Никонов В.В. <small>(фамилия, имя, отчество)</small>
СЕМЕСТР	3 семестр 2023/2024 уч. года <small>(указать семестр обучения, учебный год)</small>

Глубокое обучение и нейросети

Принципы работы нейронных сетей

В отличие от других методов, нейронные сети очень гибкие и вариативны: их можно подстроить под разные задачи, разную специфику данных, для каждой задачи можно пробовать и придумывать много разных видов нейронных сетей, усложнять и упрощать модели. Поэтому нейронные сети выделились в отдельную область машинного обучения, называемую **глубоким обучением** (англ. *deeplearning*, еще используют термин «глубинное обучение»). Как мы обсудим ниже, слово «глубина» в названии отсылает к так называемым глубоким нейронным сетям, то есть таким, которые могут обнаруживать очень сложные зависимости в данных.

Благодаря своей гибкости нейросети очень хорошо справляются с анализом «сложных» данных: изображений, видео, текстов, звука. Нейросети позволили решить множество задач по автоматизации работы человека (и не только) и сегодня широко применяются в различных сферах деятельности.



Нейросеть - это последовательность связанных друг с другом слоев искусственных нейронов. Каждый нейрон можно представить как функцию с несколькими входами и одним выходом. Его задача - собрать информацию со входов, обработать ее и выдать результат - какое-то число - на выход. Входы и выходы нейрона связывают его с другими нейронами. У каждой связи есть

вес. Именно он определяет, на сигналы от каких соседей нейрон будет реагировать больше, а на какие - меньше.

Например, у нас есть нейрон, который через две связи с определенными весами, допустим, 0,3 и 1, получает числа, складывает их, сравнивает с числом 10 и выдает результат. 1 (истина), если сумма больше 10, и 0 (ложь) - во всех остальных случаях. Допустим, на входы поступили числа 10 и 4. С учетом веса связей до нейрона добрались числа 3 и 4. Их сумма меньше 10, поэтому нейрон на выходе выдаст 0 (ложь).

От линейных моделей к нейронным сетям

Чтобы разобраться с тем, что такое нейронная сеть, оттолкнемся от линейных моделей: нейронная сеть основывается на тех же принципах, но извлекает нелинейные зависимости из данных.

Вспомним, что линейные модели выполняют предсказания, умножая признаки на веса и складывая, например:

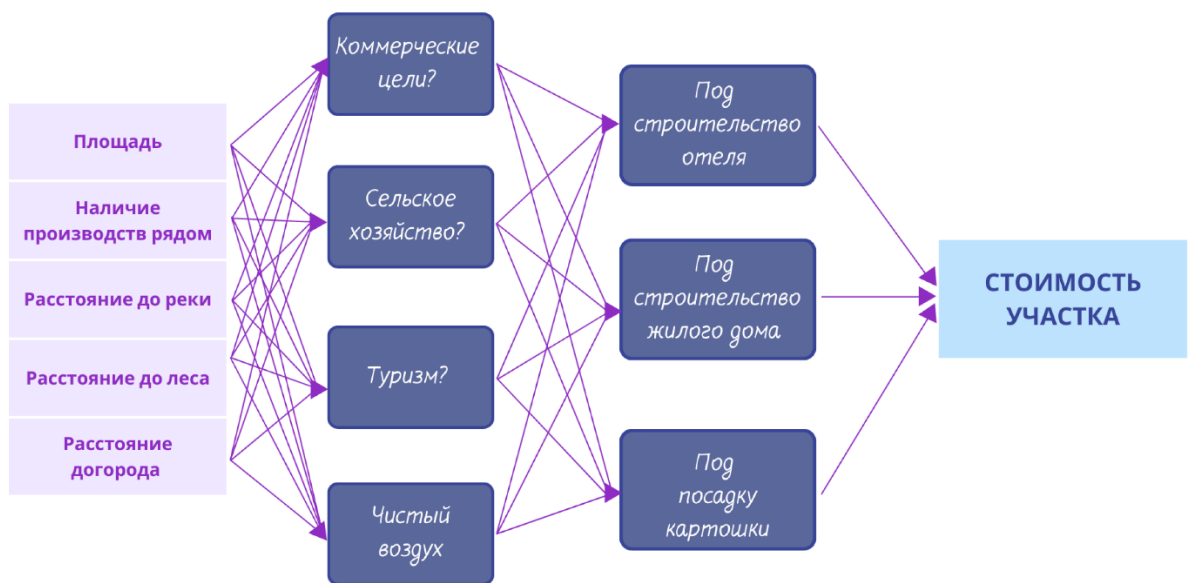
$$\text{Цена участка} \approx 2,5 \cdot \text{площадь} + 4,1 \cdot (\text{есть ли рядом река}) + 10 \cdot (\text{есть ли удобная дорога})$$

Площадь, наличие реки и дороги — это признаки, входные данные, а числа 2,5; 4,1 и 10 — это веса, которые настраиваются во время обучения по обучающим данным. Иными словами, чтобы сделать предсказание стоимости участка с помощью линейной модели, нужно использовать заданную формулу, подставив в нее информацию, известную об участке, и конкретный набор чисел.

Нейронные сети работают так же, но формула имеет более сложный вид. В нее могут входить умножения, сложения, деления, математические функции и т. д., но все эти операции выполняются с двумя видами величин: информацией, известной об объекте (признаками), и набором чисел (весами). Составляют такую формулу из известных блоков, называемых **слоями нейронной сети**. Слои можно комбинировать в разном порядке, повторять по много раз и даже ставить параллельно — большой простор для фантазии. Конкретная комбинация используемых слоев называется **архитектурой**

нейронной сети. Количество слоев, поставленных друг за другом, обычно называют **глубиной нейронной сети.** Чем глубже нейронная сеть, тем сложнее формула и тем сложнее зависимости, которые нейросеть может обнаруживать в данных.

Рассмотрим иллюстративный пример: будем предсказывать стоимость участка по признакам: площадь, наличие рядом производств, расстояние до реки, леса и ближайшего города (задача регрессии).



Трехслойная нейронная сеть

У нейронной сети на изображении три слоя.

1. Первый слой распознает простые характеристики участка: подходит ли участок под коммерческие цели, сельское хозяйство и т. д. Эти характеристики не записаны во входных данных (признаках), но могут быть определены по признакам. Например, подходит ли участок под коммерческие цели, может определяться по площади и расстоянию до города, а подходит ли участок под туризм — по наличию (а точнее отсутствию) рядом производств, расстоянию до реки и до города (чтобы туристов было недалеко возить).

2. Второй слой распознает еще более сложные характеристики участка, опираясь на результаты первого слоя. Например, подходит ли

участок под строительство отеля, может определяться по тому, подходит ли он под туризм и под коммерческие цели.

3. Наконец, третий слой предсказывает стоимость участка на основе результатов второго слоя, то есть основываясь на «сложных» извлеченных характеристиках участка. По сути, третий слой — это линейная модель на сложных признаках.

Чем больше слоев (глубина нейросети), тем сложнее характеристики объектов, которые распознает нейронная сеть, и тем более сложные задачи она может решать. Отсюда и название — глубинное обучение. Вместе с количеством слоев растет и число весов: современные нейросети могут включать миллионы и даже миллиарды параметров.

На практике понять, какие характеристики объектов распознает каждый слой, очень сложно. Иными словами, обучив нейронную сеть для задачи предсказания участка, мы можем получить отличное качество решения задачи, но мы не сможем объяснить и проинтерпретировать, по каким именно характеристикам нейросеть сделала предсказания.

Обучение нейронных сетей

Итак, мы обсудили, что нейросеть собирается из блоков (слоев) и на самом деле представляет собой сложную формулу, в которую входят признаки и веса. Признаки всегда даны — мы всегда знаем, для какого именно участка/клиента/магазина и др. хотим выполнить предсказание. Но хорошо работать нейронная сеть будет только с аккуратно настроенными весами — именно они «оживляют» формулу и позволяют ей делать качественные предсказания.

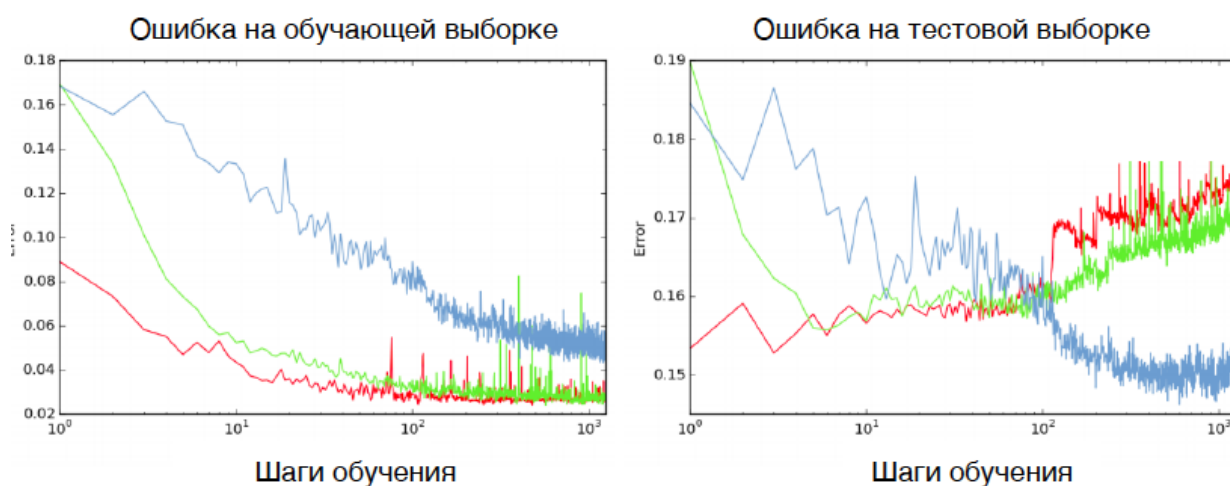
Градиентный спуск

Как и в линейных моделях, веса нейронных сетей настраиваются по обучающим данным. Основной принцип следующий: нужно найти такие веса, при использовании которых нейросеть будет делать правильные предсказания для объектов из обучающих данных (например, правильно предсказывает стоимости участков).

Чтобы их найти, используется алгоритм, называемый градиентным спуском.

Градиентный спуск работает так: в начале обучения все веса устанавливаются в произвольные значения: генерируется случайный набор чисел. А затем много раз повторяют шаг обновления весов: каждое число (каждый вес) немного меняют, чтобы ошибка предсказания для обучающих данных немного уменьшилась.

На каждом шаге ошибка уменьшается чуть-чуть (а на самом деле может и немного вырасти), но после повторения тысяч или миллионов таких обновлений весов ошибка приближается к нулю, то есть нейросеть начинает выполнять правильные предсказания для обучающих данных.



На графике приведен пример изменения ошибки в процессе обучения трех различных нейросетей в задаче регрессии: по оси x — шаги обновления весов, по оси y — ошибка.

На левом графике видно, что ошибка на обучающих данных немного колеблется, но глобально постепенно уменьшается.

Переобучение нейросетей

Благодаря тому, что нейронные сети могут выучивать очень сложные зависимости в данных, они очень легко переобучаются, то есть могут запоминать ответы на обучающих данных и при этом плохо работать на новых данных. Мы явно видим переобучение для красной и зеленой линий:

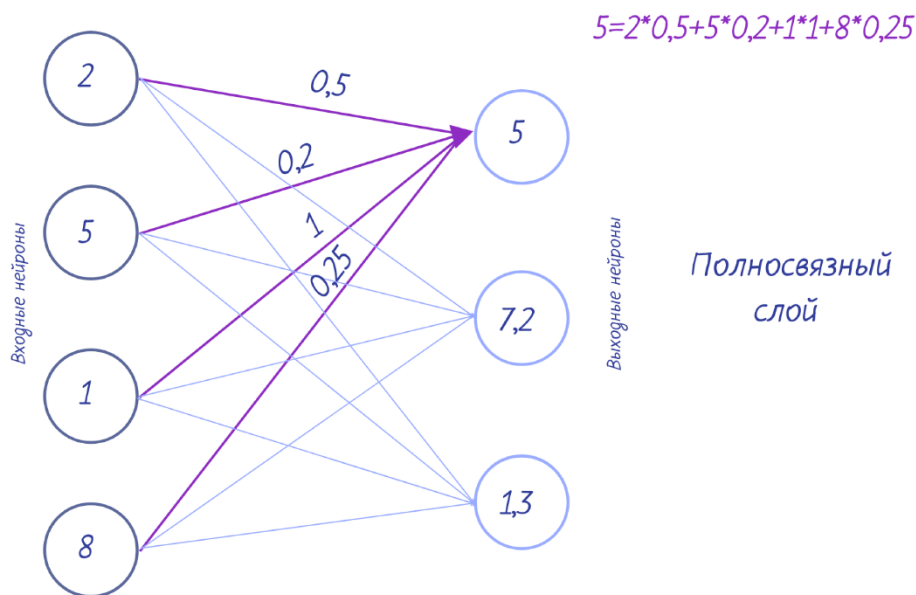
ошибка на тестовых данных растет, а не падает. А вот синяя линия соответствует качественно работающей нейросети.

Чтобы нейросети не переобучались и хорошо работали на новых данных, используют множество различных приемов — владение ими отличает опытного специалиста по глубинному обучению от новичка. А изучением и разработкой новых таких приемов занимается множество исследователей по всему миру. По-научному применение таких приемов называется **подбором гиперпараметров** — такой термин уже обсуждали в блоке про ансамбли. «Гипер» — потому что эти величины не настраиваются в процессе обучения по данным. А параметрами в нейронных сетях являются веса — они настраиваются по данным.

Полносвязный слой и слой нелинейности

Как было упомянуто ранее, слои бывают разные. Например, для обработки каждого типа данных (изображения, звук, тексты) придумывают свои подходящие виды слоев, мы поговорим об этом в следующих блоках. А сейчас разберемся с самыми простыми слоями — полносвязным слоем и слоем нелинейности.

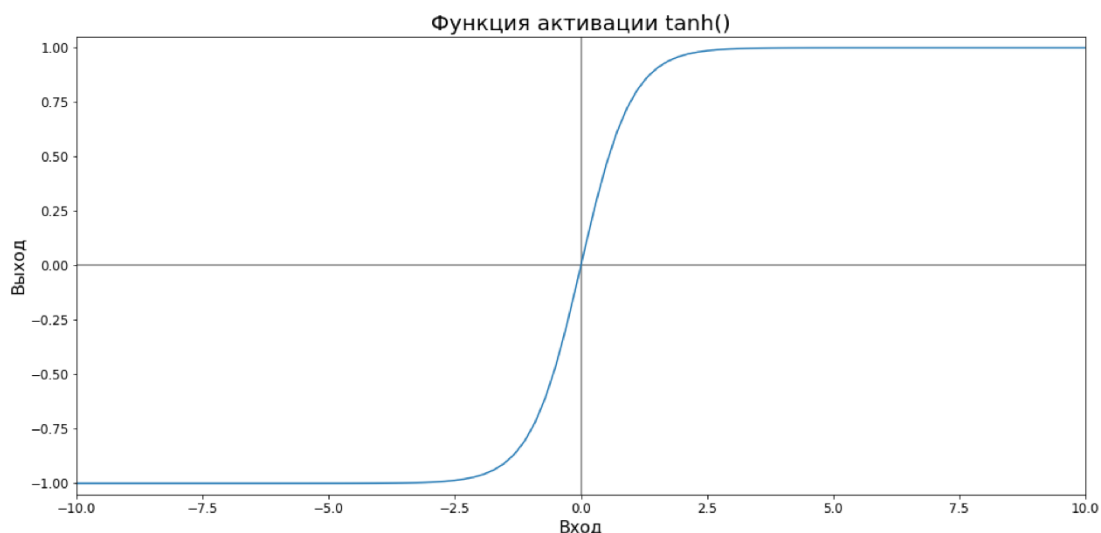
- Полносвязный слой (fully-connected layer, FC), или линейный — это, по сути, много линейных моделей. Обычно его изображают как два набора так называемых нейронов, попарно соединенных между собой (каждый соединен с каждым)



В каждом нейроне хранится число. Каждый нейрон справа (выходной нейрон) вычисляется как сумма всех нейронов слева (входных нейронов), умноженных на веса (линейная модель).

В иллюстративном примере с участком мы использовали именно полносвязные слои, и нейроны отвечали за характеристики участков (на практике сложно понять, какие именно характеристики выучила нейронная сеть). Число нейронов в слое называется **шириной слоя**. Полносвязные слои можно соединять друг с другом, тогда выходные нейроны предыдущего слоя становятся входными нейронами следующего слоя.

Однако, если мы будем просто соединять полносвязные слои, в результате получится линейная модель. Чтобы это исправить, достаточно небольшой детали: между полносвязными слоями вставлять так называемые слои нелинейности, их еще иногда называют активациями. Нелинейность преобразует нейрон согласно графику, например такому:



На практике чаще всего используют нелинейность под названием ReLU: у нее всего один излом.

Благодаря нелинейностям итоговая нейросеть распознает не линейные, а сложные зависимости.

- Если чередовать полносвязные слои и слои нелинейности, получится полносвязная нейронная сеть (Multi-LayerPerceptron, MLP) — простейшая нейросетевая архитектура. Нелинейность может иметь любую форму, главное, чтобы график не был прямой линией.

Нейрон, активация, персептрон

Полносвязная нейронная сеть была придумана американским ученым Фрэнком Розенблаттом в 1957 году как модель восприятия информации мозгом. Он считал, что линейные слои моделируют передачу и активацию сигналов между нейронами головного мозга, и назвал модель персептроном от латинского *perceptio* — восприятие.

Слово «активация» отсылает к тому, что каждый нейрон активируется при наличии некоторого сигнала, например, появление какого-то цвета в поле зрения. Позднее оказалось, что никакой связи между моделью и мозгом нет, но названия прижились и используются до сих пор.

Нет никакой связи с мозгом и у современных архитектур нейронных сетей: как мы обсудили выше, слои нейросетей придумывают по принципу

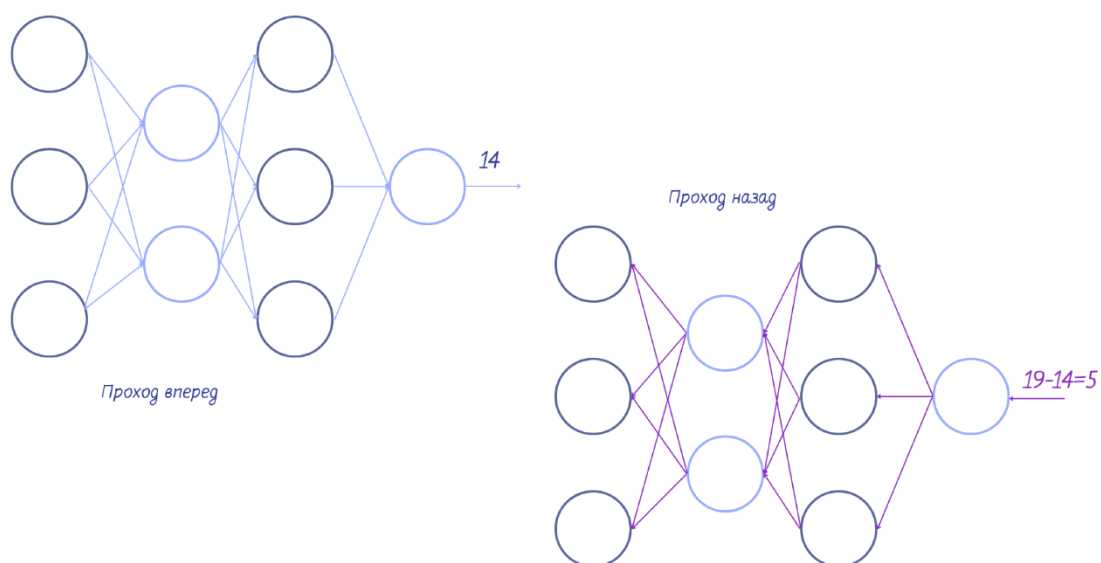
удобства обработки разных видов данных, а не по принципу повторения отдельных частей мозга.

Алгоритм обратного распространения ошибки

Ранее мы обсудили, что процесс обучения нейронной сети состоит из повторяющихся шагов обновления весов. Как же выполняются эти обновления? Рассмотрим на примере полносвязной нейронной сети для предсказания стоимости участка.

Чтобы выполнить предсказание стоимости какого-то участка, вычисляются значения всех нейронов последовательно слева направо (синие стрелки на иллюстрации). Пусть предсказана стоимость 14 тысяч долларов, а настоящая стоимость участка 19 тысяч долларов. Тогда ошибка нейросети равна пяти тысячам долларов.

Далее выполняется процесс, называемый **алгоритмом обратного распространения ошибки** (backpropagation). Этот алгоритм выясняет, как нужно изменить каждый вес, чтобы ошибка стала меньше пяти тысяч. Изменения весов вычисляются последовательно справа налево (красные стрелки на иллюстрации) — отсюда и прилагательное «обратное» в названии алгоритма.



Говоря математическим языком, алгоритм обратного распространения ошибки последовательно вычисляет производные функционала ошибки по всем весам нейронной сети.

Инструменты для обучения нейросетей

Для обучения нейронных сетей чаще всего используется язык программирования Python и его специальные библиотеки: PyTorch и TensorFlow. В этих библиотеках можно собирать архитектуры из любых видов слоев, а алгоритм обратного распространения ошибки уже встроен в эти слои.

Нейронные сети в задачах машинного обучения

Полносвязную нейронную сеть, которую мы обсудили, можно использовать в задачах регрессии и классификации при работе с табличными данными: входные нейроны первого слоя — это признаки (столбцы таблицы), а выходные нейроны последнего слоя предсказывают целевую переменную. В задаче классификации обычно предсказывают вероятности классов.

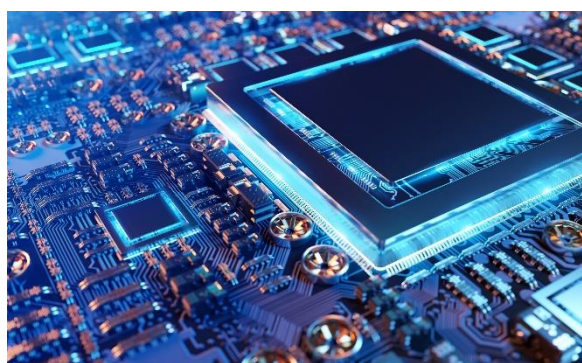
Качество работы нейронной сети в классификации сопоставимо с качеством бустинга или даже немного уступает, при этом время настройки

нейронной сети может быть больше. Поэтому для табличных данных чаще используют бустинг.



А вот классификация изображений (например, что за цветок на фотографии, доволен ли клиент обслуживанием) или текстов (например, категоризация документов) — это уже задача для нейронных сетей. Больше задач обработки этих типов данных мы обсудим в следующих блоках.

Нейронные сети используют и в других задачах машинного обучения. Например, сложные характеристики объектов, извлекаемые нейросетями, могут упростить кластеризацию или визуализацию данных. Используют нейросети и в обучении с подкреплением, чтобы предсказывать ожидаемые награды от выполнения действий в каких-то состояниях среды или для выбора следующих действий.



Обучение нейронных сетей вычислительно затратно: для этого используются специальные графические ускорители (GraphicsProcessingUnit, GPU). Кроме этого, для успешного обучения нейросети необходимо собрать много данных — без этого не получится обучить миллионы параметров нейросети. Появление мощной вычислительной аппаратуры, сбор большого

количества данных и разработка особых приемов обучения нейросетей — эти три фактора обусловили прорыв глубинного обучения, произошедший в начале 2010-х годов.

Подведем итог: нейронные сети идейно похожи на линейные модели, но обнаруживают гораздо более сложные зависимости в данных. Специалист задает архитектуру нейронной сети, то есть набор слоев, и гиперпараметры, регулирующие процесс обучения нейронной сети, а компьютер с помощью алгоритма градиентного спуска и алгоритма обратного распространения ошибки находит веса нейронной сети, используя обучающие данные.

Популярные архитектуры нейронных сетей

Нейронная сеть — это, по сути, формула, в которую входят признаки объекта и веса. В процессе обучения выполняется поиск таких весов, чтобы нейронная сеть не делала ошибок на объектах из обучающих данных. Нейронные сети распознают сложные характеристики входных объектов: чем глубже слой, тем выше уровень абстракции.

Нейронные сети для анализа изображений

Сверточные нейронные сети

Как и все методы машинного обучения, нейронные сети работают с числовыми данными. Изображение — это тоже набор чисел: оно состоит из пикселей, и цвет пикселя задается набором чисел. Если изображение черно-белое, то каждый пиксель задается одним числом от 0 до 255: 0 — черный цвет, 255 — белый цвет, посередине — серый цвет. В цветных изображениях цвет пикселя задается несколькими числами, например в цветовой схеме RGB (red, green, blue) — три числа, отвечающих за интенсивность красного, зеленого и синего цветов.

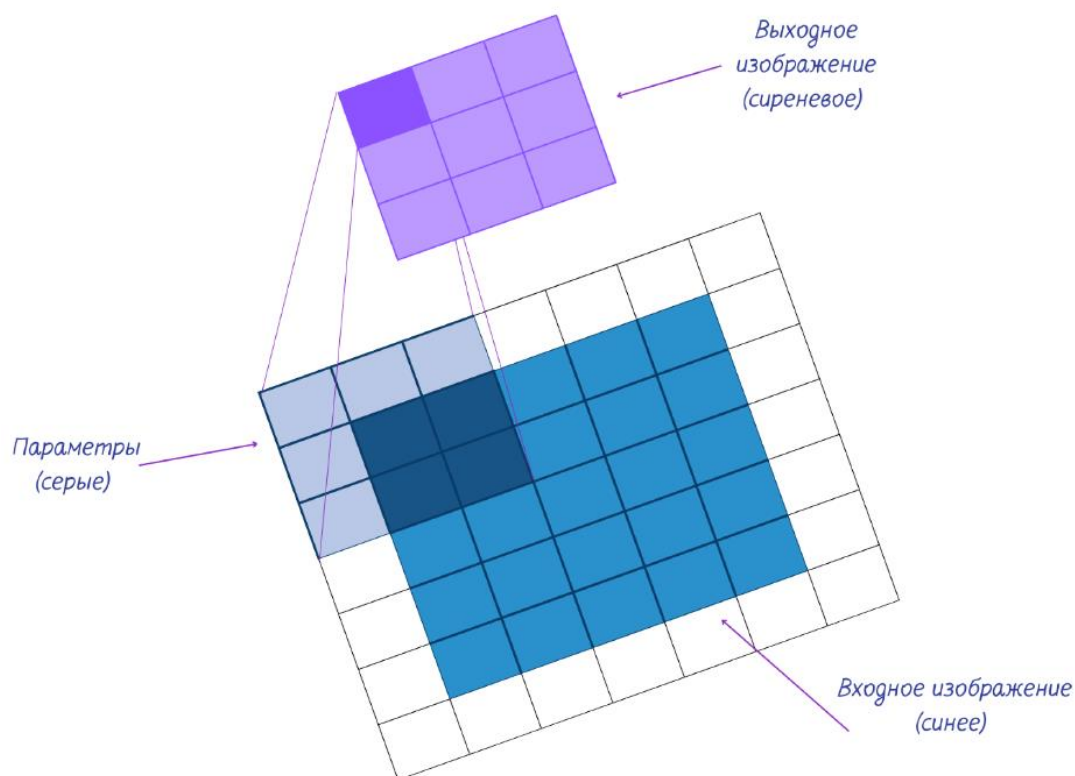


62	62	63	64	65	66	67	68	69	70	71	72	72	73	73	73	72	72	71	70	69	67	66	66	65	63	62	61	60	60			
61	62	63	64	66	66	67	68	68	69	70	71	72	73	73	72	72	71	71	70	69	68	66	66	65	65	63	62	61	60	60		
61	62	63	64	66	66	68	68	69	70	71	72	73	73	72	72	71	71	69	68	67	66	66	65	64	63	62	61	61	60	60		
61	63	64	64	66	67	68	68	69	70	71	72	73	74	73	73	73	71	70	69	68	66	66	65	64	63	62	61	61	60	60		
61	63	64	65	67	68	69	69	70	70	71	72	75	53	69	72	72	71	71	70	69	68	67	66	65	64	63	62	60	60	60		
63	64	65	66	67	68	69	69	70	70	71	72	4	5	11	48	72	71	71	69	68	67	66	65	64	62	62	60	59	59	59		
63	65	66	66	68	68	69	70	71	71	72	18	4	4	7	8	66	71	70	69	68	67	66	65	64	63	61	59	59	58	58		
63	65	67	68	69	69	70	71	72	64	4	27	24	54	33	29	52	64	68	67	66	65	64	63	62	61	60	59	58	58	58		
64	65	66	66	68	69	70	71	71	72	24	12	17	24	48	60	37	43	36	52	66	68	67	66	65	64	63	61	60	59	58		
65	66	67	67	68	69	71	70	6	6	6	5	34	36	12	47	34	17	29	54	43	63	67	66	65	64	63	62	60	59	58		
64	65	66	66	68	69	71	6	6	5	5	7	16	19	4	47	44	27	24	43	67	66	65	65	64	63	61	60	59	58	57		
63	64	65	65	67	69	6	6	5	5	5	6	8	9	20	27	51	78	41	44	66	65	65	65	64	63	62	60	59	58	57		
63	64	65	65	5	5	5	5	5	5	5	4	19	6	7	54	64	20	59	65	65	64	64	63	62	61	60	59	57	56	56		
63	64	65	65	14	5	6	5	5	4	5	4	18	7	5	4	19	10	11	65	64	64	63	61	66	62	61	60	59	58	56		
63	64	64	65	53	7	4	5	6	7	10	6	5	5	4	21	24	18	64	64	63	62	64	65	62	60	59	58	57	56	56		
64	64	64	64	65	50	4	4	4	5	11	16	6	6	4	35	16	26	66	64	63	61	72	67	63	62	61	59	58	57	56		
64	64	64	64	65	46	4	4	4	5	6	9	8	5	29	10	43	56	29	57	64	64	63	61	70	67	62	61	60	59	57		
64	64	64	65	66	77	5	4	4	5	6	6	6	18	66	20	57	60	46	36	75	70	62	61	70	67	62	61	60	59	58		
49	50	62	65	57	5	5	6	6	6	6	6	6	4	59	26	60	58	44	22	63	71	72	60	69	68	61	60	58	59	58		
42	52	57	52	26	5	5	5	5	5	5	5	5	70	50	43	61	62	64	59	42	64	60	62	56	63	65	65	67	61	53	53	
32	32	32	33	6	5	5	5	5	5	5	6	6	11	39	21	33	51	50	45	46	18	32	36	33	23	44	70	71	51	42	27	31
50	50	51	39	5	5	5	5	5	5	6	6	42	69	28	34	42	39	43	37	26	29	40	26	29	26	35	42	35	33	18	16	
52	53	51	22	5	5	5	5	5	5	6	6	5	44	56	17	51	54	53	54	56	51	22	54	54	55	54	53	53	53	52	52	
54	54	53	8	5	5	5	5	6	5	6	5	13	52	42	21	51	54	51	49	49	50	22	41	45	42	42	40	41	44	43	42	
52	52	54	36	8	5	5	6	6	5	6	28	55	32	32	54	53	51	51	51	51	44	25	51	49	49	50	49	48	46	46	46	
54	52	53	30	7	5	6	6	5	6	40	54	29	52	51	53	56	55	52	52	51	38	52	52	50	49	46	46	45	46	47	47	
51	52	51	53	27	14	5	4	5	4	7	47	51	21	39	49	47	49	52	52	49	35	31	48	47	47	47	47	47	46	46	43	
48	50	51	53	25	14	17	8	4	4	17	46	40	15	43	47	46	49	52	54	53	53	54	14	50	49	46	47	47	47	47	45	
49	49	49	49	22	12	20	24	6	14	35	51	30	48	48	50	51	51	49	51	51	52	50	41	58	48	47	47	47	45	46	46	
51	49	50	50	22	13	19	36	13	12	42	50	46	73	50	50	50	49	48	49	48	49	45	51	46	44	44	42	45	47	47	47	
47	49	49	47	20	16	26	22	21	15	36	48	42	61	47	48	51	47	50	51	51	51	49	47	52	47	47	44	43	45	46	46	
48	50	48	52	19	13	33	36	18	18	36	49	51	54	47	47	49	46	46	49	49	49	47	44	53	44	48	44	46	46	45	45	

Каждый пиксель -- это число, задающее уровень яркости.

Нейронные сети довольно легко адаптировать для анализа различных видов данных: для этого используют специальные слои, или блоки, из которых строится нейронная сеть. Для анализа изображений широко используются так называемые сверточные слои.

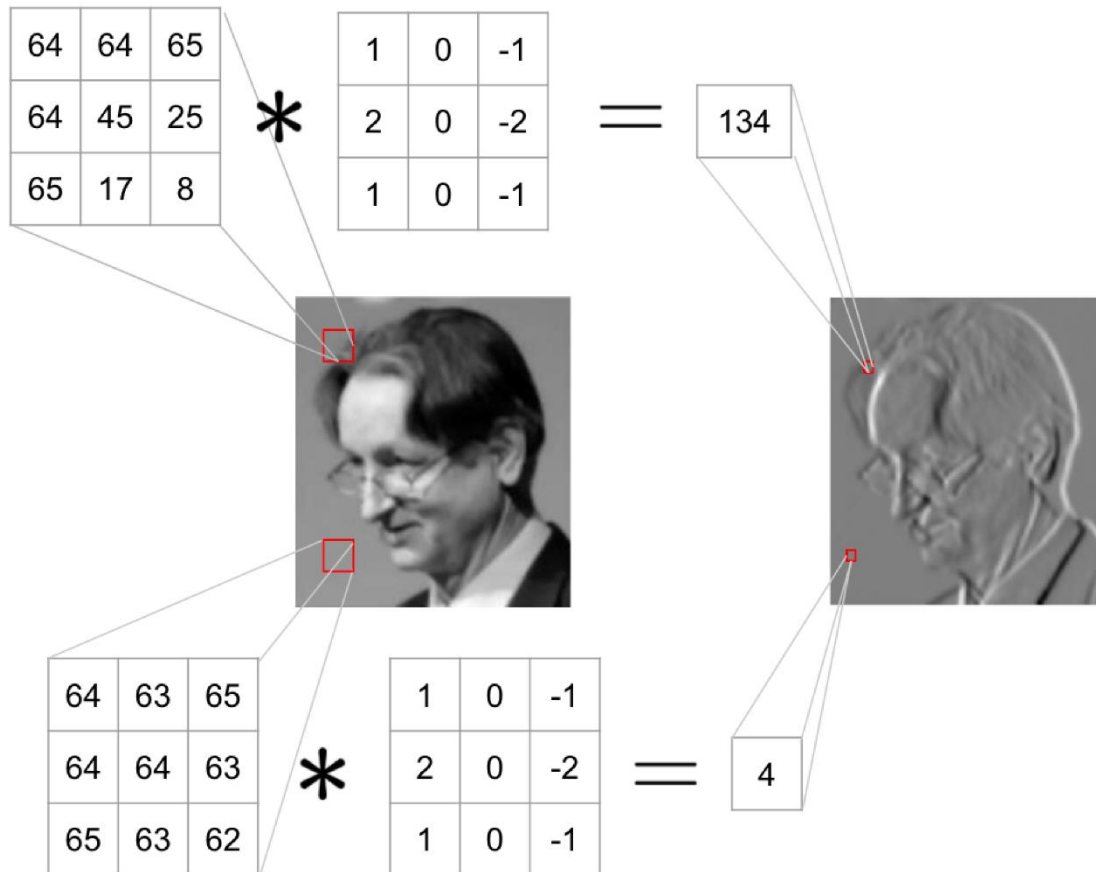
Свертка — это процедура агрегации информации о пикселе и соседних с ним пикселях.



Пример работы фильтра Собеля

Например, возьмем фильтр размера 3×3 , с нулевым вторым столбцом и противоположными значениями в левом и правом столбце. Приложим его к двум различным частям изображения: одна относится к фону, а другая — к границе фона и волос.

При применении такой свертки к фрагменту фона изображения получится $1 \times 64 - 1 \times 65 + 2 \times 64 - 2 \times 63 + 1 \times 65 - 1 \times 62 = 4$. Поскольку все пиксели на фоне примерно одинаковые, а числа в фильтре суммируются в ноль, значение свертки получается приблизительно равным нулю. При применении описанной свертки к фрагменту на границе фона и волос получится $1 \times 64 - 1 \times 65 + 2 \times 64 - 2 \times 25 + 1 \times 65 - 1 \times 8 = 134$.



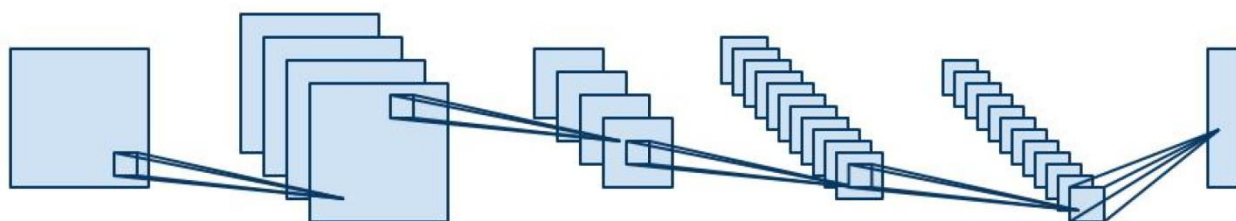
Поскольку между пикселями фона и волос происходит резкий перепад яркости, значение свертки оказывается далеким от нуля. В итоге мы получили, что описанный фильтр позволяет выделять границы объектов: в однородных зонах значение свертки близко к нулю, в зона перехода —

далеко от нуля. Такой фильтр называется фильтром Собеля. Подробнее прочитать про устройство свертки можно [здесь](#), а про различные виды сверток — [здесь](#).

Свертки были придуманы для анализа и обработки изображений, например, они используются в графических редакторах наподобие Photoshop для наложения фильтров — собственно, наложение фильтра на изображение в прямом смысле означает применение свертки. При этом в графических редакторах используются заранее заданные фильтры, например, разобранный выше фильтр Собеля.

Конструкция сверточнойнейросети

В нейронных сетях фильтры — это настраиваемые параметры. Именно числа, записанные в фильтрах, являются весами сверточного слоя нейронной сети, и эти фильтры настраиваются в процессе обучения по данным. Один сверточный слой состоит из нескольких сверток, и сверточные слои можно ставить друг за другом, по аналогии с полносвязными слоями. Первый сверточный слой применяется непосредственно к самому изображению, второй слой — к выходу первого сверточного слоя и т. д. Выход сверточного слоя формально тоже является изображением, но на глубоких слоях нейронной сети это «изображение» уже не будет интерпретироваться человеком. Между сверточными слоями, как и между полносвязными, вставляют слои нелинейности, а в конце сверточной архитектуры обычно вставляют один или несколько полносвязных слоев.



Пример сверточной нейронной сети, состоящей из четырех сверточных слоев и одного полносвязного слоя

Как и в других видах нейронных сетей, в сверточныхнейросетях при увеличении номера сверточного слоя повышается уровень абстракции. Первые слои распознают простые переливы яркости и отдельные цвета, слои чуть глубже распознают простые геометрические формы, еще более глубокие слои распознают части изображений, например глаза, губы и нос при анализе лиц, а самые глубокие слои отвечают за распознавание целых объектов.



MIT: Alexander Amini, 2018 intro.todeeplearning.com

Задачи анализа изображений

Задача классификации изображений

Самая популярная задача анализа изображений — задача классификации. В этой задаче на вход нейронной сети подается изображение, а выход — это метка класса. В качестве примеров можно привести:

- идентификацию владельца смартфона по фотографии (классы: владелец или не владелец);
- распознавание эмоций клиента при посещении офиса (классы: доволен, возмущен, нейтрален и т. д.);
- распознавание дорожных знаков в беспилотных автомобилях (классы: знаки).

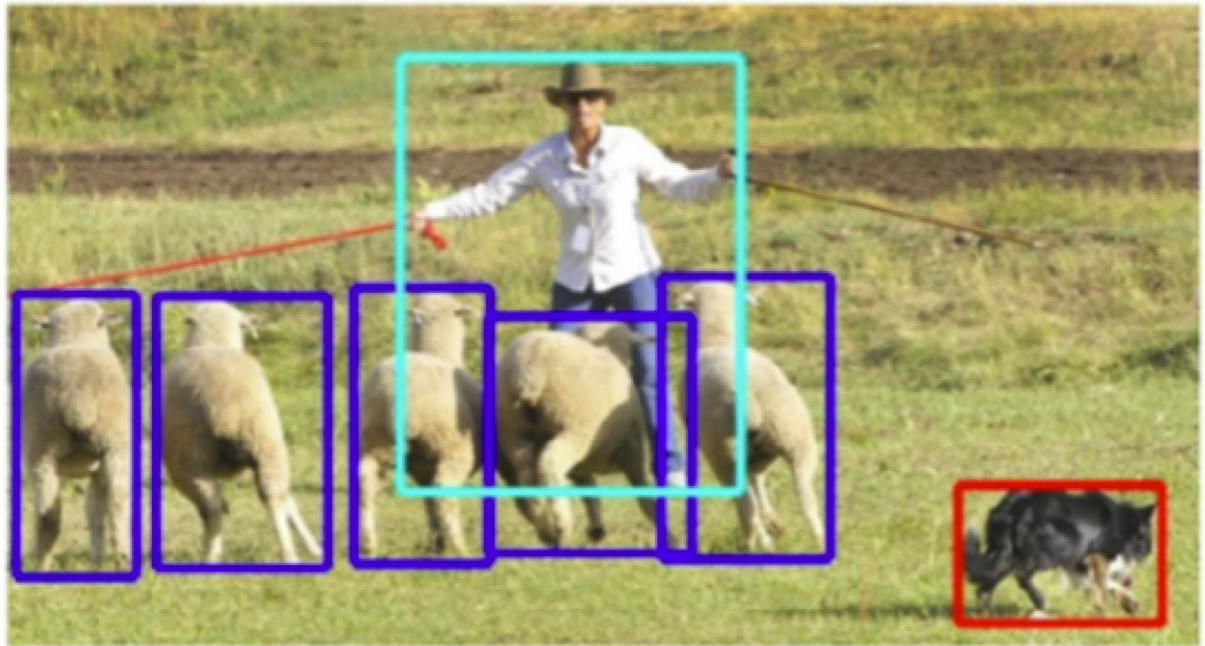
Для обучения классифицирующей нейронной сети потребуется большой набор данных (десятки или сотни тысяч объектов), состоящий из пар (изображение, класс). Например, для распознавания эмоций потребуется множество фотографий, для которых известна настоящая эмоция клиента.



Классификация изображения

Задача детекции изображений

Однако классификация — это несколько упрощенная задача: она лишь определяет, какие классы есть на изображении, но не определяет их расположение. Часто также решают задачу детекции: в ней нейронная сеть должна выделить прямоугольниками все объекты, находящиеся на изображении. Как и классификация, детекция — это задача обучения с учителем: для обучения нейронной сети потребуется набор изображений с объектами, выделенными прямоугольниками. Выполнять такую разметку может быть утомительно, часто для этого используют краудсорсинговые сервисы, например Яндекс.Толока.



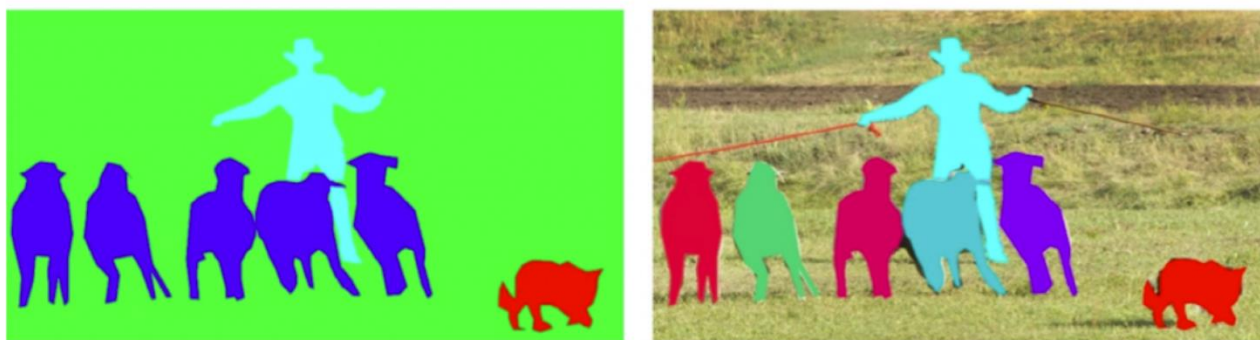
Детекция

В 2018 году компания Amazon открыла магазины Amazon.Go: в них покупатели берут товары с полок и выходят из магазина, не проходя через кассу, а деньги списываются автоматически через приложение. Камеры в магазинах снимают видео покупателей, и далее по этому видео решается задача детекции: нейросеть выделяет товары, которые выбрал покупатель, и распознает вид товаров.

Детекцию по видеокамерам активно применяют и в обычных магазинах, например, для подсчета количества товаров, оставшихся на полке. Нейросеть анализирует фотографию полки, выделяет все единицы товаров, считает их, и, если товара осталось мало, компьютер отправляет сигнал на склад, чтобы ассортимент товара пополнили. X5 RetailGroup в десятки раз ускорили контроль планограмм, сократили на 10% количество людей, уходящих из магазина без покупок, и на 20% — потери магазинов с помощью системы видеоаналитики «Всевидящее око», работающей на основе нейронных сетей.

Задача сегментации изображений

Если нужно проанализировать изображение еще более детально, решают задачу сегментации. В этой задаче нейронная сеть должна для каждого пикселя определить, к какой области он относится (области — это фон и объекты, находящиеся на изображении). Собрать размеченные данные в задаче сегментации еще сложнее, чем в задаче детекции. Сегментация активно применяется в обработке изображений и для составления коллажей: например, можно «вырезать» человека (выделив все пиксели, отнесенные к области «человек») и вставить на другой фон.



Семантическая сегментация

Мы рассмотрели три самых распространенных задачи анализа изображений, однако нейронные сети можно использовать для решения самых разных задач. Современные нейронные сети умеют хорошо понимать содержание изображения и анализировать его детали. Например, многие современные онлайн-магазины предлагают покупателям загрузить фотографию понравившегося комплекта одежды, а программа подберет похожие вещи из ассортимента магазина. По сути, при этом решается задача понижения размерности, о которой мы говорили в курсе ранее: для всех товаров магазина и для фотографии, загруженной пользователем, нейросеть определяет высокоуровневые признаки, по соответствию которым далее подбираются товары.

Задача генерации изображений

Мы обсудили, что современные нейронные сети умеют отлично понимать, что изображено на фотографии или видео. Однако за последние несколько лет произошел прорыв в решении обратной задачи — генерации

фотореалистичных изображений и видео. Как и в анализе изображений, в генерации изображений используются сверточные нейронные сети. На вход нейросети подаются характеристики изображения, которое нужно сгенерировать, а на выход нейросеть выдает новое изображение.

В 2016 году высокую популярность приобрели приложения, стилизующие изображения под известные картины, например Призма.



Научно данная задача называется переносом стиля.

Перенос стиля работает так: на вход нейросети подаются два изображения — то, которое нужно перерисовать, и то, с которого нужно скопировать стиль, а на выход выдается стилизованное изображение.

С помощью переноса стиля можно изменить время дня на фотографии.



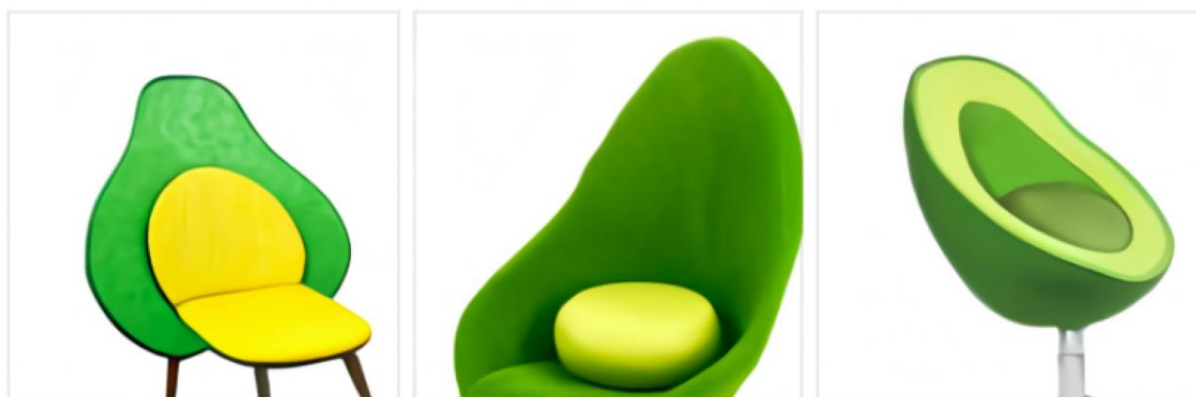
Чтобы генерировать фотореалистичные изображения, используют так называемые генеративно-сопоставительные сети (generative adversarial networks, GAN).

GAN состоит из двух нейросетей: первая генерирует изображение, а вторая решает задачу классификации и учится определять, на входе настоящее оно или сгенерированное. Задача первой нейросети — "обмануть" вторую, и она "вынуждена" генерировать реалистичные кадры.



Примеры лиц знаменитостей, сгенерированных нейросетями.

В начале 2021 года компания OpenAI представила нейронную сеть DALL·E, которая генерирует изображения по текстовому описанию. [Подробнее по ссылке](#). Например, DALL·E может создавать новые предметы интерьера или даже целые интерьеры. Ниже - примеры, сгенерированные нейросетью по запросу «кресло в форме авокадо»:



"Николай Иронов"

А студия Артемия Лебедева, например, целый год разрабатывала логотипы для реальных заказчиков с помощью искусственного интеллекта Николая Иронова. Клиенты и коллеги «Николая» из студии Лебедева не знали, что работы создаются генеративной моделью. За год модель выполнила более 20 коммерческих проектов.

При этом в Николая заложено много экспертных знаний: о том, как строить композицию, какие цвета комплементарные, как можно трансформировать шрифты. Подробнее об устройстве «искусственного дизайнерского интеллекта» можно почитать по ссылке.

Множество интернет-сервисов позволяют в интерактивном режиме познакомиться с нейросетями, генерирующими или анализирующими изображение. К примеру, в англоязычном сервисе <https://quickdraw.withgoogle.com/> пользователю предлагается нарисовать рисунок, а нейросеть попытается угадать, что нарисовано. Сервис <https://deepart.io/> стилизует изображения пользователя под известные картины. А сервис <https://affinelayer.com/pixsrv/> превращает набросок в фотореалистичное изображение.

Для работы с изображениями чаще всего используются сверточные нейронные сети. Три самых популярных задачи анализа изображений — классификация, детекция и сегментация. Помимо качественного анализа существующих изображений, нейросети сами могут генерировать новые фотореалистичные изображения. Благодаря гибкости нейронных сетей их можно использовать для решения самых разных задач, в том числе в неклассических постановках.

Нейросети для работы с текстом

Анализ текстов

Нейронные сети отлично подходят для обработки данных, в которых есть некоторая внутренняя структура. Применение нейронных сетей для анализа еще одного такого типа данных — текстов.

В качестве простого примера задачи анализа текстов можно привести классификацию текстов, например, определение темы документа или определение эмоционального окраса отзыва клиента. Ниже мы рассмотрим и более сложные задачи.

Сама область обработки текстов по-английски называется NaturalLanguageprocessing (NLP). NLP охватывает все методы обработки текстов, включая, например, линейные модели для анализа текстов.

Однако большой прорыв в этой области произошел с приходом нейронных сетей, поэтому сегодня NLP во многом ассоциируется именно с нейросетевыми подходами.

Предобработка текстов

Перед тем, как обучать алгоритм на данных, их необходимо подготовить к обработке: очистить и собрать словарь, который нейросеть потом будет использовать для перевода текста на "машинный" язык.

Очистка данных

Например, данные, скачанные из социальных сетей, могут содержать ссылки, хэштеги, упоминания других пользователей (@username), а тексты, скачанные из интернета, — остатки разметки. Иногда также выполняют удаление пунктуации и приведение всех слов к нижнему регистру, однако это может быть полезно не во всех задачах. Например, в задаче определения эмоционального окраса отзыва использование большого количества заглавных букв может указывать на недовольство клиента.

Сборка словаря

После очистки текстов выполняется сборка словаря: для каждого слова вычисляется количество раз, сколько оно встретилось в текстах, и словарь составляется из наиболее часто встречаемых слов, обычно 50 тысяч или 100 тысяч.

Редкие слова заменяются на слово «Пропуск»: если слово встретилась всего несколько раз во всем наборе данных, нейросеть не сможет хорошо выучить, что оно означает. Иногда, чтобы не терять информацию о редких словах, их делят на части, например, слово «эмбеддинг» можно разделить так: «эм», «бе», «д», «д», «инг». Такой подход называется BytePairEncoding (BPE).

В результате предобработки каждый текст представляется в виде последовательности номеров слов в словаре.

Лето закончится через девятнадцать дней

2 6 5 7 5

Словарь:

я	1
лето	2
мы	3
через	4
дней	5
закончится	6
девятнадцать	7

Векторные представления слов

Нейронные сети для анализа текстов обычно состоят из двух компонент: первая компонента обрабатывает лексику (распознает смысл слов), а вторая обрабатывает структуру (извлекает зависимости из взаимного расположения слов).

Начнем с первой компоненты: она называется слоем векторных представлений, англ. Embeddinglayer. По-русски векторные представления тоже часто называют эмбедингами.

Для каждого слова в словаре слой извлекает набор чисел фиксированного размера, например 256 чисел — этот набор и называется векторным представлением, или эмбедингом.

У разных слов — разные векторные представления. При этом часто получается, что у слов, близких по смыслу, векторные представления похожи. Каждое число в эмбединге соответствует какому-то «смыслу».

Пусть у нас есть отзыв «Отличное, замечательное мобильное приложение». Предположим, что первое число в эмбединге (первая координата вектора) соответствуют смыслу «является прилагательным», второе — смыслу «дает положительную оценку», а третье — смыслу «связано со смартфонами».

Отличное

Замечательное

Мобильное

Приложение

1
0,9
0

1
1
0

1
0
1

0
0
0,8

Эмбединги

Первое слово «отличное» является прилагательным и дает положительную оценку (почти идеальную), но не связано со смартфонами, поэтому ему соответствует эмбединг

$[1, 0.9, 0]$

. Третье слово «мобильное» является прилагательным и связано со смартфонами, но не связано с положительными оценками, — его эмбединг

$[1, 0, 1]$

. Аналогично получаются эмбединги двух других слов.

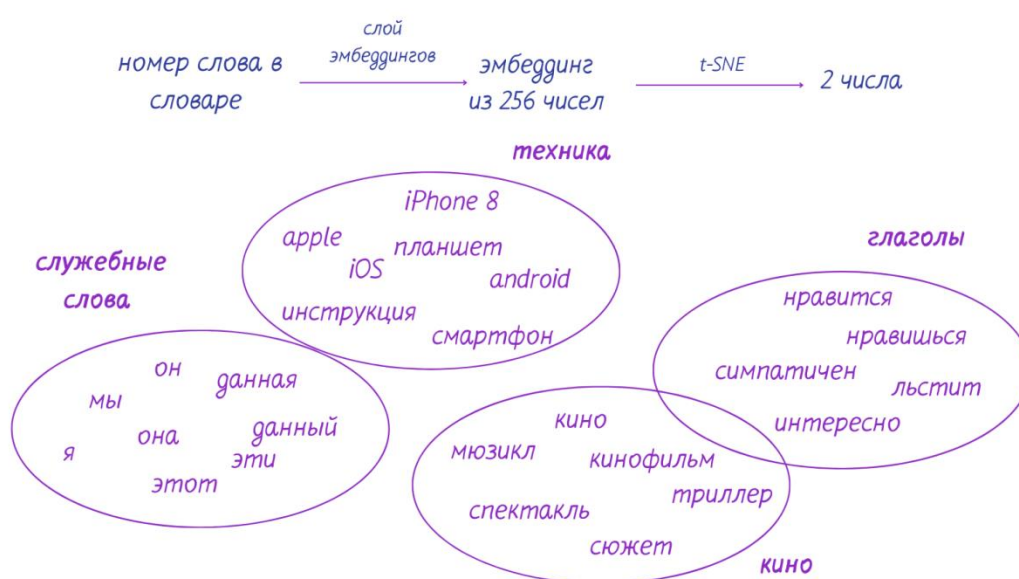
В итоге у слов «замечательное» и «приложение» получились совсем разные эмбединги, а у слов «замечательное» и «отличное» — наоборот, очень похожие, потому что они близки по смыслу.

Эмбединг на практике

На практике человек не придумывает, какому смыслу соответствует каждое число в эмбединге — эмбединги слов компьютер учит сам. Числа от 0 до 1 использованы для примера, в реальности они могут быть и положительными, и отрицательными. Как и с другими слоями нейронных

сетей, как правило, мы не можем проинтерпретировать, за какие конкретно смыслы отвечает каждое число в эмбединге, но компьютер сам их «понимает» и благодаря эмбедингам распознает значение слов.

Сами числа — это обучаемые параметры, или веса, они настраиваются по обучающим данным вместе со всеми остальными весами нейронной сети. После обучения к этим эмбедингам можно применить методы понижения размерности, например t-SNE, и визуализировать их. Каждому слову тогда будет соответствовать точка на графике.



Пример визуализации для задачи анализа эмоционального окраса отзывов

Овалами были вручную выделены группы похожих слов, а в целом визуализация показывает, что у похожих по смыслу слов — похожие эмбединги.

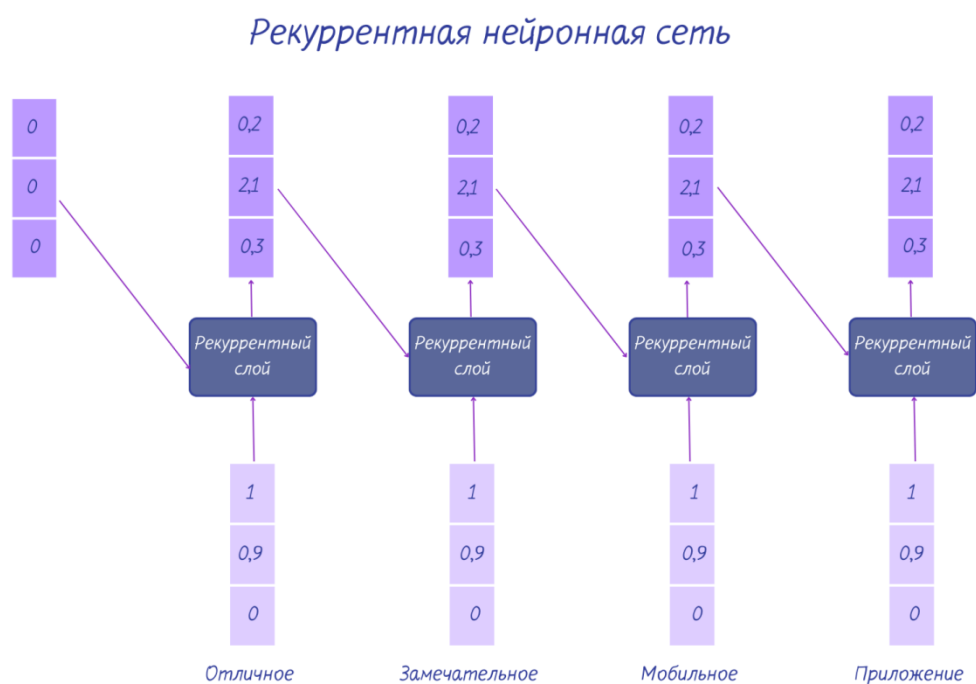
Архитектуры нейросетей для анализа текстов

После слоя эмбедингов идет слой, отвечающий за обработку взаимного расположения слов в тексте — обычно используется рекуррентный слой или архитектура Трансформер. Каждый из этих слоев берет на вход эмбединги слов и выдает на выходе новые векторные представления слов, эти новые представления часто называют скрытыми

состояниями. По сути, это новые наборы чисел для каждого слова, которые учитывают взаимосвязи слов.

Основная идея рекуррентного слоя — повторение одной и той же процедуры обработки для каждого слова.

Внутри рекуррентный слой устроен как полносвязный слой: в качестве входных нейронов выступают эмбединг слова и предыдущее скрытое состояние, в качестве выходных нейронов — новое скрытое состояние. Самое первое скрытое состояние обычно делают состоящим из нулей. Рекуррентных слоев можно использовать несколько, один за другим. В результате получится рекуррентная нейронная сеть (RecurrentNeuralNetwork, RNN).



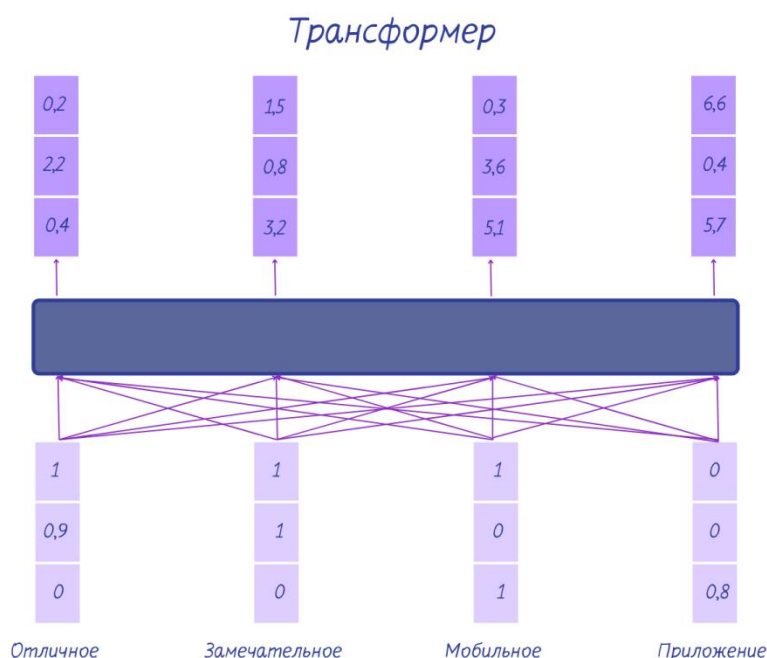
Преимущество рекуррентных слоев по сравнению с обычными полносвязными состоит в том, что они могут обрабатывать последовательности произвольной длины: сколько слов в предложении, столько раз повторяется слой. Из недостатков — слабая связь между словами, расположенными далеко друг от друга в тексте: в рекуррентной нейросети сигнал передается по цепочке от слова к следующему слову, и давно прочитанные слова могут «забываться».

Трансформер

Недостатки рекуррентных нейросетей, отсутствие связи между далекими словами и забывание слов, были исправлены в другой архитектуре, называемой Трансформер.

В Трансформере новое скрытое состояние для слова вычисляется на основе векторных представлений всех остальных слов.

В итоге каждое слово связано с каждым и проблем с учетом «длинных» зависимостей нет. Веса в Трансформере «спрятаны» в попарных взаимосвязях слов: они помогают определить, какой вклад каждое слово вносит в каждое другое слово. Обычно Трансформер состоит из четырех или шести слоев, каждый из которых включает попарные связи.



На практике Трансформер обычно работает лучше, поэтому он более распространен в прикладных проектах.

Задачи анализа и генерации текстов

Нейронные сети позволили решить множество интеллектуальных задач анализа текстов, которые не удавалось решить классическими методами. Эти задачи связаны с пониманием смысла или структуры текста и написанием новых текстов.

Более простыми считаются задачи, в которых текст выступает в качестве входных данных — такие задачи называют **дискриминативными**. Классические примеры — задача классификации текстов или задача регрессии.



Например, можно по тексту объявления о вакансии предсказывать заработную плату — такие алгоритмы используют на сайтах поиска вакансий.



Другой пример — задача модерации сообщений в социальных сетях или на сайтах объявлений: в ней нужно обнаруживать сообщения, написанные в грубой или оскорбительной форме. Например, Facebook и Instagram с 2016 года используют нейросети для модерации комментариев и рекомендации постов.



Отдельно выделяют задачи тегирования: в них нужно выполнить отдельное предсказание для каждого слова в тексте. К этой группе относятся задача определения частей речи для каждого слова и задача определения синтаксических ролей в предложении — такие алгоритмы помогают, в частности, выполнять автокоррекцию текста.



Еще один интересный пример — задача выделения именованных сущностей. Она состоит в том, чтобы найти в тексте города, компании, персоналии и другие объекты реального мира. Их выделение помогает совершенствовать поиск по Интернету.

Противоположными дискриминативным считаются **генеративные задачи**: в них текст должна сгенерировать нейронная сеть, то есть текст — это выходные данные. Подробно генерация текстов разбирается в видеокейсе. Попробовать сгенерировать текст с помощью нейронной сети можно, используя сервис Порфирьевич. А в этом сервисе можно загрузить фотографию, и нейросеть сгенерирует к ней подпись.

Сказать, что нейросети достигли полного понимания текстов, написанных людьми, и способны генерировать абсолютно качественные тексты самостоятельно, пока нельзя, однако существенный прорыв в области NLP уже произошел. Собрать и обработать текстовые данные, как правило, проще, чем, например, данные изображений, и нейросети, анализирующие тексты, сегодня активно применяются в различных областях бизнеса (юриспруденция, журналистика, маркетинг), науки, культуры.

Seq2seq

Отдельное место в NLP занимают задачи, в которых и входные, и выходные данные являются текстами: по-английски соответствующий раздел называется seq2seq (последовательность в последовательность). Иными словами, нейросеть читает текст и на его основе генерирует новый текст.

По сути, любая задача, в которой данные представляются как набор пар (входной текст, выходной текст), является задачей seq2seq. Например, можно собрать набор пар (литературное произведение, школьное сочинение по нему) и научить нейросети писать сочинения. Приведем другие примеры.



Стандартный пример такой задачи — машинный перевод: по тексту на одном языке нужно сгенерировать текст на другом языке. Сегодня эта задача решается с очень высоким уровнем качества благодаря использованию архитектуры Трансформер и обучению на больших массивах данных.



Еще один интересный пример задачи seq2seq — задача суммаризации текста. В ней на вход нейросети поступает длинный текст, а на выходе нейросеть выдает краткий пересказ текста. В этой задаче нейросети пока преуспели меньше, чем в машинном переводе. В упрощенной постановке методы суммаризации можно использовать для генерации, например, новостных заголовков или темы письма.



Как и с нейросетями для изображения, важно понимать, что каждая нейросеть для анализа текстов решает ровно одну задачу. Сложные же системы собираются из отдельных нейронных сетей и программных блоков. Например, нейронные сети положены в основу современных голосовых помощников.

Голосовым помощником решается целый ряд отдельных задач:

1. Перевод звука в текст (задача seq2seq).
2. Распознавание запроса по тексту (задача классификации с классами: «поставить будильник», «установить напоминание», «открыть приложение», «свободное общение» и т. д., при необходимости извлекаются дополнительные атрибуты: время будильника, название приложения и др.).
3. Для заранее определенных действий (например, установка будильника) запускается соответствующая программа и выбирается ответ из набора заранее заданных ответов, в случае выбора опции «свободное общение» используется модуль генерации ответа (задача генерации текстов).
4. Озвучка текста (задача генерации звука).

Подведем итог: нейросети для анализа текстов обычно состоят из двух компонент: слой эмбедингов, отвечающий за лексику, и либо рекуррентный слой, либо Трансформер, отвечающие за распознавание структуры текста. Задачи обработки текстов подразделяются на дискриминативные (текст на входе нейросети), генеративные (текст на выходе нейросети) и смешанные (seq2seq, текст и на входе, и на выходе). При этом задачи seq2seq предполагают наличие параллельного набора данных, то есть состоящего из пар (входной текст, выходной текст).

Прочитать об успехах нейронных сетей в 2020 году можно в обзоре.