

Data science

# Лекция 6. Глубокое обучение и нейросети

2025/2026 учебный год

Доцент кафедры МО&МО, Махно В.В.

©Создано при помощи <https://sberuniversity.ru/>

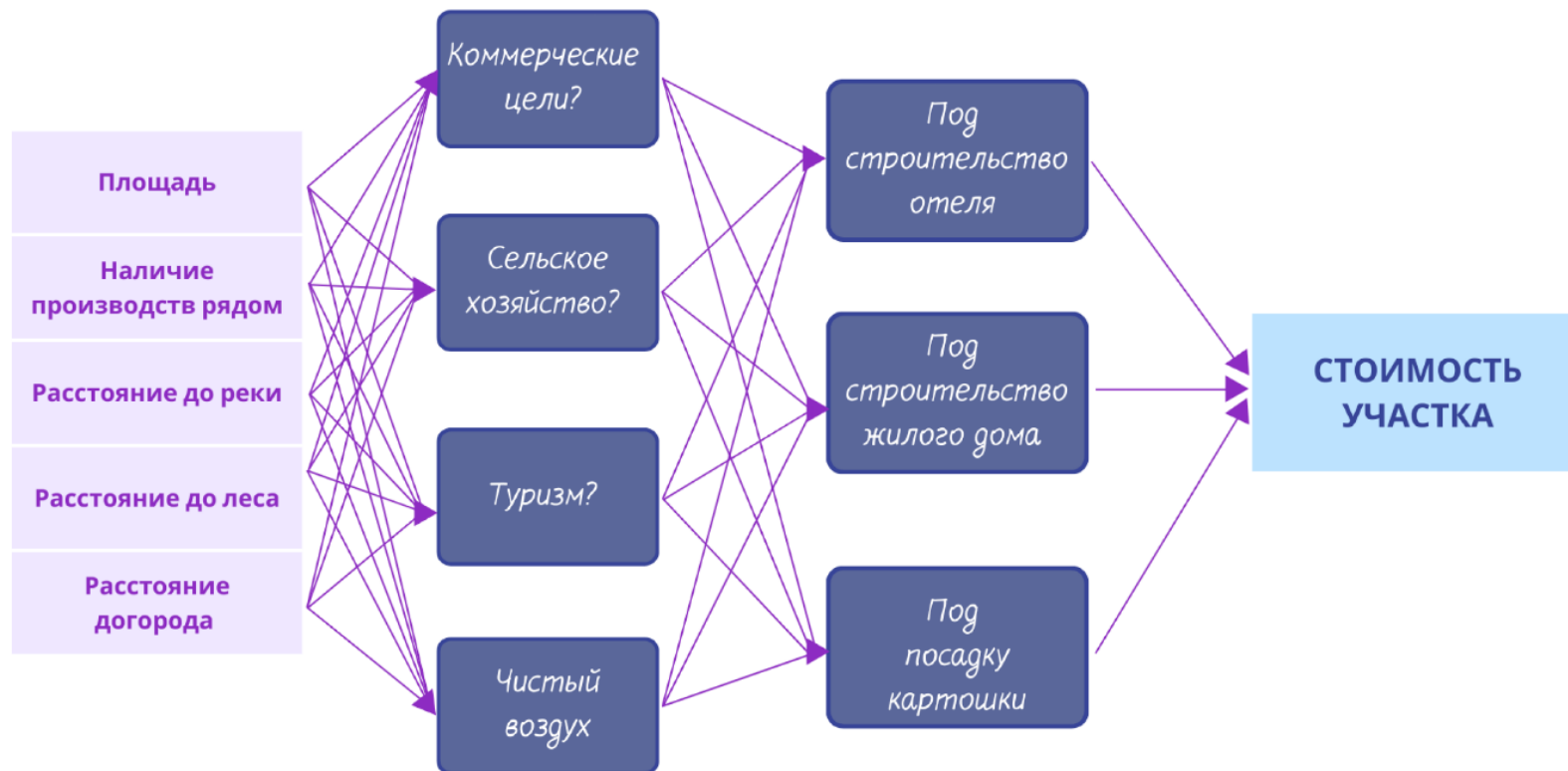


# Виды нейросетей и какие задачи они решают

Тип нейросети	Что делает / Где применяется	Примеры задач
<b>MLP (полносвязная сеть)</b>	Классическая нейросеть для простых табличных данных	Классификация, регрессия, предсказание спроса
<b>CNN (сверточная сеть)</b>	Обрабатывает изображения и видео	Распознавание лиц, объектов, диагностика по снимкам
<b>RNN (рекуррентная)</b>	Обрабатывает <b>последовательности</b>	Текст, временные ряды, прогноз погоды
<b>LSTM / GRU</b>	Улучшенные RNN, умеют "помнить" долгосрочные зависимости	Перевод текста, чат-боты, генерация текста
<b>Transformer</b>	Параллельно обрабатывает последовательности	ChatGPT, BERT, перевод, поиск, ответы на вопросы
<b>GAN (генеративные сети)</b>	Создают новые данные, похожие на обучающие	Генерация лиц, DeepFake, создание картин
<b>AutoEncoder</b>	Сжимают данные и восстанавливают обратно	Сжатие, поиск аномалий, удаление шума
<b>VAE (вариационные автоэнкодеры)</b>	Генеративная версия автоэнкодеров	Генерация изображений, данных
<b>Siamese Network</b>	Сравнивает объекты и учит "похожесть"	Сопоставление лиц, рекомендация похожих товаров
<b>Capsule Network</b>	Учитывает иерархию и положение объектов	Распознавание сложных образов на изображениях

# Нейронные сети, архитектура

Конкретная комбинация используемых слоев называется **архитектурой нейронной сети**. Количество слоев, поставленных друг за другом, обычно называют **глубиной нейронной сети**.

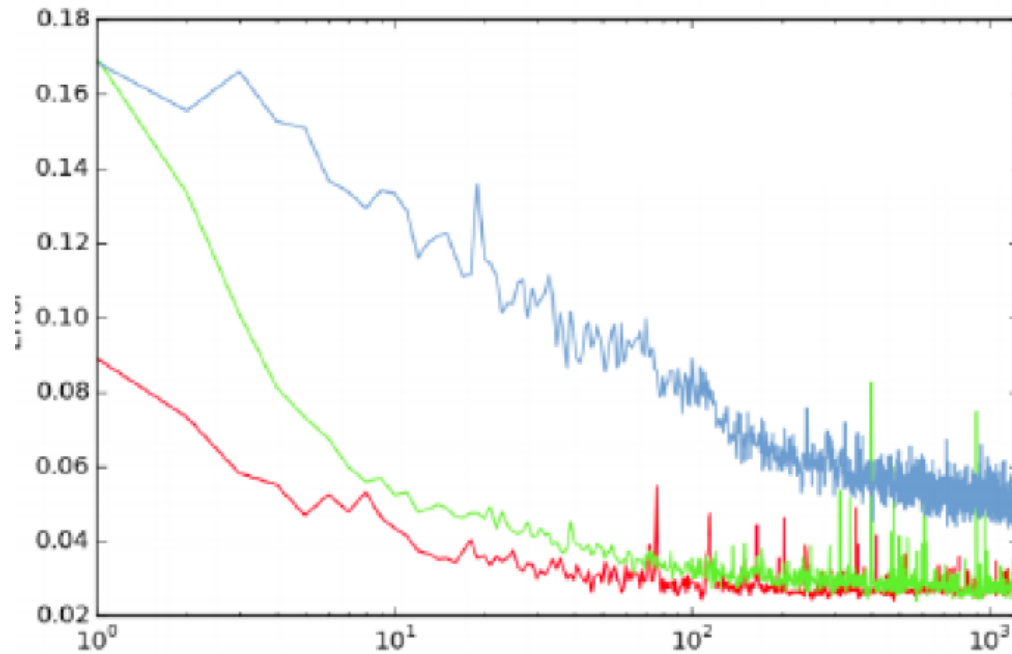


Трехслойная нейронная сеть

# Обучение нейронных сетей. Градиентный спуск

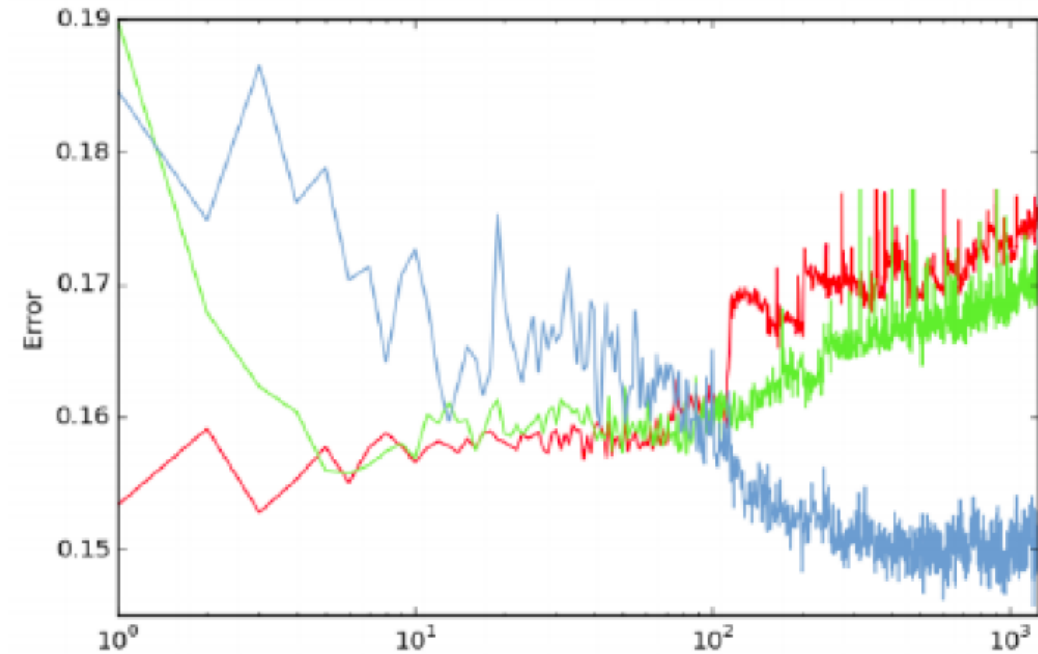
Градиентный спуск работает так: в начале обучения все веса устанавливаются в произвольные значения: генерируется случайный набор чисел. А затем много раз повторяют шаг обновления весов: каждое число (каждый вес) немного меняют, чтобы ошибка предсказания для обучающих данных немного уменьшилась.

### Ошибка на обучающей выборке



Шаги обучения

### Ошибка на тестовой выборке



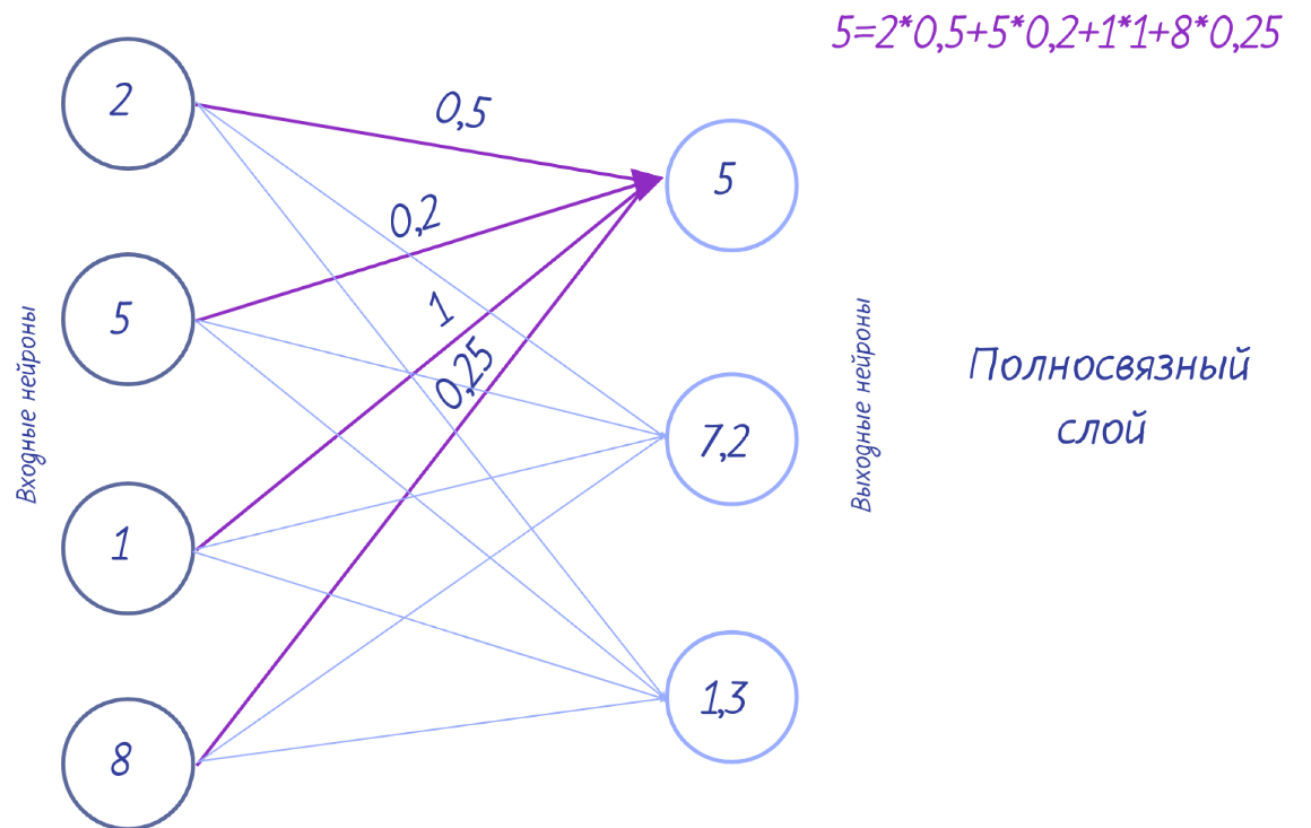
Шаги обучения

# Переобучение нейросетей

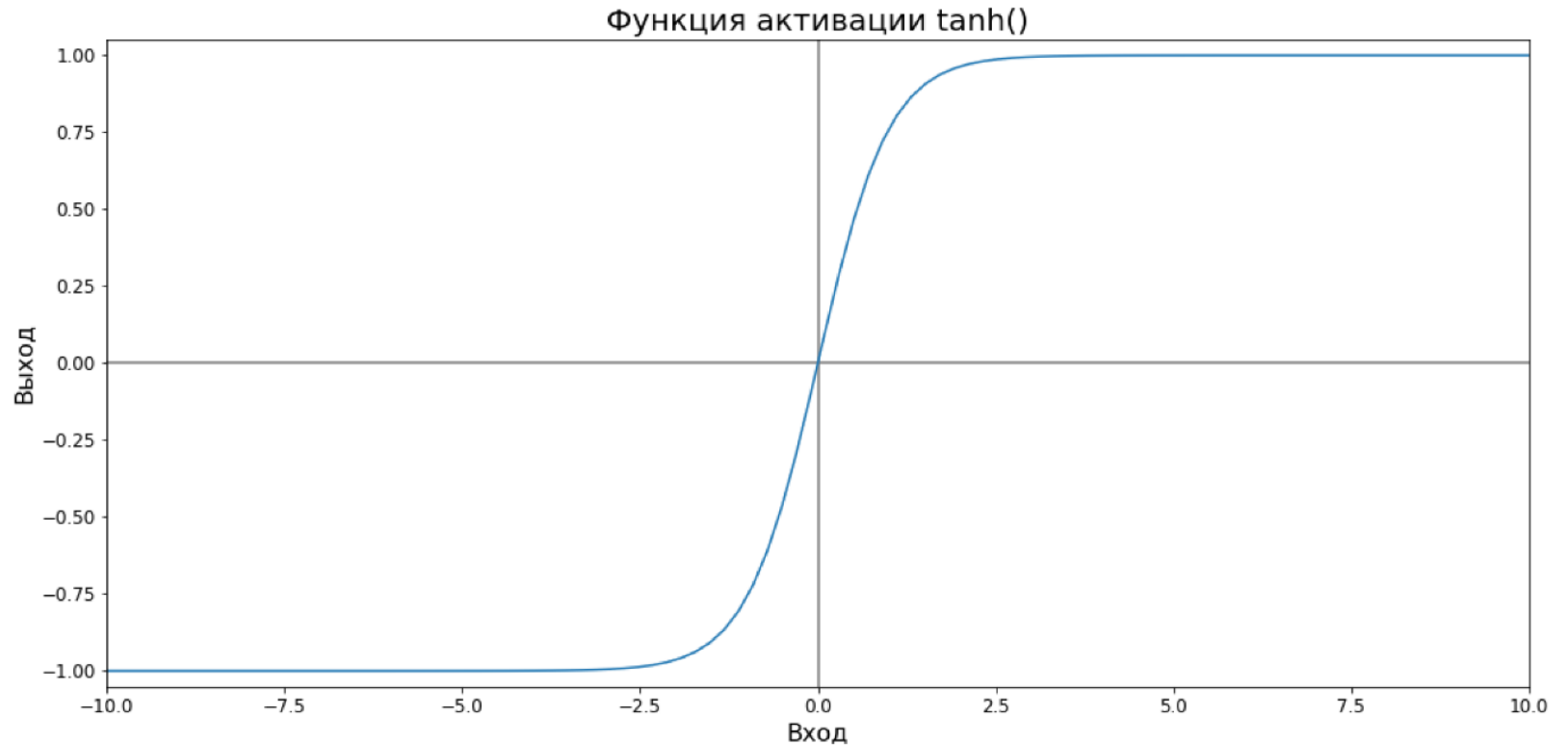
Благодаря тому, что нейронные сети могут выучивать очень сложные зависимости в данных, они очень легко переобучаются, то есть могут запоминать ответы на обучающих данных и при этом плохо работать на новых данных. Мы явно видим переобучение для красной и зеленой линий: ошибка на тестовых данных растет, 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 — **подбор гиперпараметров**. «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 (fully-  
connected layer,  
FC)



Полносвязный  
слой (fully-  
connected layer,  
FC)

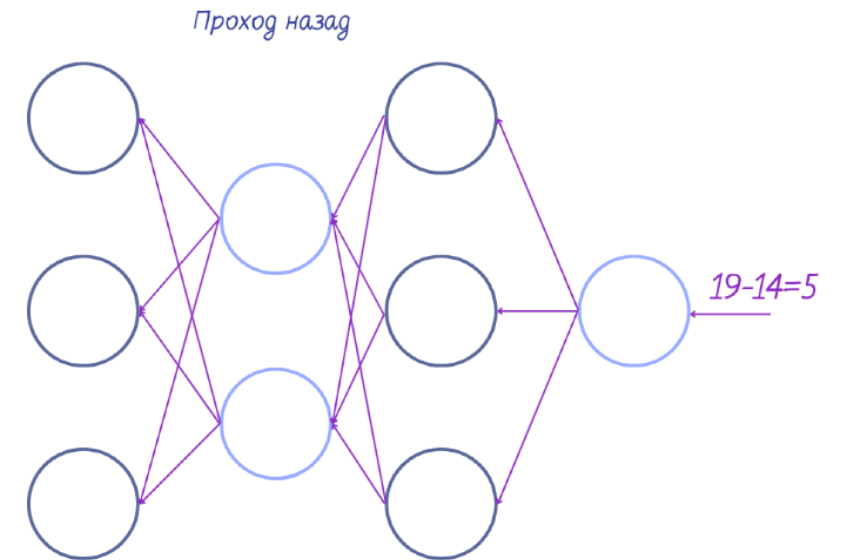
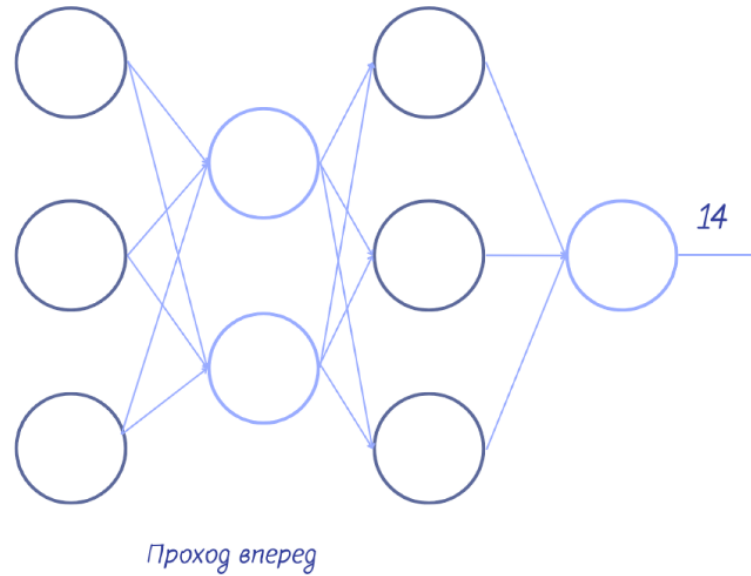


Если чередовать полносвязные слои и слои нелинейности, получится полносвязная нейронная сеть (Multi-Layer Perceptron, MLP) — простейшая нейросетевая архитектура.

Нейрон, активация,  
персептрон.  
Алгоритм обратного  
распространения  
ошибки

---

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.



# 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.

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

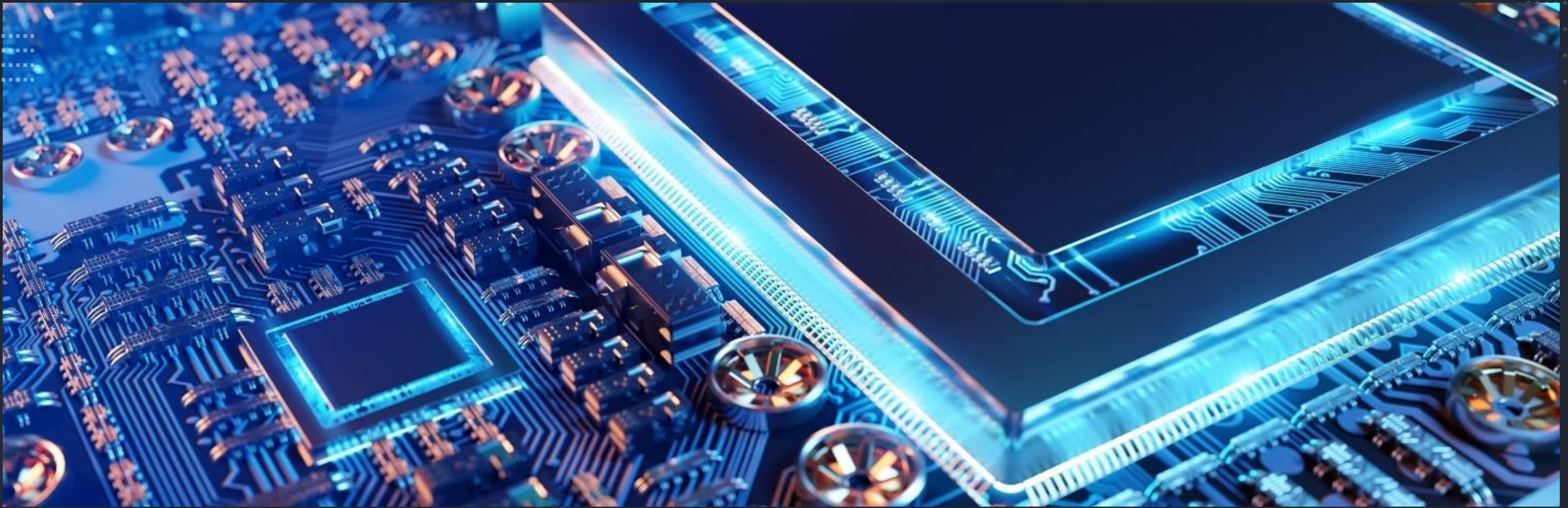


# Нейронные сети в задачах машинного обучения



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

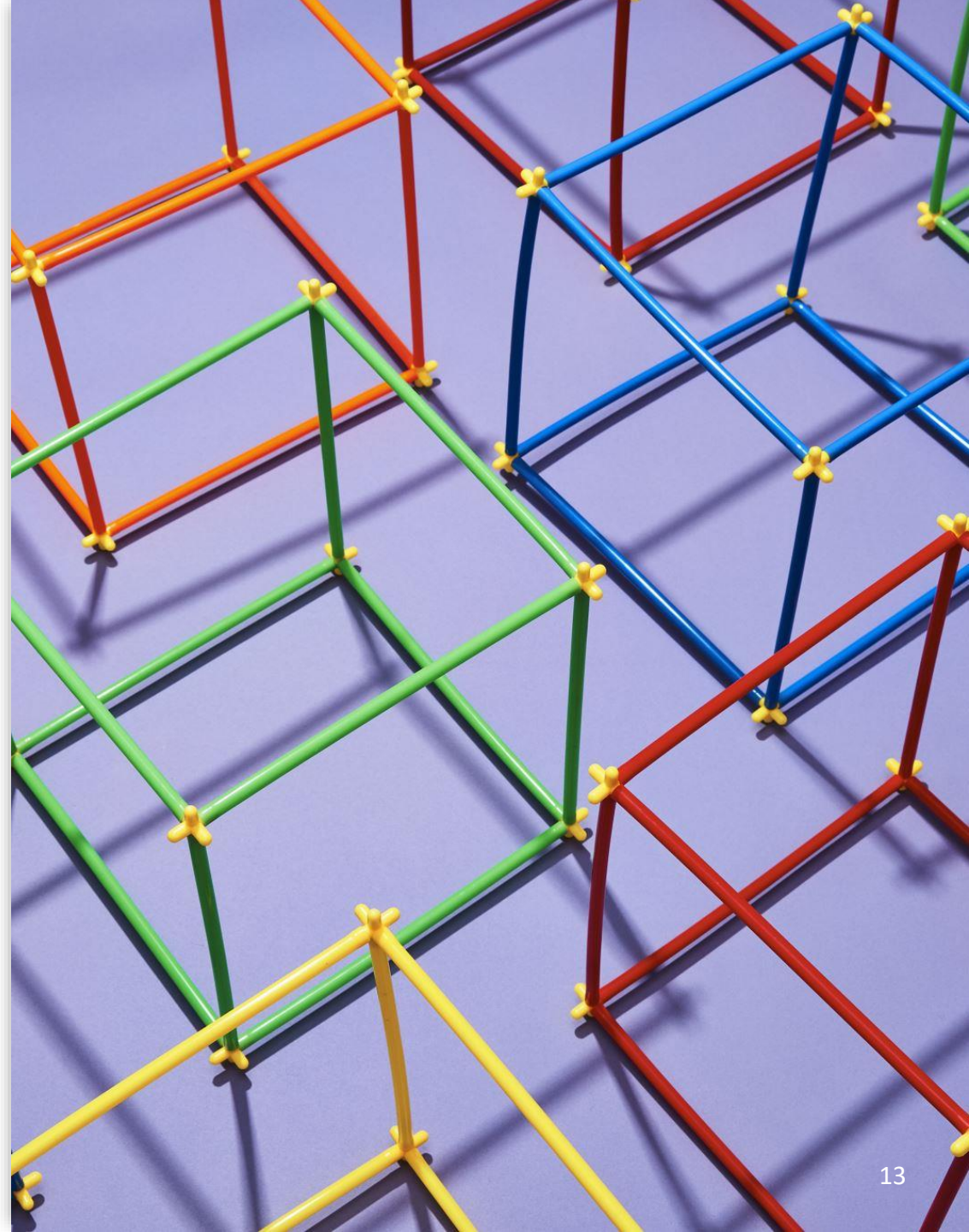


Обучение нейронных сетей вычислительно затратно: для этого используются специальные графические ускорители (Graphics Processing Unit, GPU). Кроме этого, для успешного обучения нейросети необходимо собрать много данных — без этого не получится обучить миллионы параметров нейросети. Появление мощной вычислительной аппаратуры, сбор большого количества данных и разработка особых приемов обучения нейросетей — эти три фактора обусловили прорыв глубинного обучения, произошедший в начале 2010-х годов.

# Популярные архитектуры нейронных сетей

## Сверточные и рекуррентные нейросети

Основная особенность такой нейросети в том, что она не просто преобразует информацию от слоя к слою, а имеет “память”. Каждый нейрон “помнит”, какая информация проходила через него ранее. В итоге эти сети лучше прослеживают связи между блоками информации, даже при небольшом размере самой сети. Поэтому рекуррентные сети лучше других анализируют последовательную информацию – например, человеческую речь или музыку.



.....  
.....  
.....  
.....

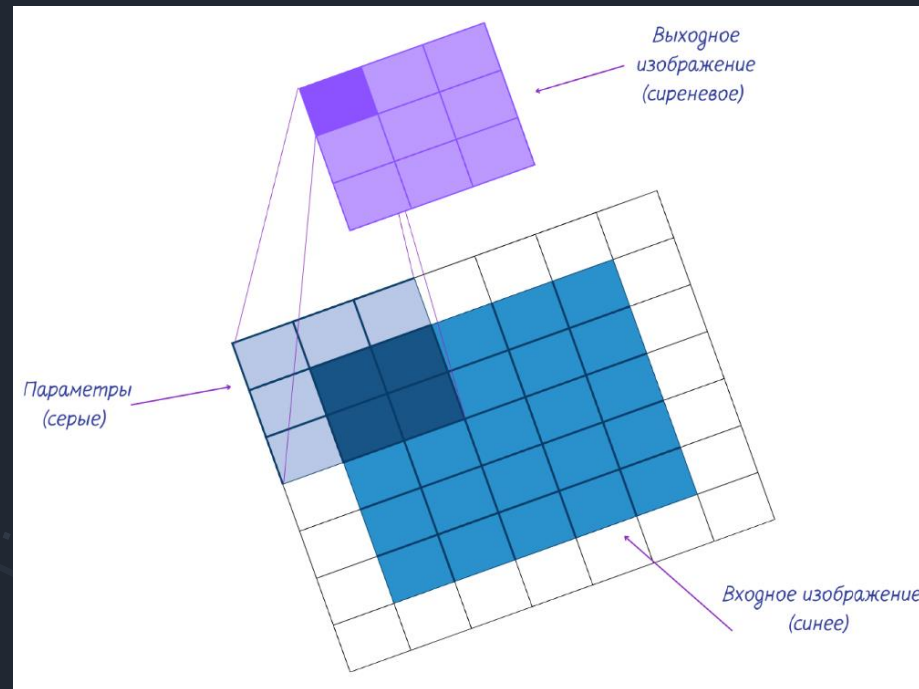


62 62 63 64 65 66 67 67 69 70 71 72 72 73 73 73 72 72 71 70 69 67 66 66 66 65 63 62 61 60 60  
61 62 63 64 66 66 67 68 68 69 70 71 71 72 72 73 72 72 71 71 70 69 68 66 66 65 65 63 62 61 60 60  
61 62 63 64 66 66 68 68 69 70 71 72 73 73 73 72 72 71 71 69 68 67 66 66 65 65 64 63 62 61 61  
61 63 64 64 66 67 68 68 68 69 70 71 71 73 73 74 73 73 71 70 69 68 66 66 65 64 63 62 61 61 60  
61 63 64 65 67 68 69 69 70 70 71 71 72 55 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 45 4 5 11 48 72 71 71 69 69 68 67 66 65 64 62 62 60 59 59  
63 65 66 66 68 68 69 70 71 71 72 18 4 4 7 8 66 71 70 69 68 68 67 66 65 64 63 61 59 59 58  
63 65 67 67 68 69 69 70 71 71 72 64 4 27 24 54 33 29 52 64 68 68 67 66 65 64 63 62 61 59 58 58  
64 65 66 66 68 69 70 71 41 24 24 12 17 24 45 00 37 43 36 52 66 68 67 66 65 64 63 61 60 59 58 57  
65 66 67 67 68 69 71 30 6 6 6 5 34 36 12 47 34 17 29 54 43 63 67 66 65 64 63 62 60 59 58 57  
64 65 66 66 68 69 38 6 6 5 5 7 16 19 4 47 44 27 24 40 67 66 66 65 65 64 63 61 60 59 58 57  
63 64 65 65 67 30 6 6 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 34 5 5 5 5 5 5 4 19 6 7 54 64 20 59 65 65 64 64 64 63 62 61 60 59 57 56  
63 64 64 65 14 5 6 5 5 4 5 4 18 7 5 4 19 10 11 65 64 64 64 63 61 66 62 61 60 59 58 56  
63 64 64 65 53 7 4 5 6 6 7 10 6 5 5 4 21 24 18 64 64 64 63 62 64 65 62 62 60 59 58 57  
64 64 64 64 65 50 4 4 4 5 11 16 6 6 4 6 35 16 26 66 64 64 63 61 72 67 63 62 61 59 58 57  
64 64 64 64 65 46 4 4 4 5 6 9 8 5 29 10 43 50 29 57 64 64 63 61 70 67 62 64 65 59 59 57  
64 64 64 65 66 27 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 58  
49 50 62 65 57 5 5 6 5 6 6 6 4 59 28 60 58 44 22 63 71 72 60 69 68 61 60 58 59 59 58  
42 52 57 55 26 5 5 5 5 5 5 5 70 40 43 61 62 64 39 42 64 60 62 56 63 65 65 67 61 53 53  
32 32 32 33 6 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 37 5 5 5 5 6 5 6 6 42 60 58 34 42 39 43 37 26 29 40 26 29 26 35 42 35 33 18 19  
52 53 51 22 5 5 5 5 6 5 6 5 44 50 17 51 54 53 54 56 51 22 54 54 55 55 54 53 53 52 52  
54 54 53 8 5 5 5 5 6 5 6 13 52 42 21 51 54 51 49 49 50 22 41 45 42 42 41 40 41 44 43 42  
52 52 54 37 8 5 5 6 6 5 6 26 55 32 32 54 53 51 51 51 51 44 25 51 51 49 49 50 49 48 46 46  
54 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  
51 52 51 53 27 14 5 4 5 4 7 47 51 21 39 49 47 49 52 52 52 49 35 31 48 46 47 47 47 46 46 43  
48 50 51 53 25 14 17 8 4 4 17 46 40 18 43 47 46 49 52 54 53 53 54 18 50 49 46 47 47 47 47 45  
49 49 49 49 22 12 20 24 6 14 35 51 39 48 48 50 51 51 49 51 51 52 50 41 58 48 47 47 47 45 45 46  
51 49 50 50 22 13 19 36 13 12 42 50 40 73 50 50 50 49 48 49 49 48 49 45 51 46 44 44 44 42 45 47  
47 49 49 47 20 16 26 39 21 15 36 48 42 61 47 48 51 47 50 51 51 51 49 47 47 52 47 47 44 43 45 46  
48 50 48 52 19 13 33 36 18 18 30 49 51 54 47 47 49 46 46 49 49 49 47 44 53 44 48 44 46 46 45

# Нейронные сети для анализа изображений

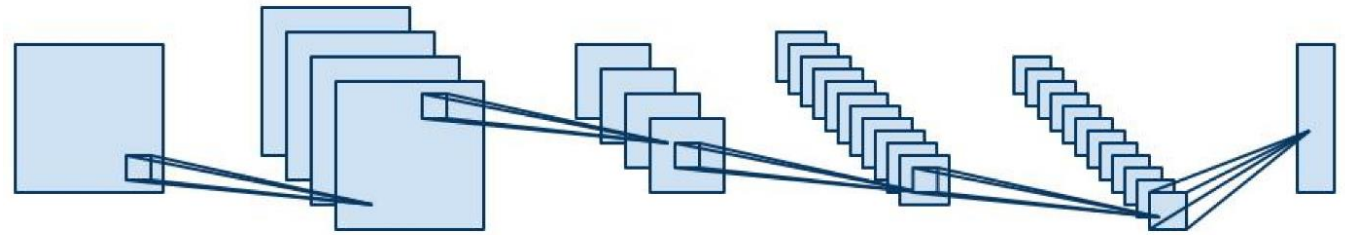
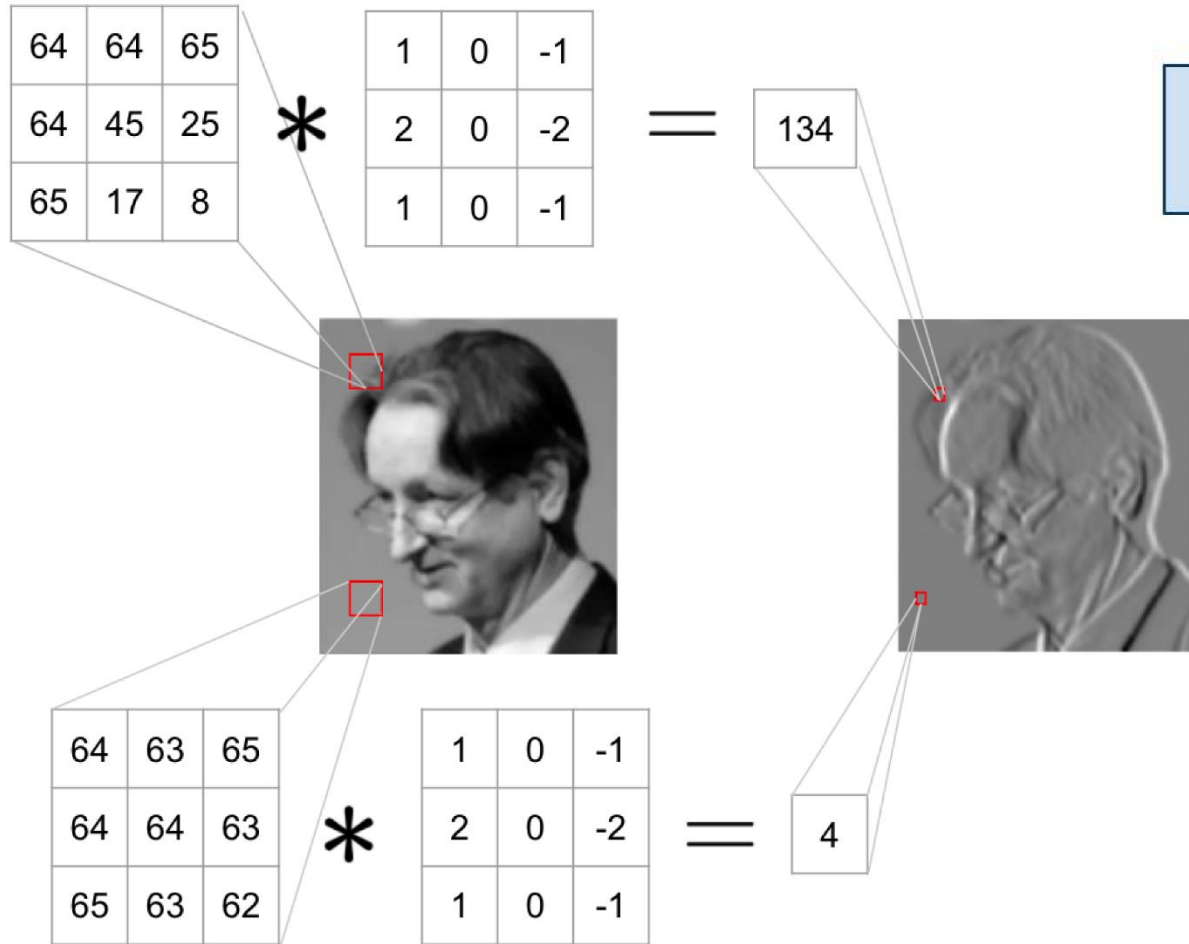
Сверточные нейронные сети

# Нейронные сети для анализа изображений



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

# Пример работы фильтра Собеля



MIT: Alexander Amini, 2018 [introtodeeplearning.com](http://introtodeeplearning.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).

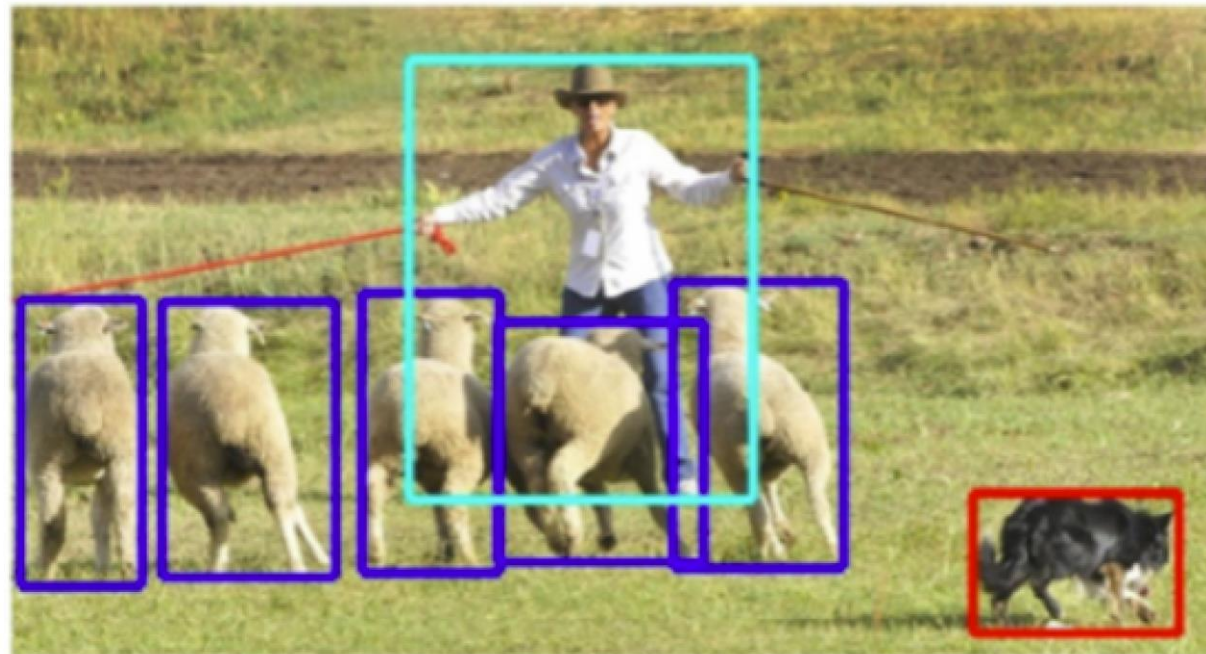


Классификация изображения

# Задачи анализа изображений

## 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.

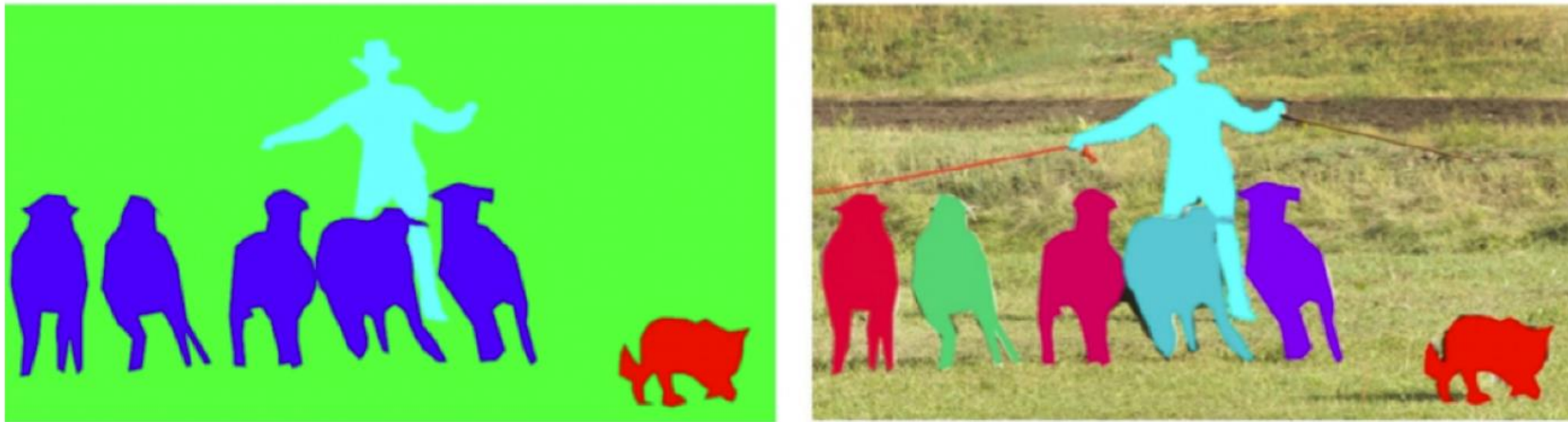


Детекция

# Задачи анализа изображений

## 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