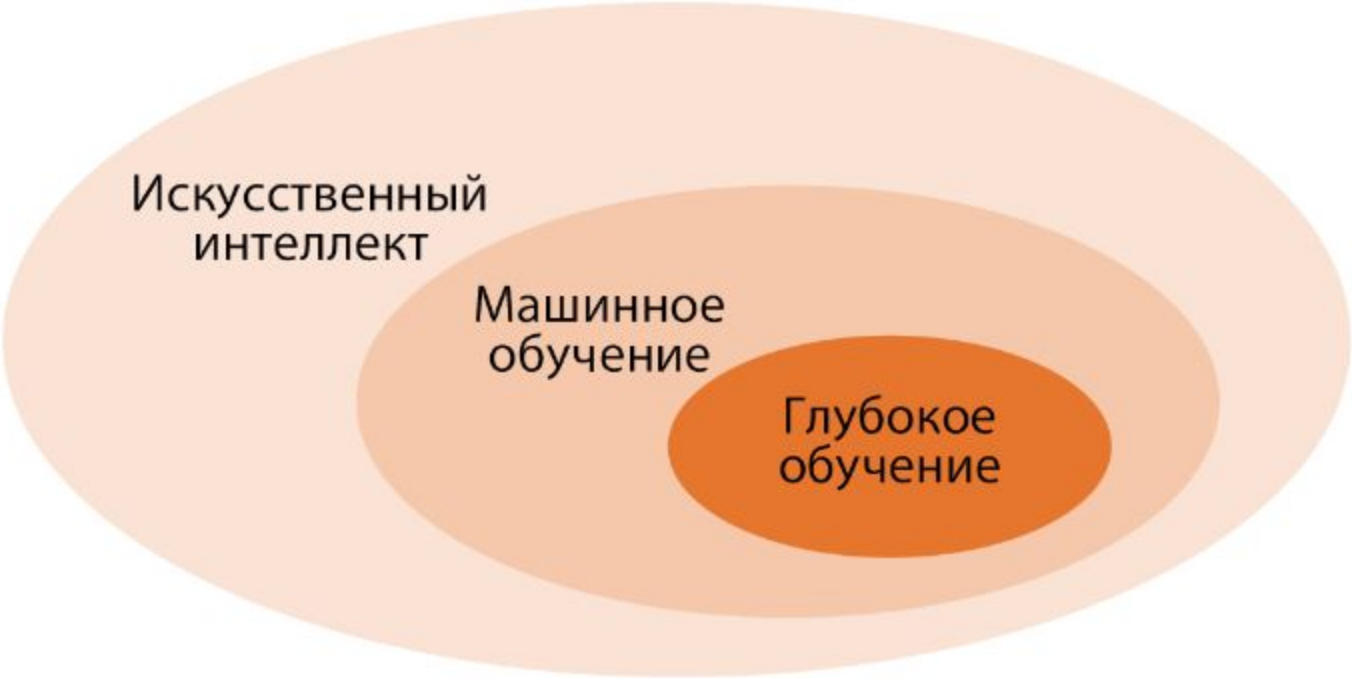
Нейронные сети

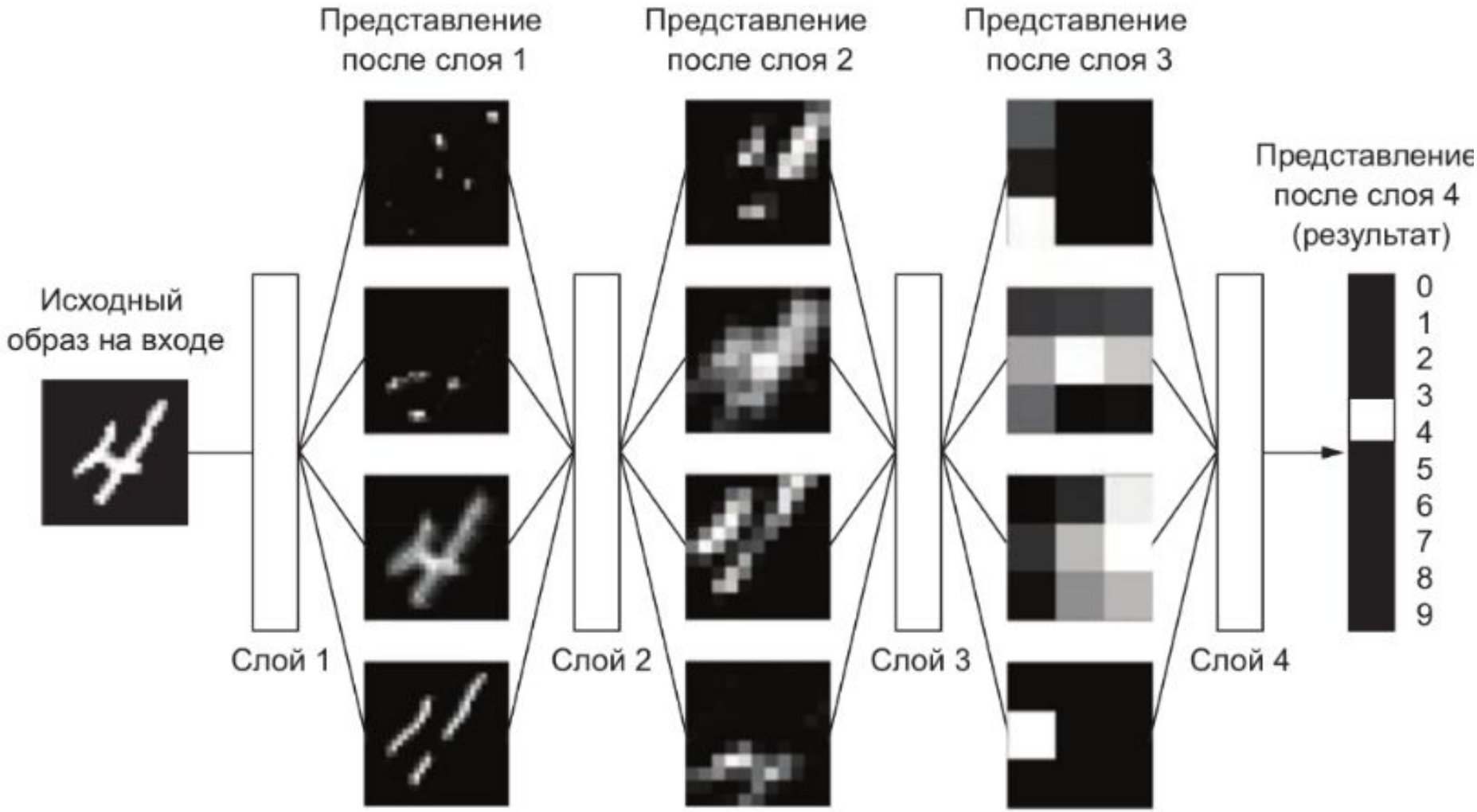

Искусственный интеллект
Машинное обучение
Глубокое обучение


Правила
Данные
Классические алгоритмы
Ответы
Данные
Ответы
Машинное обучение
Правила

Глубокое обучение

- особый раздел машинного обучения: делает упор на анализ последовательных слоев (или уровней) все более значимых представлений.
- Под глубиной в глубоком обучении подразумевается не более глубокое понимание, которое достигается этим подходом, а многослойность в представлении модели.
- Количество слоев, из которых состоит модель данных, называют глубиной модели.
- «многослойное обучение» и «иерархическое обучение».
- В глубоком обучении многослойные представления изучаются (чаще всего) с применением моделей, называемых нейронными сетями. Их структура представлена в виде слоев, наложенных друг на друга.
- Понятие нейронной сети заимствовано из нейробиологии. Хотя источником некоторых основополагающих идей глубокого обучения частично являются науки о мозге, модель глубокого обучения не является моделью мозга человека.
- Нет фактов, доказывающих, что мозг работает по принципам, подобным механизмам, которые используются в современных моделях глубокого обучения.

Представления, полученные алгоритмом глубокого обучения



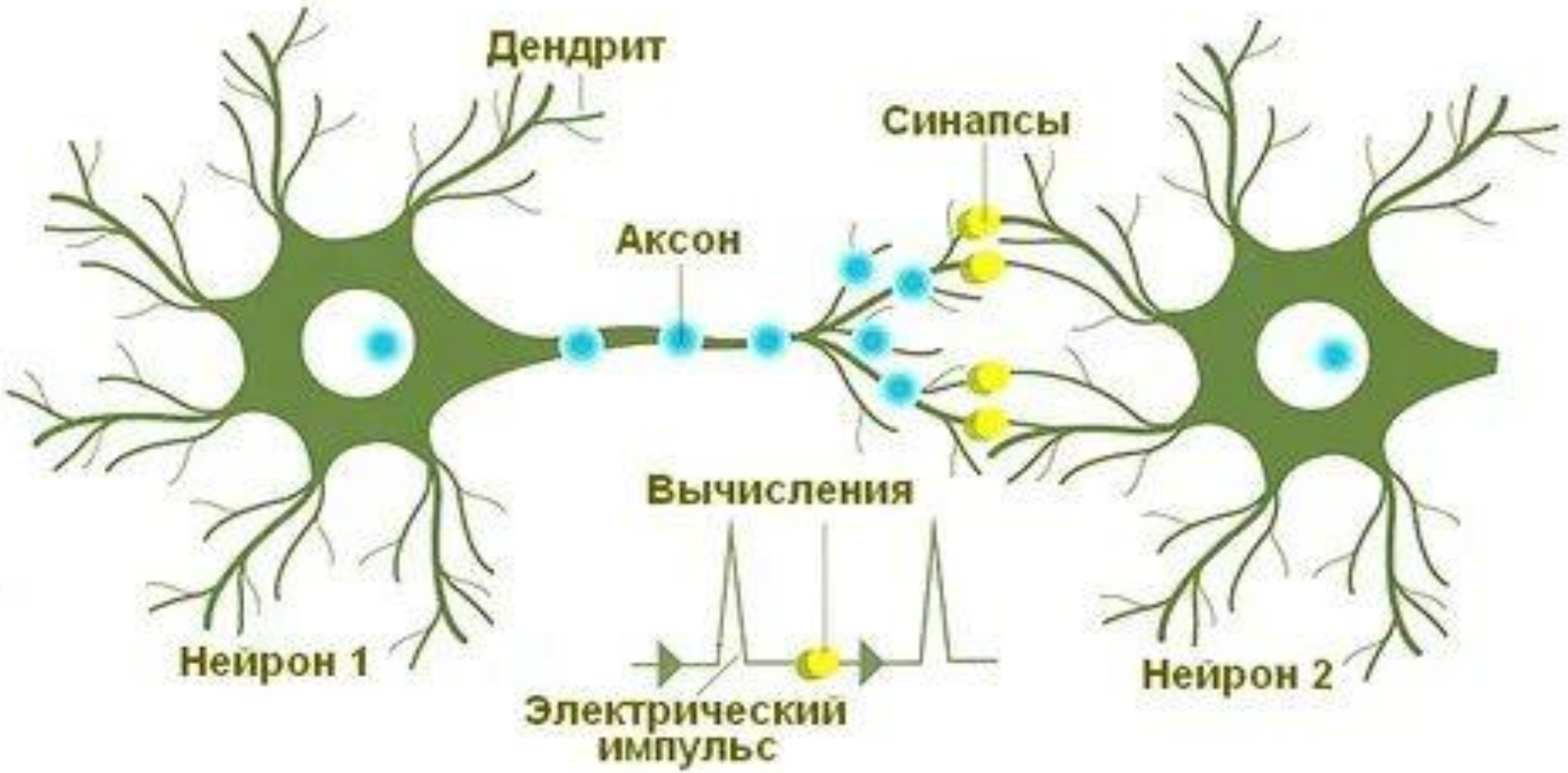
Задачи, решаемые глубокими нейронными сетями

- классификация изображений на уровне человека;
- распознавание речи на уровне человека;
- распознавание рукописного текста на уровне человека;
- повышение качества машинного перевода с одного языка на другой;
- улучшение качества чтения текста вслух машиной;
- создание цифровых помощников: Yandex – Алиса, Google Now и Amazon – Alexa;
- управление автомобилем на уровне, сравнимом с человеком;
- повышение точности целевой рекламы Google, Baidu и Bing;
- повышение релевантности поиска в Интернете;
- машины могут отвечать на вопросы, заданные вслух;
- алгоритм обыграл человека в «Го».

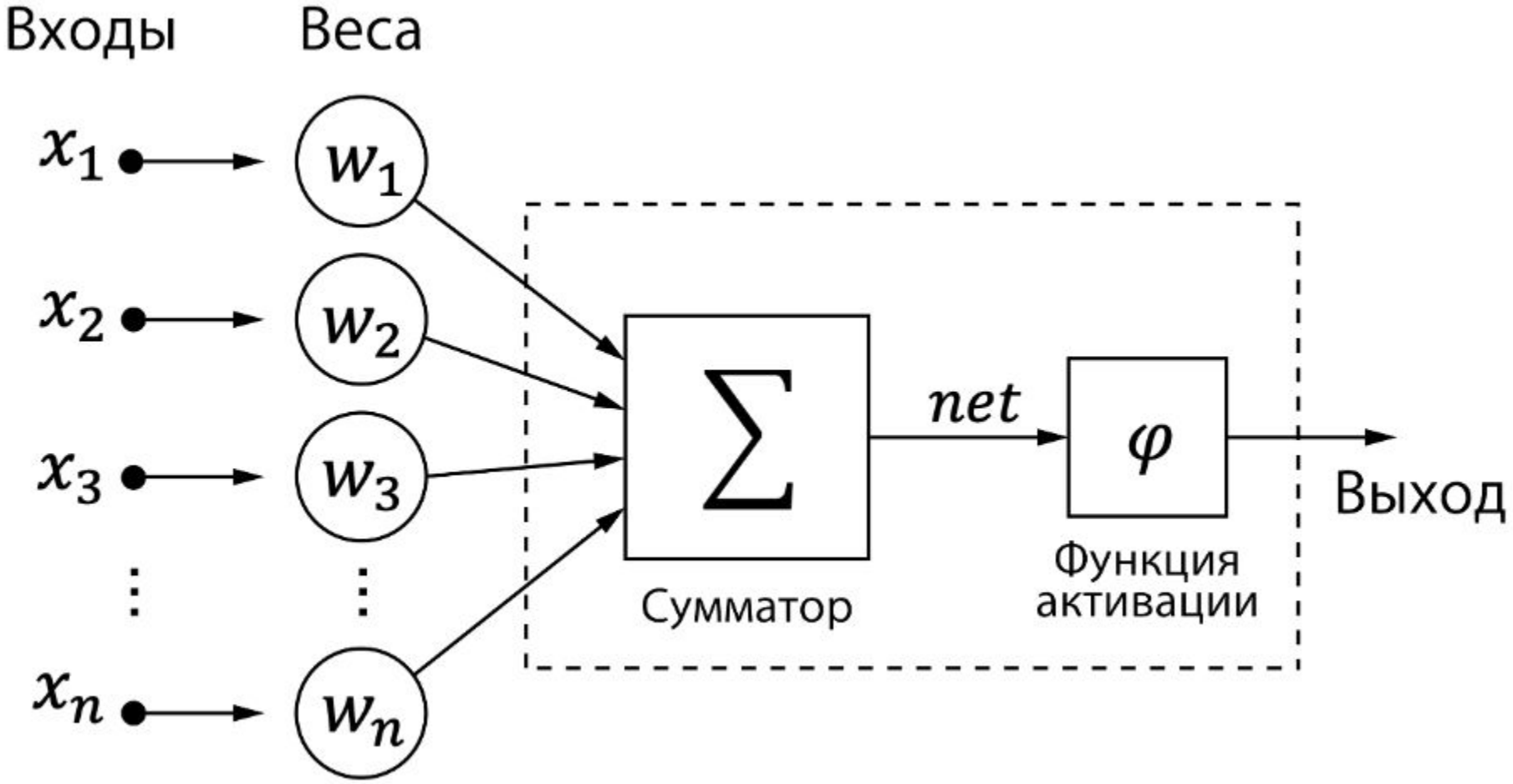
Методика глубокого обучения

- имеет две важные характеристики:
  - Поэтапно, послойно конструирует все более сложные представления.
  - Исследует промежуточные представления совместно, за счет чего каждый слой обновляется в соответствии с информацией, полученной от представлений других слоев.

Нейрон человека



Модель искусственного нейрона



- Входы – это некий вектор $X = [x_1, x_2, \ldots, x_n]$ входных значений. Являются аналогом дендритов в реальном нейроне.
- Веса – это вектор обучаемых параметров нейронной сети $w = [w_1, w_2, \ldots, w_n]$, которые обозначают значимость каждого входного признака для итогового результата (по сути, аналог синапсов реального нейрона). В конечном итоге, если какой-то вес $w_i = 0$, то признак x_i является абсолютно незначимым для нейронной сети. И наоборот, чем больше значение w_i, тем сильнее влияет признак x_i на результат, выданный нейронной сетью. Длина вектора весов совпадает с длиной вектора входных параметров.
- Сумматор – функция, которая просто суммирует произведения признаков на их веса. Можно сказать, что это ядро нейрона, которое аккумулирует заряд

$$x_1 w_1 + x_2 w_2 + \cdots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

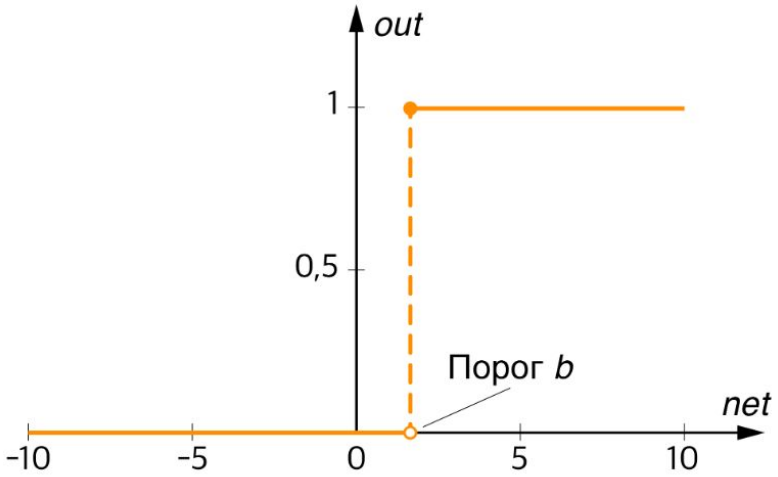
- Функция активации – функция, которая принимает на вход результат сумматора и выполняет некое преобразование, чтобы превратить сумму взвешенных входов в адекватный выход, который можно интерпретировать с точки зрения решения поставленной задачи. Так как в случае малой величины сумматорной функции функция активации может вернуть 0, то есть «ничего», то она является неким аналогом того механизма в реальном нейроне, которые отвечает за возбуждения

Функции активации

- *Функция единичного скачка* (*она же функция Хэвисайда*)

$$\theta(x) = \begin{cases} 0, x < 0; \\ 1, x \geqslant 0. \end{cases}$$

- где x – взвешенная сумма входных параметров.
- Данная функция принимает на вход взвешенную сумму входных параметров и, если они меньше 0, то возвращает 0, иначе – 1.



Функции активации

- где θ – заданный порог, S – взвешенная сумма входных параметров.

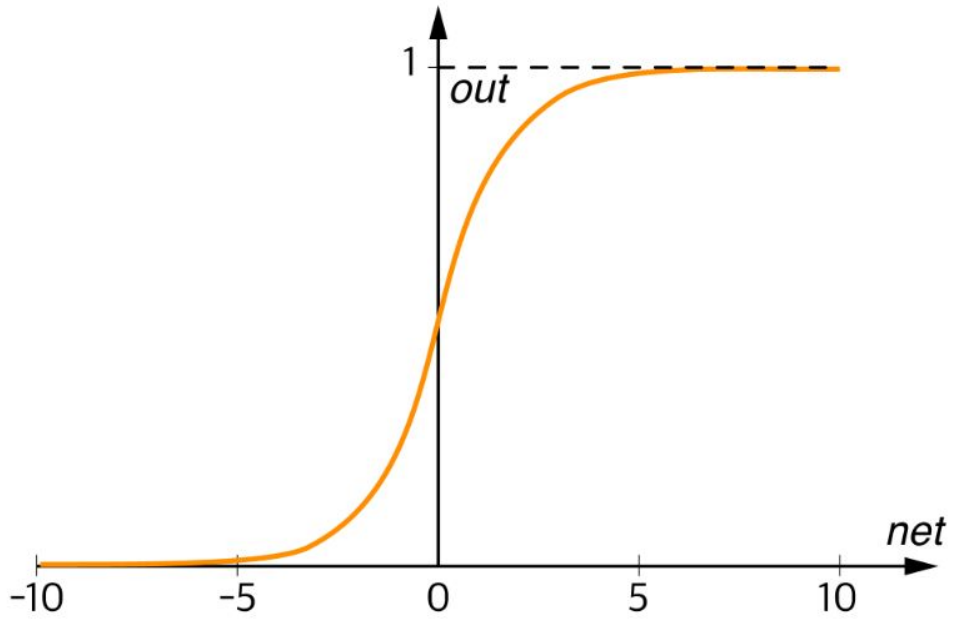
$$f(S) = \begin{cases} 0, S < \theta \\ 1, S \geq \theta \end{cases}$$

Функции активации

- *Сигмоидальная функция (она же логистическая функция)*

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

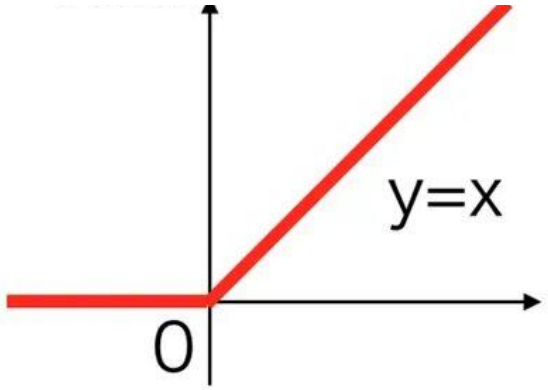
- где x – взвешенная сумма входных параметров.



Функции активации

- *ReLU (rectified linear units)*

$$f(x) = max(0, x)$$



Преимущества

- является хорошим аппроксиматором – подходит для любой функции.
- создает разреженность при активации нейронов
- значимо повышает скорость сходимости градиентного спуска по сравнению с сигмоидой и гиперболическим тангенсом.

Недостатки

- проблема умирающего *ReLU*

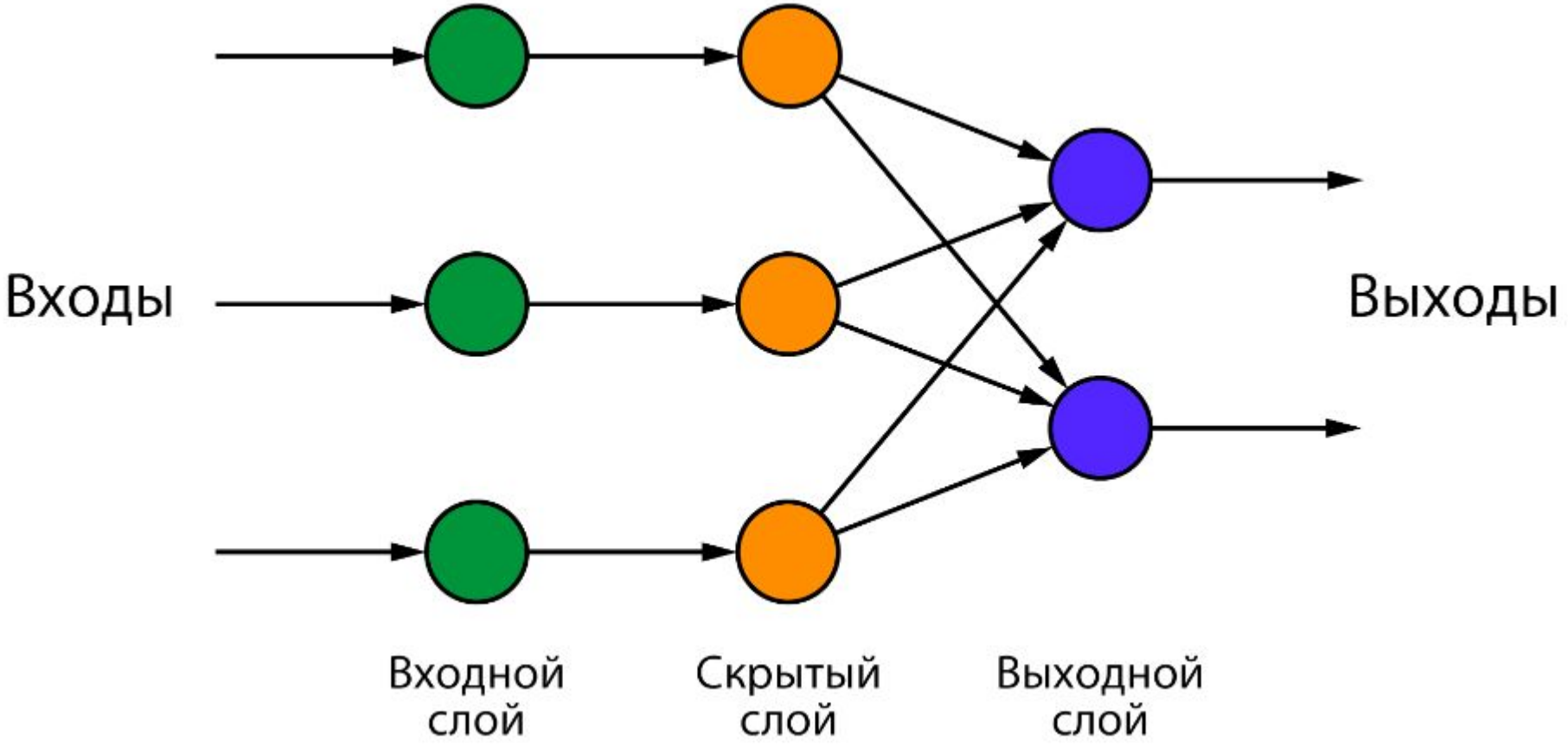
Существуют модификации *ReLU*, которые частично решают эту проблему: *Leaky ReLU*, *Parametric ReLU*, *Randomized ReLU*.

Классификация нейронных сетей

По типу распространения сигнала нейронные сети можно разделить на:

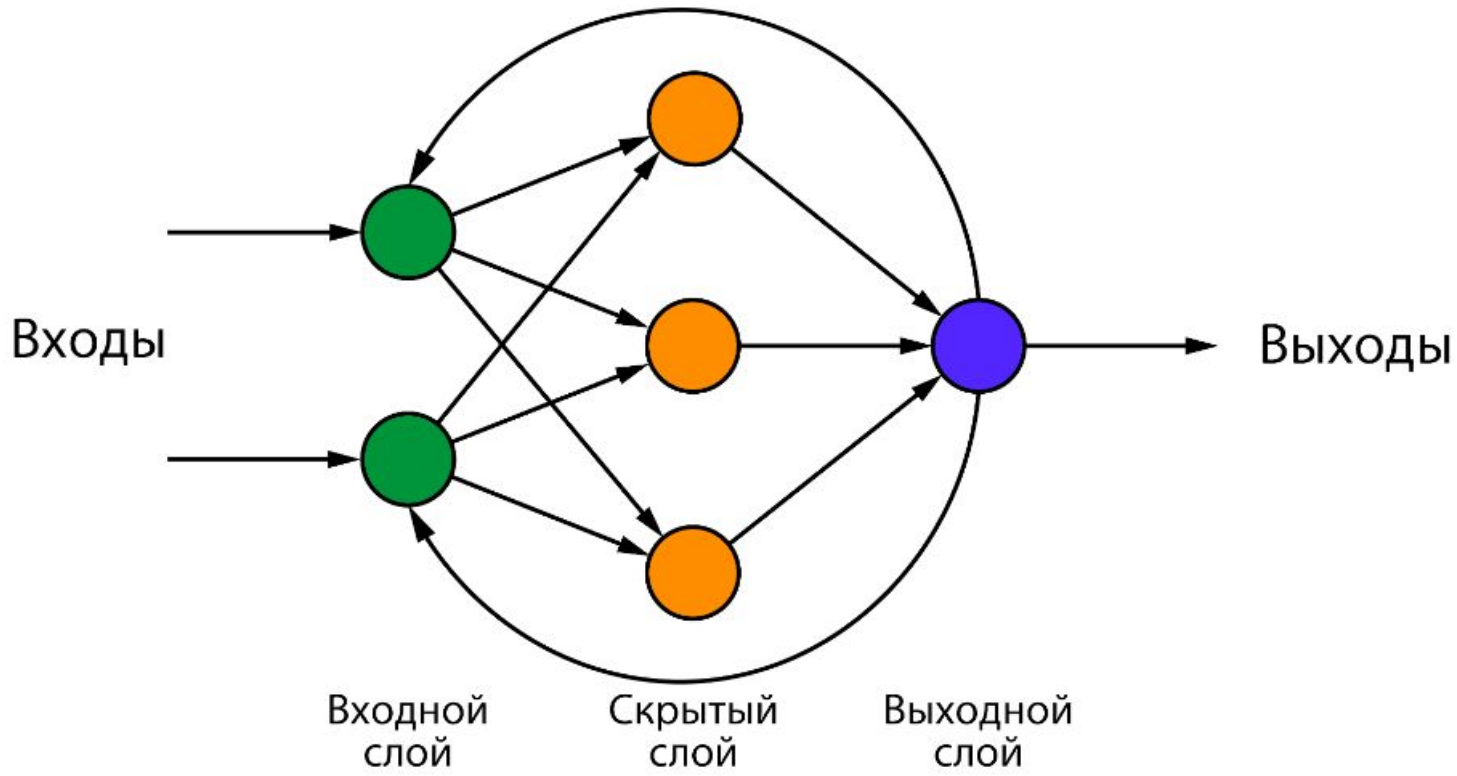
- нейронные сети с прямым распространением сигнала;
- нейронные сети с наличием по крайней мере одной обратной связи (рекуррентные нейронные сети).

Классификация нейронных сетей



Сети с прямым распространением сигнала решают множество задач классификации и регрессии на табличных данных, а также практически все задачи на изображениях и часть задач обработки естественного языка

Классификация нейронных сетей



Необходимость в рекуррентных нейронных сетях возникает при решении таких задач, когда требуется владеть информацией о предыдущих элементах обучающей последовательности: временные ряды, обработка текста с учетом последовательности слов.

Классификация нейронных сетей

- Однослойной считается нейронная сеть, имеющая лишь входной и выходной слой, без наличия скрытых слоев.
  - В такой сети сигналы попадают сразу с входного слоя на выходной. Сеть подобного рода способна настроиться не под все сложные зависимости, очень чувствительна к предварительной обработке признаков и требует их тщательного отбора. Преимуществом является достаточно быстрая скорость обучения.
- Многослойной (глубокой) нейронной сетью считается сеть с минимум двумя скрытыми слоями (входной и выходной, естественно, также остаются).
  - Способна самостоятельно от слоя к слою преобразовывать входные признаки и проводить их отбор за счет уменьшения/увеличения весов. Но наличие большого количества слоев, а значит и обучаемых параметров, также ведет к увеличению времени обучения (некоторые нейронные сети обучались по несколько месяцев). А еще из-за очень сложной структуры данный тип сетей подвержен

Обучение нейронной сети

- Сила нейронных сетей в обучении весов связей между нейронами.
- В ходе множества итераций нейронная сеть сравнивает свой выход с настоящим ответом и, в случае ошибки, корректирует все свои веса таким образом, чтобы на следующей итерации добиться верного ответа.
- Алгоритм такого обучения называется «алгоритм обратного распространения ошибки». Это связано с тем, что нейронная сеть понимает, ошиблась она или нет, только на выходе, и затем информацию об ошибке она распространяет в обратном порядке от выхода ко входу.
- Чем больше в нейронной сети слоев и нейронов, тем больше связей (а значит, и весов, эти связи характеризующих) и тем более тонкой может быть настройка нейронной сети при

Обучение с учителем

Прикладная сфера	Входные данные	Верный ответ
Классификация изображений	Матрицы изображений рукописных цифр (одинаковой размерности)	Число от 0 до 9 – цифра, которая реально написана на изображении
Классификация текстов	Вектор, полученный из текста новостной ленты	Тема новостей: спорт, политика, финансы и т. д.
Регрессия	Вектор с параметрами дома: площадь, количество комнат, удаленность от центра и т. д.	Число, обозначающее реальную стоимость дома
Временной ряд	Вектор с последовательным почасовым трафиком посетителей на сайте	Число (или вектор чисел), обозначающее число посетителей в следующий момент времени (ряд последовательных моментов времени)

Обучение без учителя

- **кластеризация** – задача разбиения выборки на *N* (число кластеров может задаваться заранее, а может вычисляться алгоритмом в процессе обучения) кластеров по заранее неведомым признакам. Отличие от классификации в том, что метки классов заранее не заданы и полученные кластеры необходимо затем интерпретировать в язык человеческих терминов;
- **сокращения размерности** – задача представления многомерного исходного вектора в вектор меньшей размерности с минимальной потерей информации. Таким образом может происходить отсев малоинформативных признаков, что уменьшит шум в данных и увеличит скорость обработки данных алгоритмом, а также может повысить качество какой-нибудь последующей задачи (например, классификации). Вырожденные случаем сокращения размерности (до двумерных или трехмерных векторов) можно использовать для визуализации данных.

Обучение с подкреплением

- модель не имеет информации о системе, но при этом может производить некие действия, влияющие на систему.
- при воздействии модели на систему та переходит в новое состояние.
- в зависимости от того, приближает это модель к желаемой цели или отдаляет от нее, система посылает модели положительное или отрицательное вознаграждение.
- Например, есть сеть улиц, по которым едет машина. Модель может управлять скоростью машины и ее поворотами в заданных пределах. Целевую функцию можно сформулировать, как «доехать из пункта A в пункт B за минимальное время», а к тому же можно еще наложить ограничение, запрещающее врезаться в стены. Тогда модель будет управлять машиной, а система будет ее штрафовать за удар о стену или поощрять за оптимально выбранный маршрут.

Другие подходы к обучению

- **полное обучение (пакетный метод)** – подход, при котором все обучающие данные подаются одним единым блоком. Преимуществом является то, что происходит значительная экономия времени на обучение, но есть высокая вероятность, что пострадает точность модели;
- **онлайн-обучение (стохастическое обучение)** – подход, при котором обучающие примеры подаются по одному и после каждого из них происходит процесс обратного распространения ошибки. Метод более затратный по времени и ресурсам. Есть риск попасть в локальный минимум. Также модель может «забыть» информацию о более ранних примерах;
- **обучение на мини-выборках (мини-пакетах)** – попытка найти золотую середину между двумя предыдущими вариантами. За счет того, что мини-выборки формируются случайным образом и усредняют ошибку, по всей подвыборке снижается риск попасть в локальный минимум. Метод быстрее онлайн-обучения. Размер мини-выборки можно регулировать исходя из технических возможностей машины, а также подбирать экспериментально для схождения к лучшему результату.

Библиотеки для обучения нейронной сети

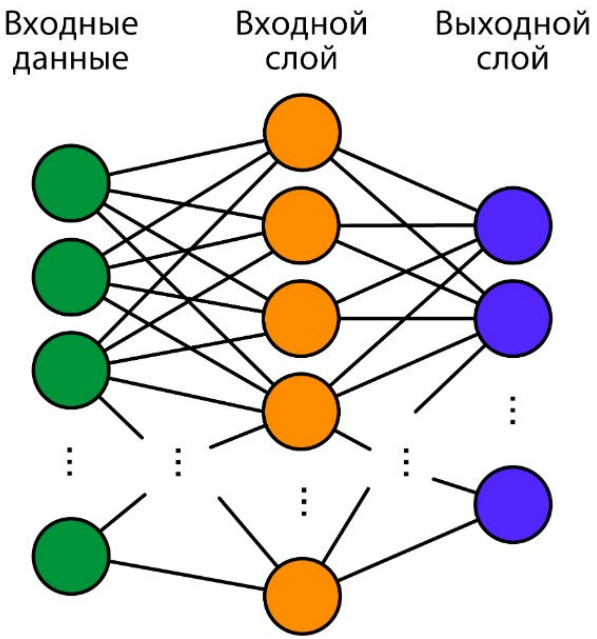
- **TensorFlow**
  - Базовый язык C++, но имеет API для Python. Разработан Google. Вычисления с использованием графов потоков данных. Предлагает мощные средства мониторинга процесса обучения моделей и визуализации. Поддерживает распределенное обучение. Имеет достаточно высокий входной порог
- **Theano**
  - Базовый язык Python. Разработан университетом Монреаля. Вычисления с использованием графов потоков данных. Эффективная обработка тензоров. Вычисления выражаются NumPy – подобным синтаксисом. Имеет высокий входной порог
- **PyTorch**
  - Базовый язык Lua. Поддерживает API для Lua, Python, Java, C++. Разработан Ронаном Коллабертом. Имеет множество модульных элементов, которые легко комбинировать. Легко писать собственные типы слоев и работать на GPU. Имеет API разных уровней абстракции
- **CNTK**
- Базовый язык C++. Поддерживает API для C++, C#, Python

Библиотеки для обучения нейронной сети

- **CNTK**
  - Базовый язык C++. Поддерживает API для C++, C#, Python, Java. Разработана Microsoft. Обеспечивает скорость обучения, сравнимую с TensorFlow, а на рекуррентных сетях превосходит его. Имеет API разных уровней абстракции
- **Caffe**
  - Базовый язык C++. Поддерживает API для C++, Python. Разработан в центре компьютерного зрения и обучения Беркли. Имеет небольшую скорость относительно других библиотек
- **Keras**
  - Библиотека верхнего уровня. Поддерживает Python. В качестве вычислительного back-end использует TensorFlow или Theano. Позволяет создавать и обучать нейронные сети на очень высоком уровне абстракции. Имеет низкий порог вхождения

Распознавание предметов одежды

- **Полносвязная нейронная сеть прямого распространения** – Fully Connected Feed-Forward Neural Networks, FNN
- Как видно из названия, нейроны данной сети полностью связаны между собой. Каждый нейрон связан со всеми нейронами предыдущего слоя, собственно, как и со всеми последующего. Это же касается и входов, и выходов нейронной сети – все входы подаются на каждый нейрон, а выходы всех нейронов последнего слоя подаются на выходы нейронной сети.
- С помощью FNN достаточно
- хорошо решаются многие
- задачи классификации
- и регрессии.



Распознавание предметов одежды

Недостатки:

У данной архитектуры **слишком быстро с ростом числа входных данных растет число параметров**, которые нужно обучить. Например, для цветного изображения размерностью 100x100 пикселей только входных параметров будет 100x100x3 = 30 000. А первый же скрытый слой хотя бы в 500 нейронов приведет к увеличению параметров до 30 000x500 = 15 000 000.

Проблема затухающего градиента. При обучении нейронной сети ошибка, которую оценивают на выходе нейронной сети, распространяется обратно по всем весам нейронной сети к ее входу. И при наличии относительно большого количества слоев ошибка ближе ко входу уменьшается до значений, близких к нулю (затухает), а значит, веса нейронной сети, которые находятся в первых слоях, перестают обучаться.

Анализ набора данных с точки зрения дальнейшего построения нейронной сети

Изображения 100x100 пикселей, значения 0-255,

RGB – 3 матрицы

На вход вектор длиной 100x100x3 = 30 000

Это – размерность входного слоя нейронной сети – это первый параметр, строго завязанный на входные данные.

Классы: медведь, волк, рысь =3 классы: число 3 строго определяет количество выходных нейронов в построенной архитектуре сети.

Базовые объекты и параметры объектов глубоких нейронных сетей в TensorFlow

- **tensorflow.keras.models.Sequential** – это базовая модель нейронной сети, которая, по сути, является контейнером для последовательно помещенных в нее слоев. Не имеет параметров;
- **tensorflow.keras.layers.Dense** – это объект полносвязного слоя нейронной сети. Определяется следующими параметрами:
  - **units** – это количество нейронов в данном слое;
  - **input_dim** – размерность входного слоя (только для слоя, непосредственно следующего за входными данными, для последующих слоев входная размерность определяется автоматически исходя из размерности предыдущего слоя);
  - **activation** – функция активации, которая будет использована для данного слоя («*relu*», «*softmax*» и др.).

Объявление простой полносвязной нейронной сети

0. Импортируем необходимые объекты

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

0. Создаем объект последовательной модели нейронной сети

```
model = Sequential()
```

0. Помещаем необходимые слои в модель

1. Слой, обрабатывающий входные данные, размерностью 784 элемента. Количество нейроно
   слое зададим 700, а функцию активации – `relu`

```
model.add(Dense(units=700,
        input_dim=784,
        activation="relu"))
```

1. Выходной слой, выдающий вероятность принадлежности к одному из 5 классов с функцией
   активации `softmax`.

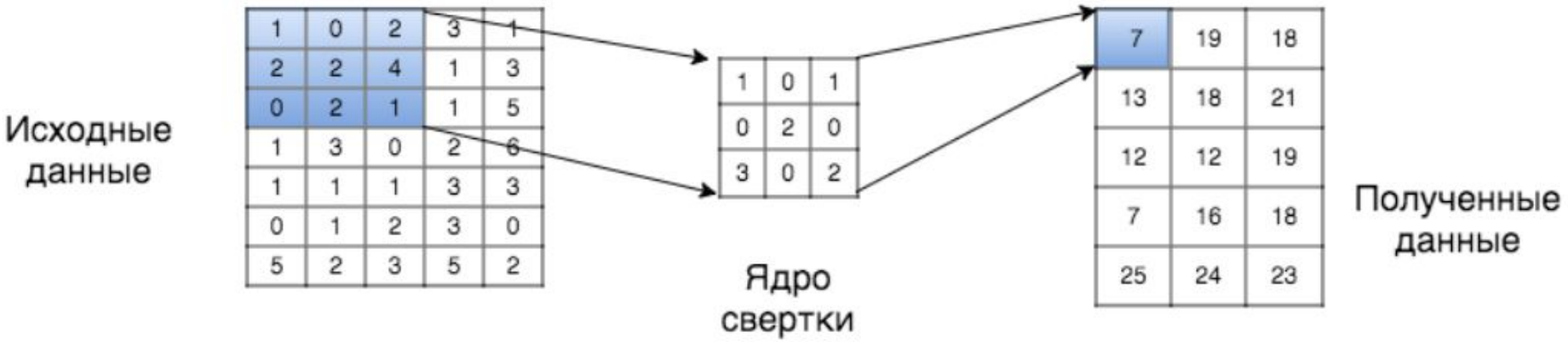
```
model.add(Dense(5, activation="softmax"))
```

Базовые объекты и параметры объектов глубоких нейронных сетей в TensorFlow

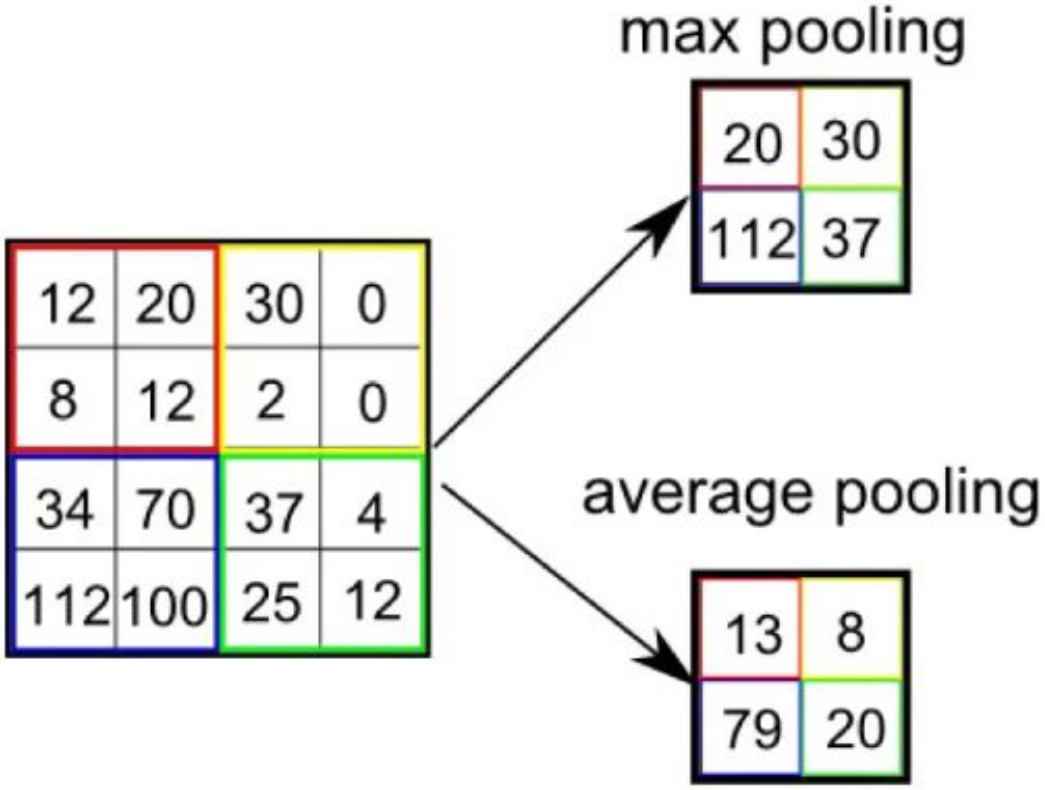
Нейронные сети для анализа изображений

- **Полносвязная нейронная сеть**
- Каждый нейрон входного слоя связан с каждым пикселем изображения, а значит, что с ростом размера изображения сильно растет и число обучаемых параметров
- Нейронная сеть работает с плоским вектором из пикселей изображения, а значит, теряет информацию о пространственной структуре данных

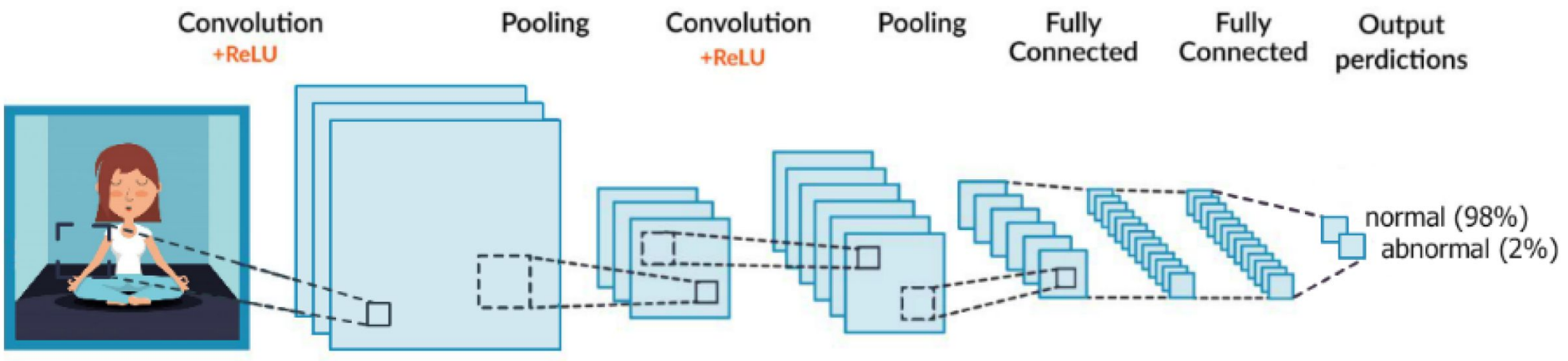
Сверточные нейронные сети



Сверточные нейронные сети

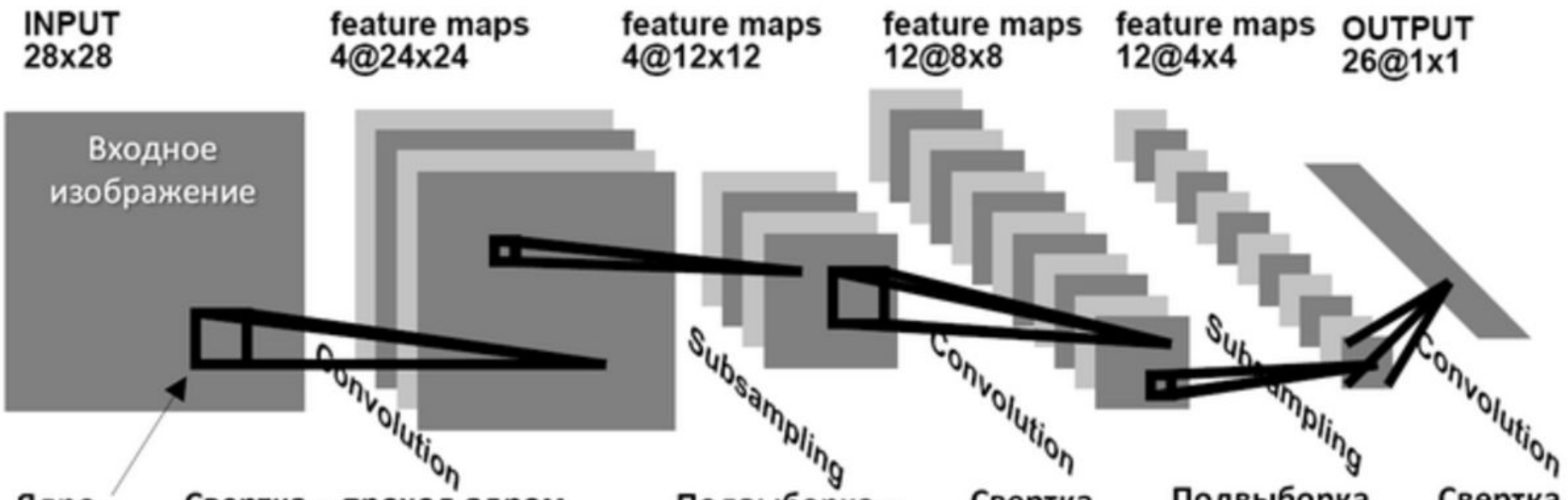


Сверточные нейронные сети



- Сама по себе сверточная нейронная сеть не способна решить задачу классификации.
- Она извлекает множество признаков как цветовых, так и пространственных, но чтобы на их основе сделать выводы, нам все равно нужен полносвязный слой, который решит задачу классификации.
- Поэтому многомерные матрицы признаков, хранимые в сверточной части сети, необходимо превратить в плоский вектор признаков и подать на вход полносвязному слою.
- Таким образом, сверточная нейронная сеть – не только сверточная.

Сверточные нейронные сети



Ядро свертки – матрица из весов, например, размером 5x5

Свертка – проход ядром свертки по всем нейронам нижнего слоя для активации нейронов верхнего слоя – получение карты признаков (feature maps). Для ядер разного типа получаются разные карты признаков.

Подвыборка – операция уменьшения размерности. Чаще всего – выбор максимума или поиск среднего по нескольким соседним нейронам.

Свертка

Подвыборка

Свертка

Нейронные сети для анализа изображений

Сверточная нейронная сеть

- Каждый нейрон входного слоя связан лишь с пикселями, попадающими в его диапазон (9, 25, 49 пикселей). А так как связь идет через ядро свертки (3x3, 5x5, 7x7 соответственно), а оно одинаково для всего изображения, то при любом размере изображения нам нужно обучить лишь значения ядра свертки. Отсюда следует, что число параметров для обучения не зависит от размера изображения
- Работает с матрицей изображения и способна учитывать пространственную структуру данных
- Каждый нейрон работает лишь с частью предыдущих пикселей (либо нейронов). Такой принцип локального восприятия позволяет уменьшить объем информации, обрабатываемой каждым нейроном.
- За счет слоев подвыборки (пулинга) применяется принцип уменьшения размерности. Таким образом с каждым следующим слоем нейронная сеть получает все более высокоуровневые признаки изображения

Обработка естественного языка

- Предварительная обработка текста, необходимо:
  - привести текст к единому регистру;
  - в зависимости от задачи провести очистку текста от лишних символов (знаки пунктуации, не очень значимые по смысловой нагрузке слова, *html/xml*-разметка и т. д.);
  - в зависимости от задачи провести *токенизациию* текста, то есть разбить монолитный блок текста на атомарные составные части: абзацы, предложения, слова, *n*-граммы символов;
  - провести разметку слов по частям речи (чтобы различать, например, слово «село» в предложениях «солнце село» и «маленькое село»);
  - привести слова в тексте к нормальной форме: *лемматизация* (приведение слова к словарной форме), *стемминг* (нахождение основы слова);
  - провести процесс преобразования текста в цифровое представление – векторизацию.

Архитектуры нейронных сетей языка

- сверточные одномерные нейронные сети (*CNN 1D*), работающие по принципу двумерных сверточных сетей, но анализирующие плоские структуры текста;
- рекуррентные нейронные сети (*RNN*), основанные на том, что «помнят» не только информацию о текущем примере, на котором обучаются, но и о предыдущих (что очень важно для текстовой информации, где важна сама последовательность слов, а не отдельные слова).

Архитектуры нейронных сетей языка

- Модели векторных представлений слов и документов можно разделить на три блока:
  - частотный подход;
  - тематическое моделирование;
  - дистрибутивная семантика.

Частотный подход

- Основан на подходе *Bag of words* («мешок слов»):
  1. Находим все уникальные слова в тексте.
  2. Присваиваем каждому слову порядковый номер от 1 до *N* (*N* – число уникальных слов в документе).
  3. Меняем в исходном документе все слова на их порядковый номер из шага п. 2.
- Пример. Есть два предложения, которые уже очищены от знаков пунктуации и токенизированы на отдельные слова:
- [[«сегодня», «был», «чудесный», «день»], [«сегодня», «был», «мой», «день», «рождения»]]Тогда можно создать вот такой словарь:
- { «сегодня»: 1, «был»: 2, «чудесный»: 3, «день»: 4, «мой»: 5,

Частотный подход

- Нейронная сеть должна получать на вход последовательности одной длины.
- Подходы для решения:
  1. *One Hot Encoding* (*OHE*);
  2. *TF-IDF.*

- Оба эти подхода представляют текст в виде вектора, длина которого равна длине словаря.

Частотный подход

- *OHE* представляет из себя разреженный вектор, в котором стоят 0 на позициях тех слов из словаря, которых нет в тексте и 1 на местах тех слов из словаря, которые есть в тексте.
- Возвращаясь к нашему примеру:
- [[1, 1, 1, 1, 0, 0], [1, 1, 0, 1, 1, 1]].

Тематическое моделирование

- Семантическое моделирование текстов.
- делим тексты на разные кластеры, но «мягко».
- каждое слово может определять несколько тем (топиков), но на отнесение текста к разным темам влияет различным образом.
- Например, слово «рецепт» может с большой вероятностью говорить о том, что текст относится к теме «Кулинария», но также может с некой меньшей долей вероятности отнести текст и к теме «Медицина».

Тематическое моделирование

- К данной группе методов относятся:
  - вероятностный латентно семантический анализ (*PLSA, Probabilistic latent semantic analysis*). Основан на скрытом семантическом анализе и выявлении скрытых тем в корпусе текстов на основе сингулярного разложения;
  - латентное размещение Дирихле (*LDA, Latent Dirichlet Allocation*). Основано на генеративной вероятностной модели, извлекающей скрытые темы из корпуса текстов.

DALL-E 2 (2022)

- Генерировать изображения по текстовому описанию на английском языке;
- «Дорисовать» картину, расширив изображение за его исходные пределы;
- Вносить изменения, добавляя объекты;
- Создавать несколько вариантов похожих изображений на основе оригинала.

3D-рендер левитирующего футуристического замка в небе, цифровое искусство

Ковбой, прогуливающийся по освещенным неоновым светом улицам и переулкам футуристического Токио, окутанного густым туманом

Imaginary soundscape (2018)

https://imaginarysoundscape.net/

- Озвучивать случайное место на земле на Google Map;
- Озвучивать изображения.

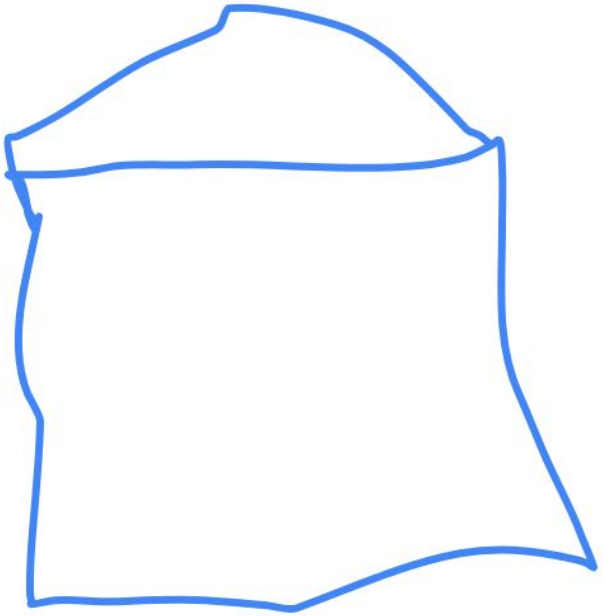
Boating on the River
Emilio Sanchez-Perrier

Tokyo Dome
Japan

Der Eichenwald
Robert Zund

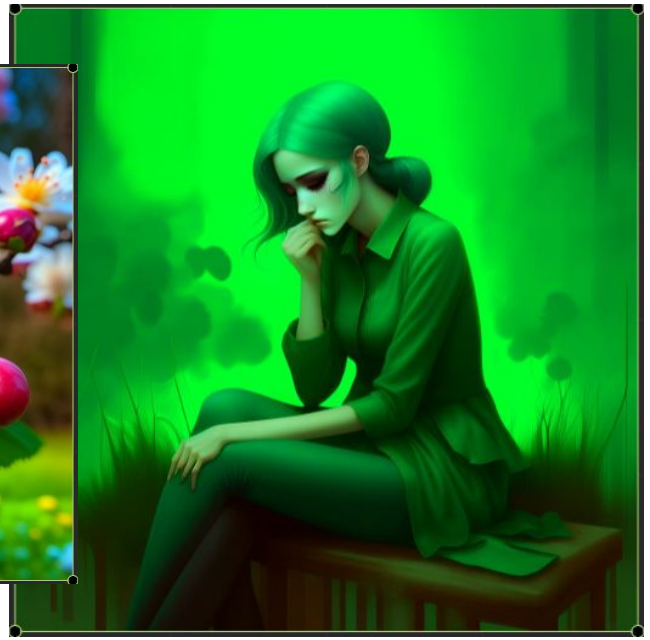
Autodraw

- https://www.autodraw.com/
- **Как пользоваться:** интерфейс сервиса напоминает упрощенный Paint. Из кнопок есть кисть, автокисть, текст, заполнение, фигуры и выбор цвета. Пользователь выбирает автокисть и начинает рисовать фигуру — программа автоматически пытается «угадать», что имеет в виду юзер, и предлагает похожие готовые фигурки.



Kandinsky 2.1

- https://editor.fusionbrain.ai/
- Kandinsky 2.1 – новейшая разработка от Сбера, способная генерировать уникальные изображения на основе текстового запроса. Разработчики заявляют о поддержке более 100 языков, в том числе и русского, чем не могут похвастаться англоязычные Midjourney или Dall-e 2. Нейросеть отлично распознает русскоязычные запросы, что влияет на качество генерируемых картинок.



TurboText

- https://turbotext.pro/
- Нейросеть TurboText тоже помогает в работе с текстами. Она может написать любую статью, содержательные отзывы, описания товаров, а также грамотно переведет текст на любой язык. Сервис будет полезен владельцам интернет-магазинов, журналистам, копирайтерам, блогерам и студентам.

Вопрос: крыло самолета истребителя

Крыло самолета истребителя является одной из ключевых компонентов его конструкции. Оно выполняет несколько важных функций:

1. Поддержка полета: Крыло создает подъемную силу, необходимую для поддержания самолета в воздухе. Оно генерирует аэродинамическую силу, которая противодействует силе тяжести и позволяет самолету подниматься и плавно маневрировать.

2. Управление полетом: Крыло осуществляет контроль над направлением и углом атаки самолета. Путем изменения угла атаки и используя аэродинамические поверхности, такие как закрылки и элероны, пилот может управлять полетом и выполнением маневров.

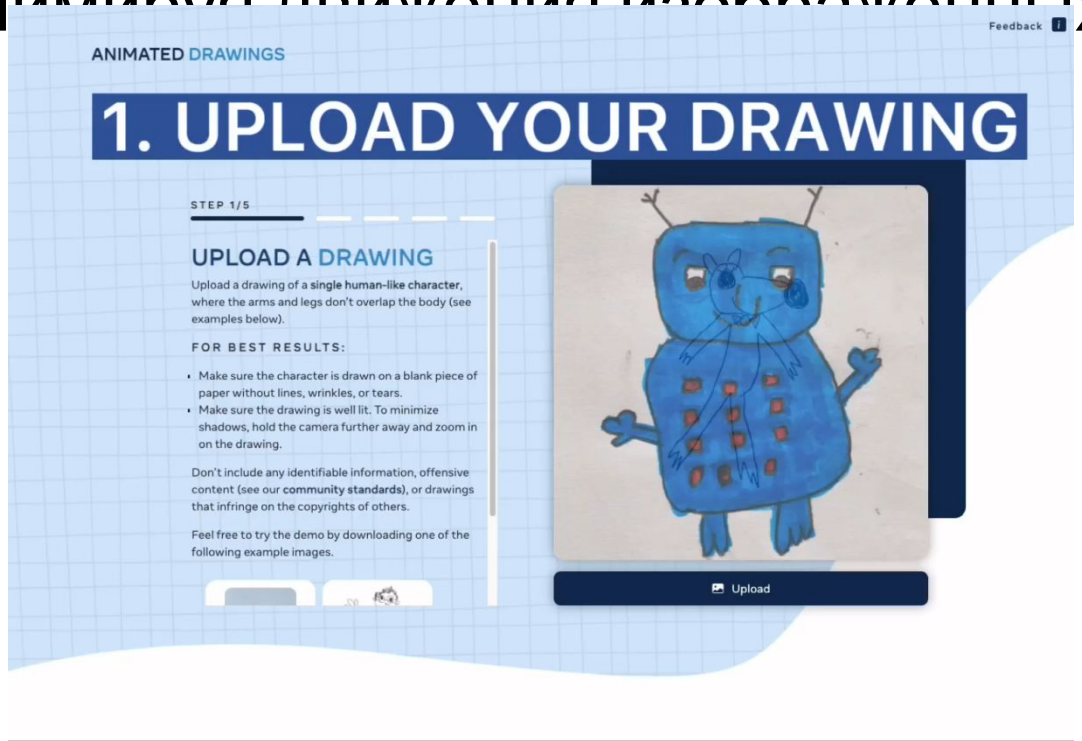
3. Хранение топлива: В крыле самолета истребителя часто размещаются топливные баки. Это позволяет увеличить дальность полета и обеспечить необходимое количество топлива для выполнения задач.

4. Размещение систем и оборудования: Крыло может использоваться для размещения различных систем и оборудования, таких как оружие, радары, электроника и другие компоненты, необходимые для выполнения боевых задач.

Конструкция крыла истребителя может быть различной, включая разные формы, размеры и аэродинамические характеристики, в зависимости от конкретной модели самолета и его задач.

Sketch Metademolab

- https://sketch.metademolab.com/
- сервис, который позволяет оживлять детские рисунки.
- Находится в бесплатном онлайн-доступе для всех желающих.
- Что умеет делать: оживляет скетчи и детские рисунки, анимируя движения изображенных существ.



Riffusion

музыкальная нейросеть, созданная Сетом Форсгреном и Айком Мартиросом.

https://www.riffusion.com/?&prompt=emotional+disco

- Что умеет: генерирует мелодии по текстовым подсказкам.
- Как пользоваться: сервис максимально прост в использовании — в строке с текстом нужно описать мелодию, которую хочет услышать пользователь. После того, как она будет сгенерирована, — нажать на play и прослушать