Введение в глубокое обучение. Часть 1

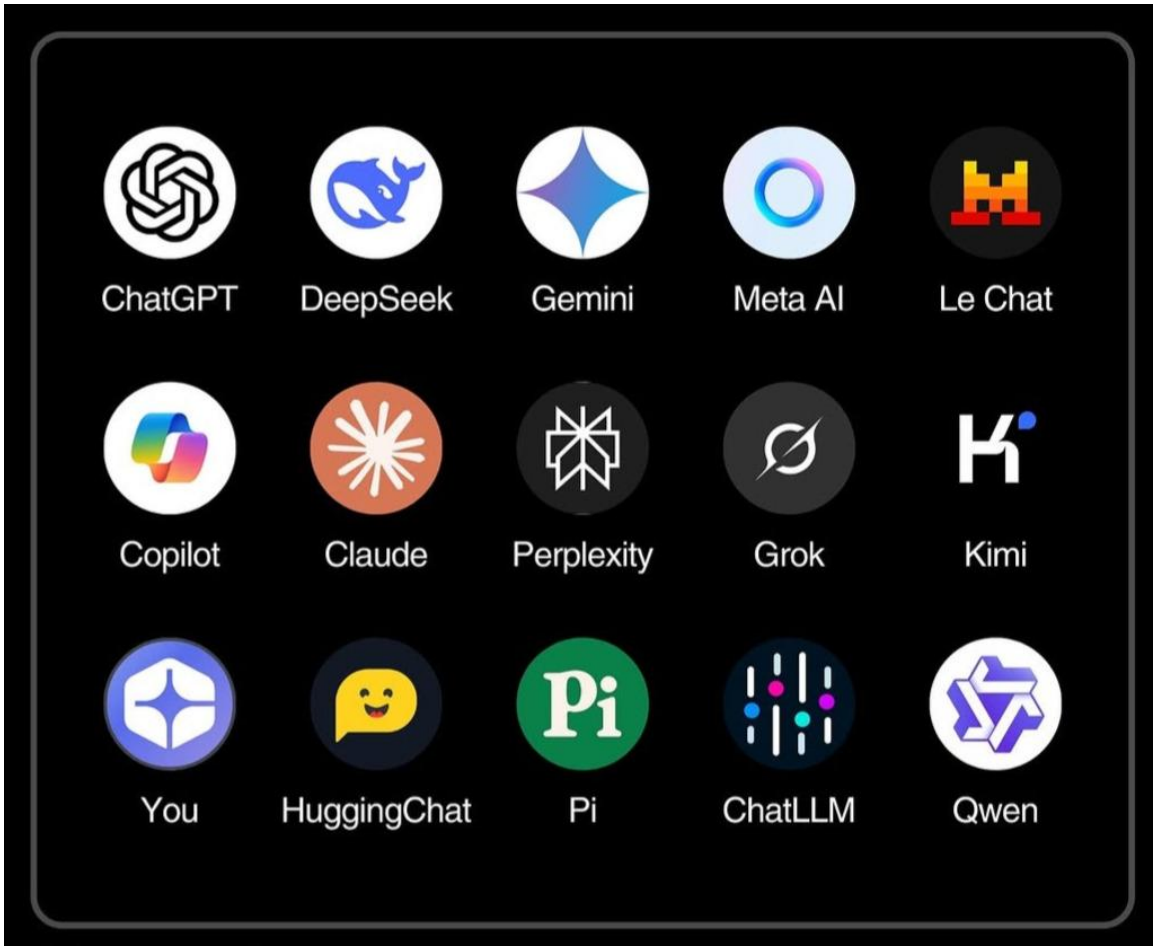
Машинное обучение

Богданов Александр

ИСП РАН

15 мая 2025

Чат-боты



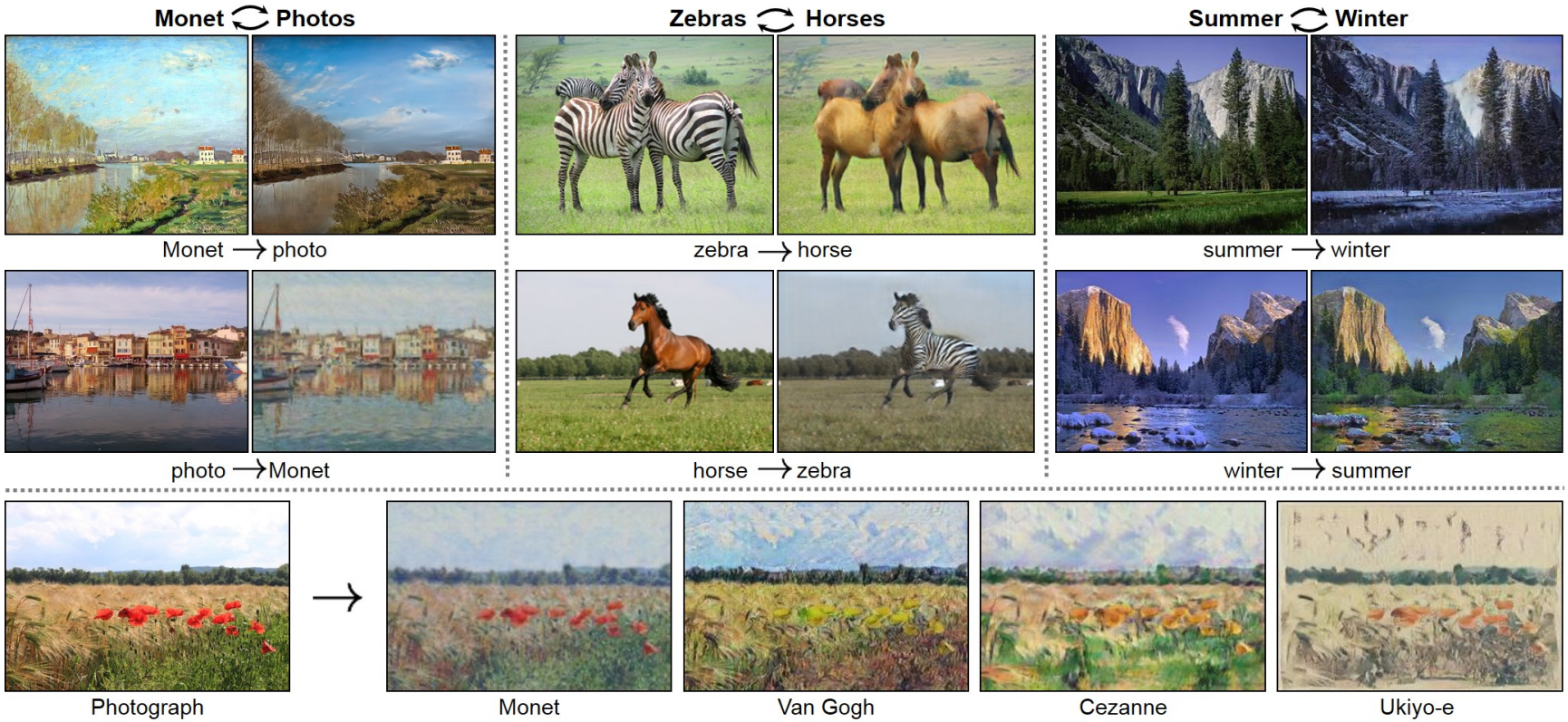
Пример: https://chatgpt.com

Генерация изображений

Figure: Бегемот в пальто

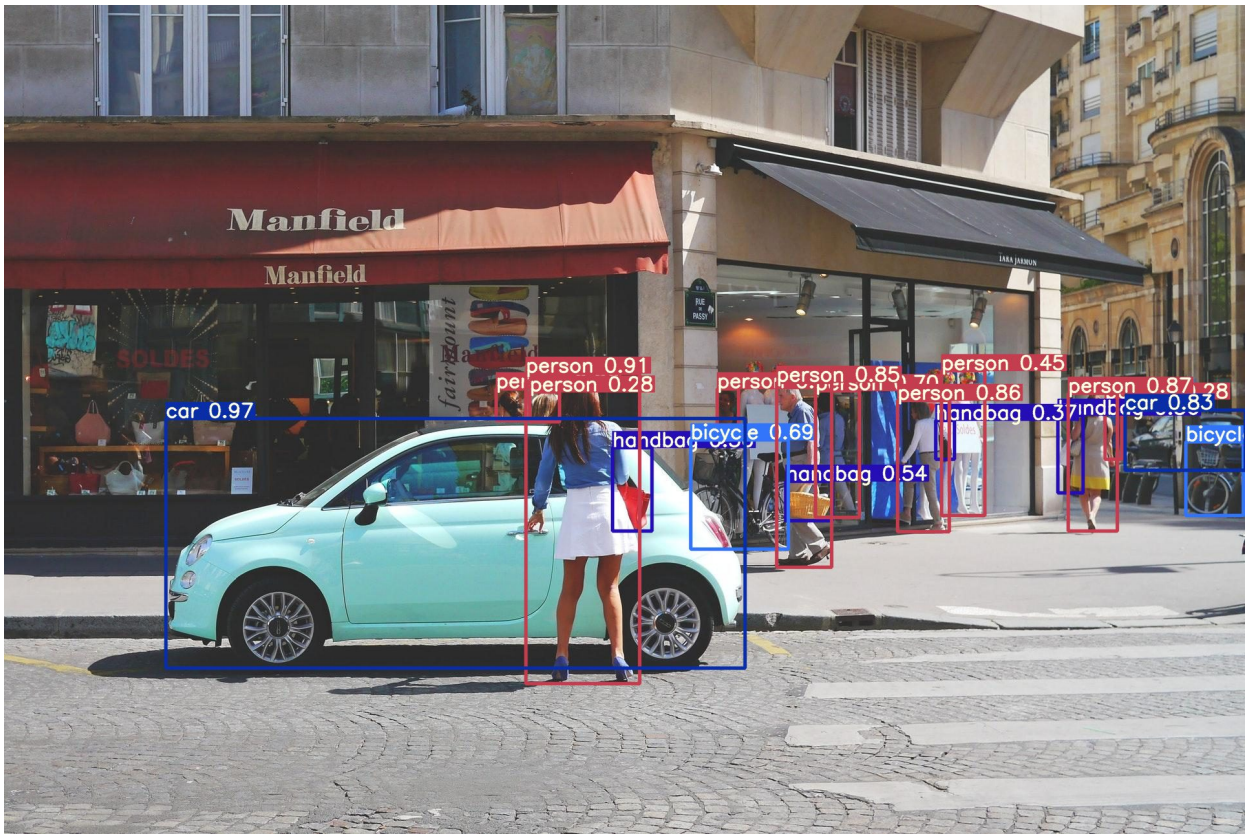
Пример: https://shedevrum.ai

Наложение стиля



Пример: https://junyanz.github.io/CycleGAN/

Детекция объектов



Пример: https://viso.ai/deep-learning/object-detection/

Обработка звука

Пример: https://fjiang9.github.io/NKF-AEC/

Игры

Пример: https://www.youtube.com/watch?v=sMHr0H15Ubg
https://ale.farama.org/environments/

Самоуправляемые автомобили

Пример: https://www.youtube.com/shorts/Q0r5YJJTj8w

Что такое нейронная сеть?

Нейронная сеть — это сложная дифференцируемая функция, задающая отображение из исходного признакового пространства в пространство ответов.

Нейросети были придуманы еще в 1970 годах, но их развитие началось примерно с 2012 года.

Причины возникновения нейронных сетей

Причины возникновения нейронных сетей

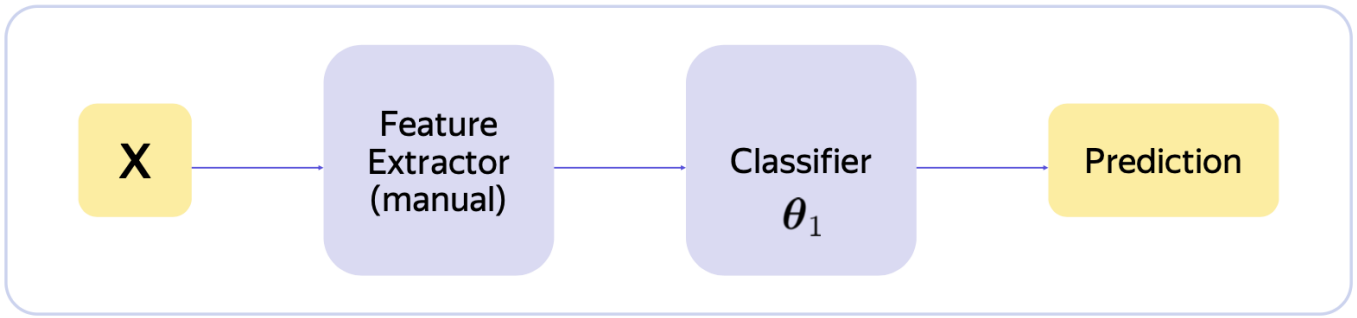
- Стремление к переходу от построения сложных пайплайнов, каждая компонента которых тренируется сама по себе решать кусочек задачи, к end-to-end обучению всей системы, как одного целого.

Причины возникновения нейронных сетей

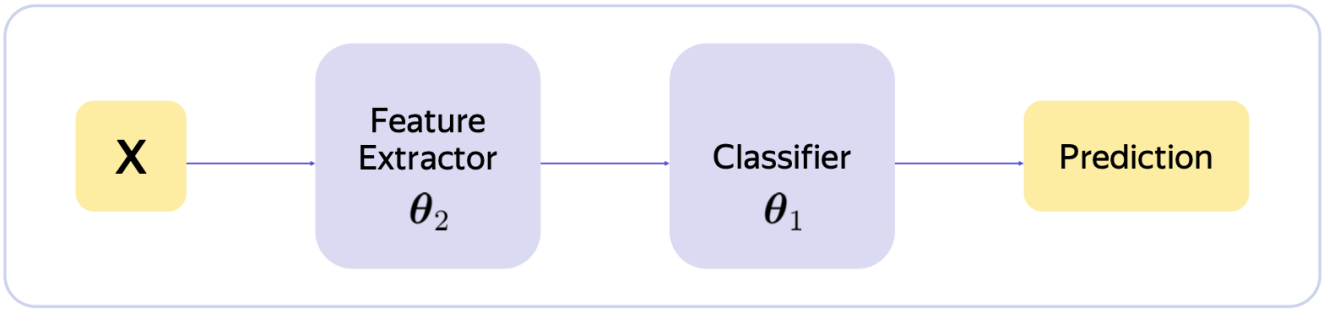
- Стремление к переходу от построения сложных пайплайнов, каждая компонента которых тренируется сама по себе решать кусочек задачи, к end-to-end обучению всей системы, как одного целого.
- Автоматизация процесса отбора признаков.

Различие пайплайнов

Классический



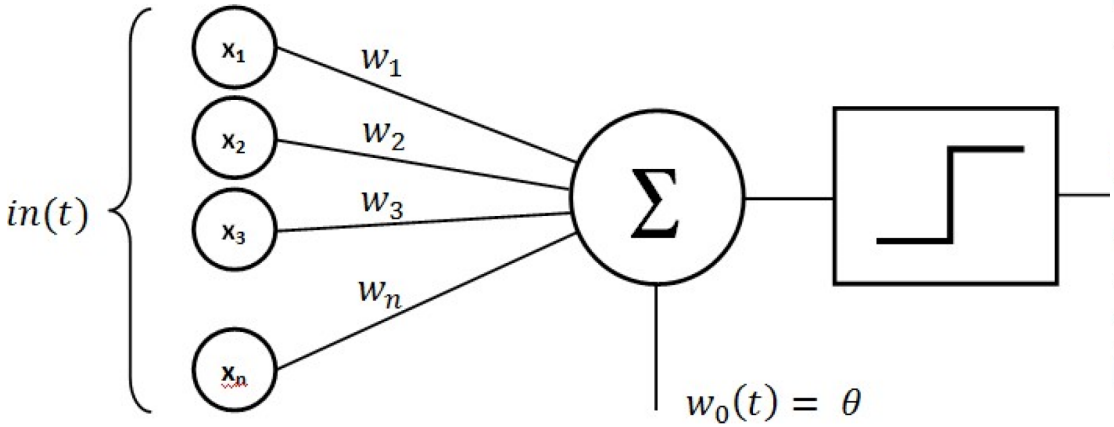
Нейросетевой

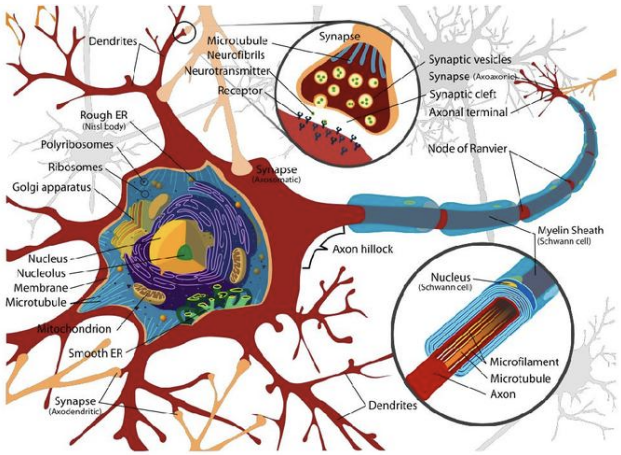


Математическая модель нейрона

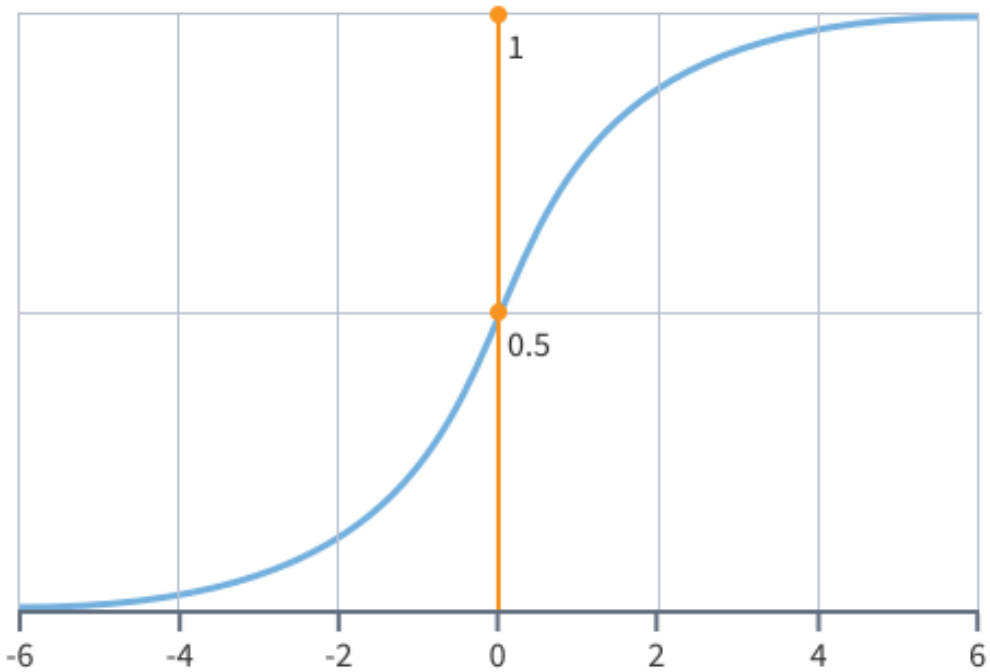
$$a(x, w) = \sigma(\langle x, w \rangle) = \sigma\left(\sum_{i=1}^{n} w_i f_i(x) - w_0\right)$$

- $\sigma(z)$ — функция активации (нелинейная);
- w_i — весовые коэффициенты синаптических связей;
- w_0 — порог активации;
- $f_i(x) \equiv x_i$ — обобщение.





Модель логистической регрессия



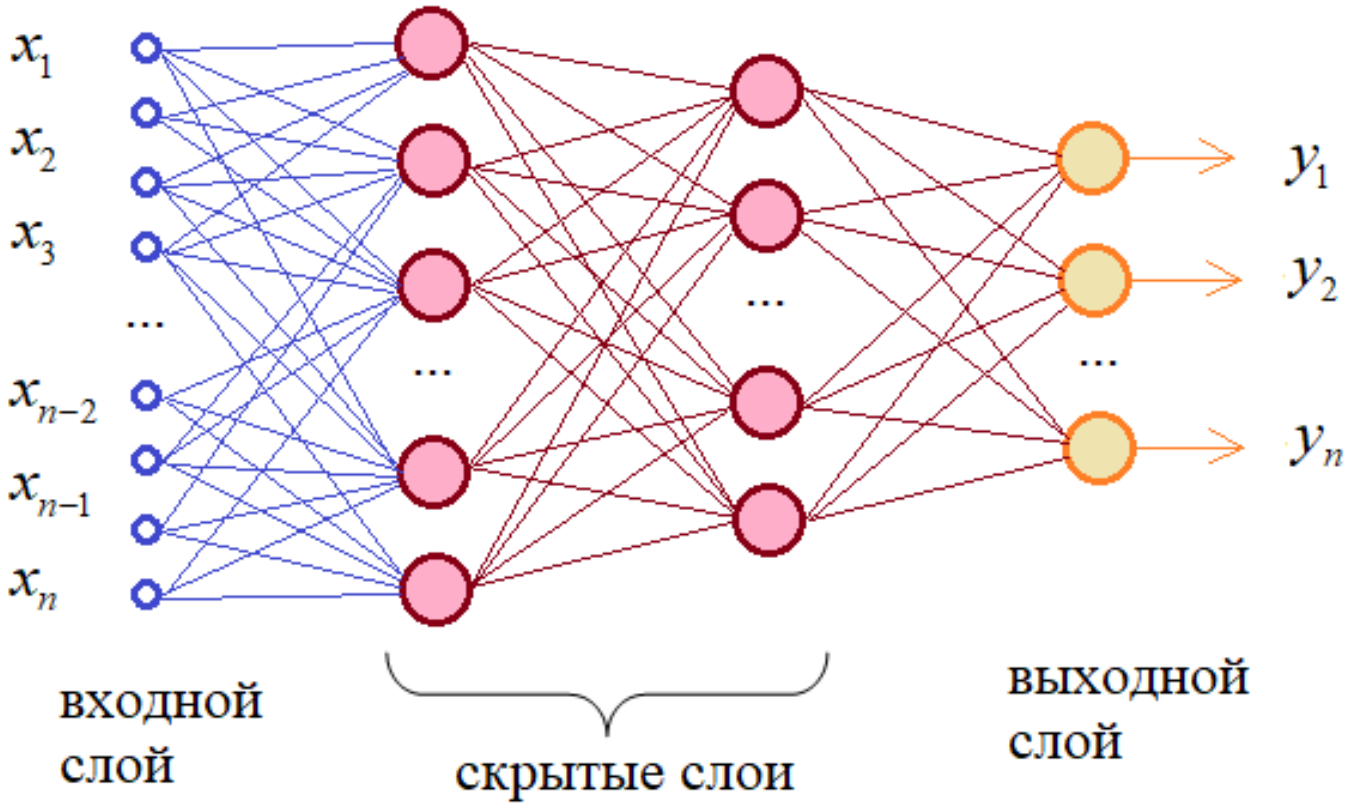
$$a(x, w) = \sigma(\langle x, w \rangle) = \frac{1}{1 + e^{-\langle x, w \rangle}}$$

Линейный слой

Линейный слой — линейное преобразование над входящими данными. Его обучаемые параметры — это матрица W и вектор b:

$$x \to Wx + b, \quad W \in \mathbb{R}^{k \times d},\ x \in \mathbb{R}^{d},\ b \in \mathbb{R}^{k}.$$

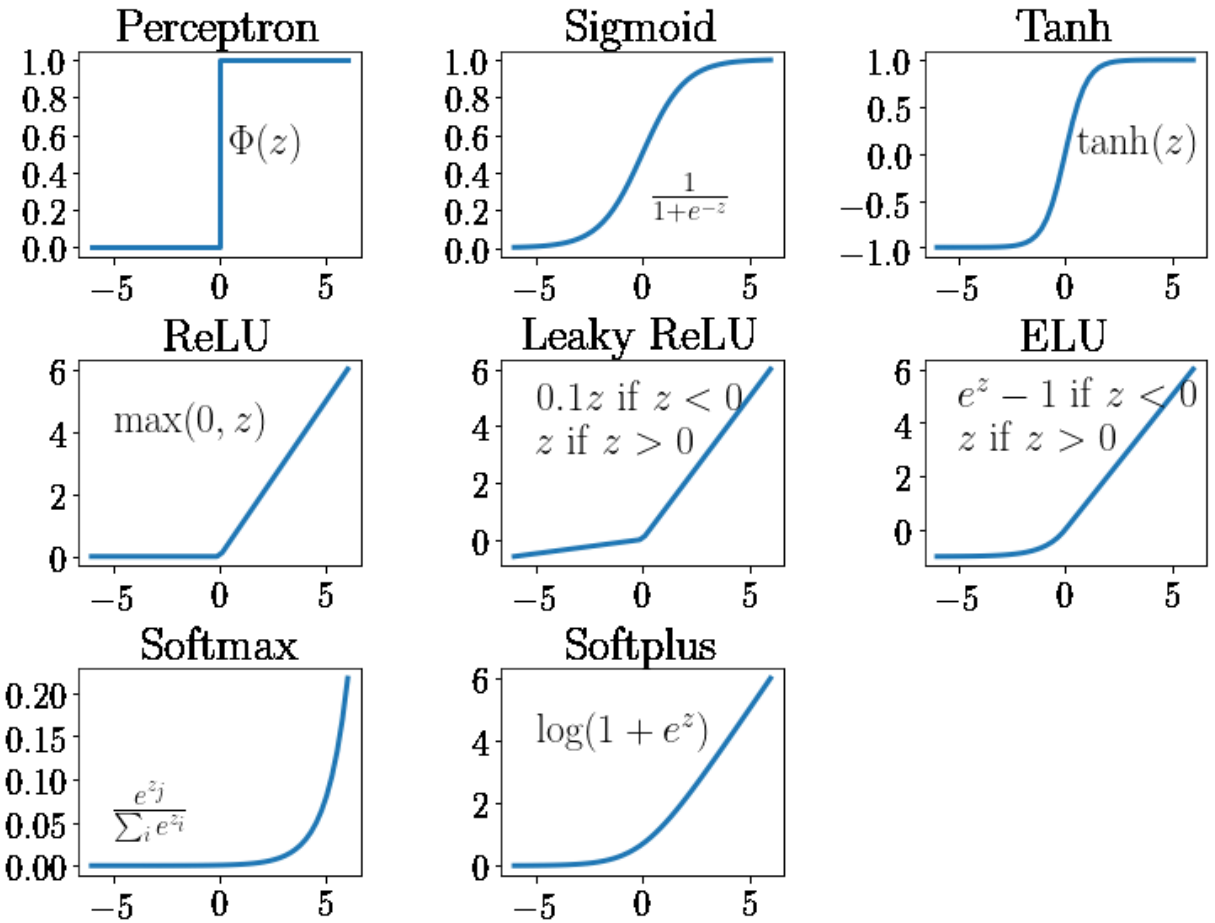
Полносвязанная нейронная сеть



Функции активации

Функция активации — нелинейное преобразование, поэлементно применяющееся к пришедшим на вход данным. Благодаря функциям активации нейронные сети способны порождать более информативные признаковые описания, преобразуя данные нелинейным образом.

Функции активации



Теорема Цыбенко

Теорема

Если $\sigma(z)$ — сигмоида, тогда для любой непрерывной на $[0,1]^n$ функции $f(x)$ существуют такие значения параметров H, $\alpha_h \in \mathbb{R}$, $w_h \in \mathbb{R}^n$, $w_0 \in \mathbb{R}$, что двухслойная сеть

$$a(x) = \sum_{h=1}^{H} \alpha_h \sigma(\langle x, w_h \rangle - w_0)$$

равномерно приближает $f(x)$ с любой точностью ε:

$$|a(x) - f(x)| < \varepsilon, \quad \text{для всех } x \in [0,1]^n.$$

Пайплайн обучения

Пайплайн обучения

- Подготовить данные;

Пайплайн обучения

- Подготовить данные;
- Загрузить данные в Dataloader;

Пайплайн обучения

- Подготовить данные;
- Загрузить данные в Dataloader;
- Подготовить модель;

Пайплайн обучения

- Подготовить данные;
- Загрузить данные в Dataloader;
- Подготовить модель;
- Выбрать подходящий loss;

Пайплайн обучения

- Подготовить данные;
- Загрузить данные в Dataloader;
- Подготовить модель;
- Выбрать подходящий loss;
- На каждой итерации делается backpropogation;

Пайплайн обучения

- Подготовить данные;
- Загрузить данные в Dataloader;
- Подготовить модель;
- Выбрать подходящий loss;
- На каждой итерации делается backpropogation;
- На каждом шаге делается шаг оптимизатора;

Данные

С помощью нейронных сетей мы решили проблему с выделением признаков. Но осталась проблема, которая мешает ускорению развития нейронных сетей, какая?

Данные

С помощью нейронных сетей мы решили проблему с выделением признаков. Но осталась проблема, которая мешает ускорению развития нейронных сетей, какая?

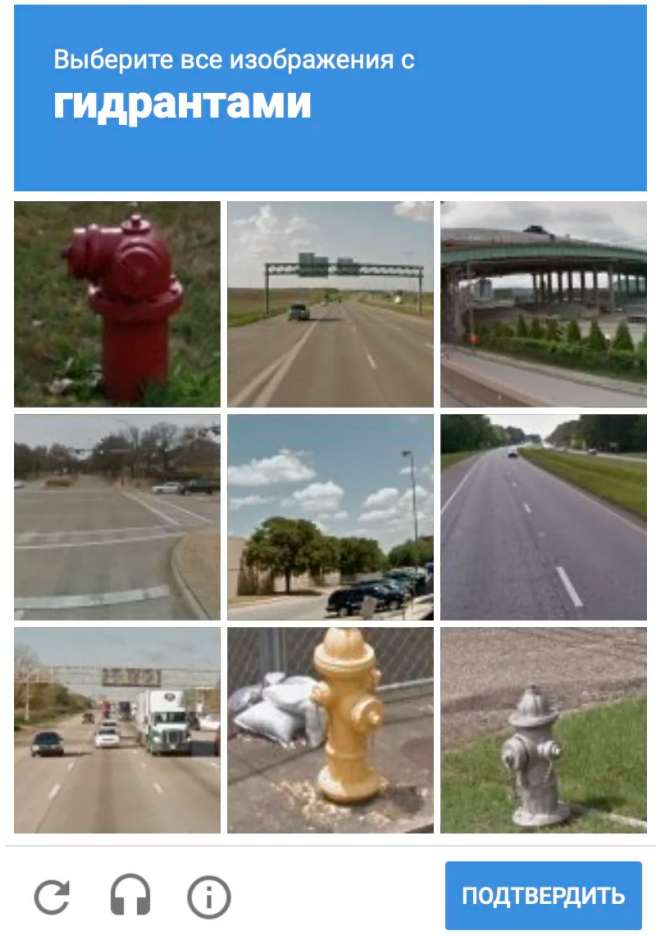
Очень скромное количество размеченных данных. При этом большинство больших датасетов закрытые, а владеют ими большие компании.

Данные

Как размечают данные большие компании?

Данные

Как размечают данные большие компании? Капчи



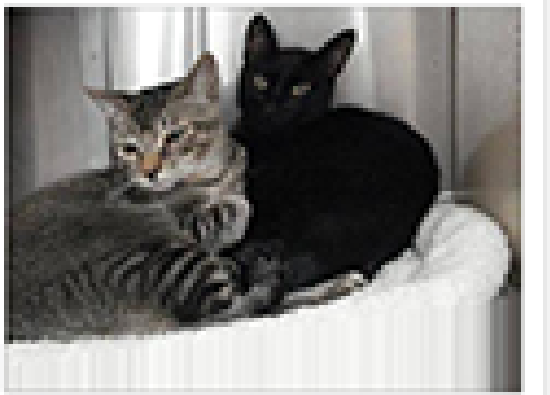
Данные

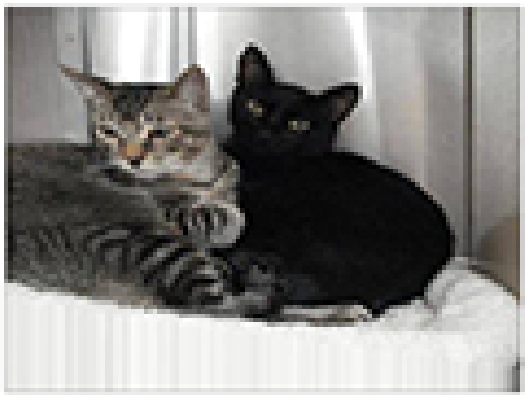
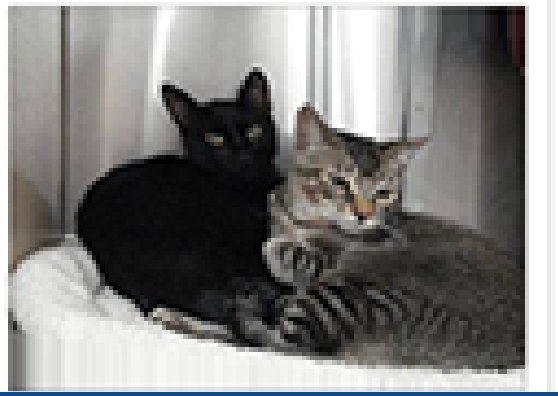
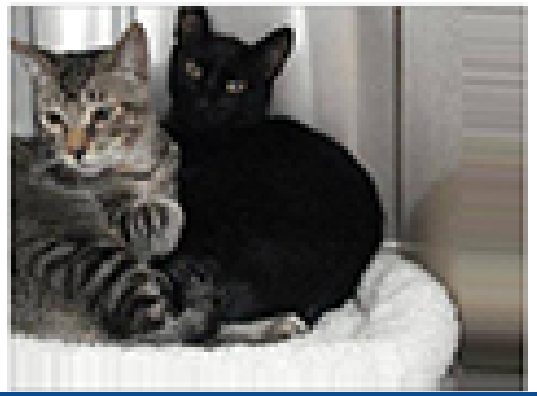
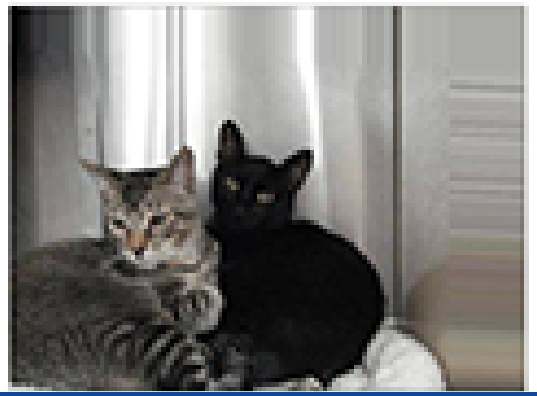
Как можно искуственно увеличивать количество данных?

Данные

Как можно искуственно увеличивать количество данных?

Аугментации.



Dataloader

Dataloader — компонент, подающий данные в модель во время обучения. Что он позволяет?

Dataloader

Dataloader — компонент, подающий данные в модель во время обучения. Что он позволяет?

- Перемешивать данные;
- Указывать число потоков;
- Изменять размер батча;

Dataloader

Dataloader — компонент, подающий данные в модель во время обучения. Что он позволяет?

- Перемешивать данные;
- Указывать число потоков;
- Изменять размер батча;

Для чего нужно разбиение данных на батчи?

Dataloader

Dataloader — компонент, подающий данные в модель во время обучения. Что он позволяет?

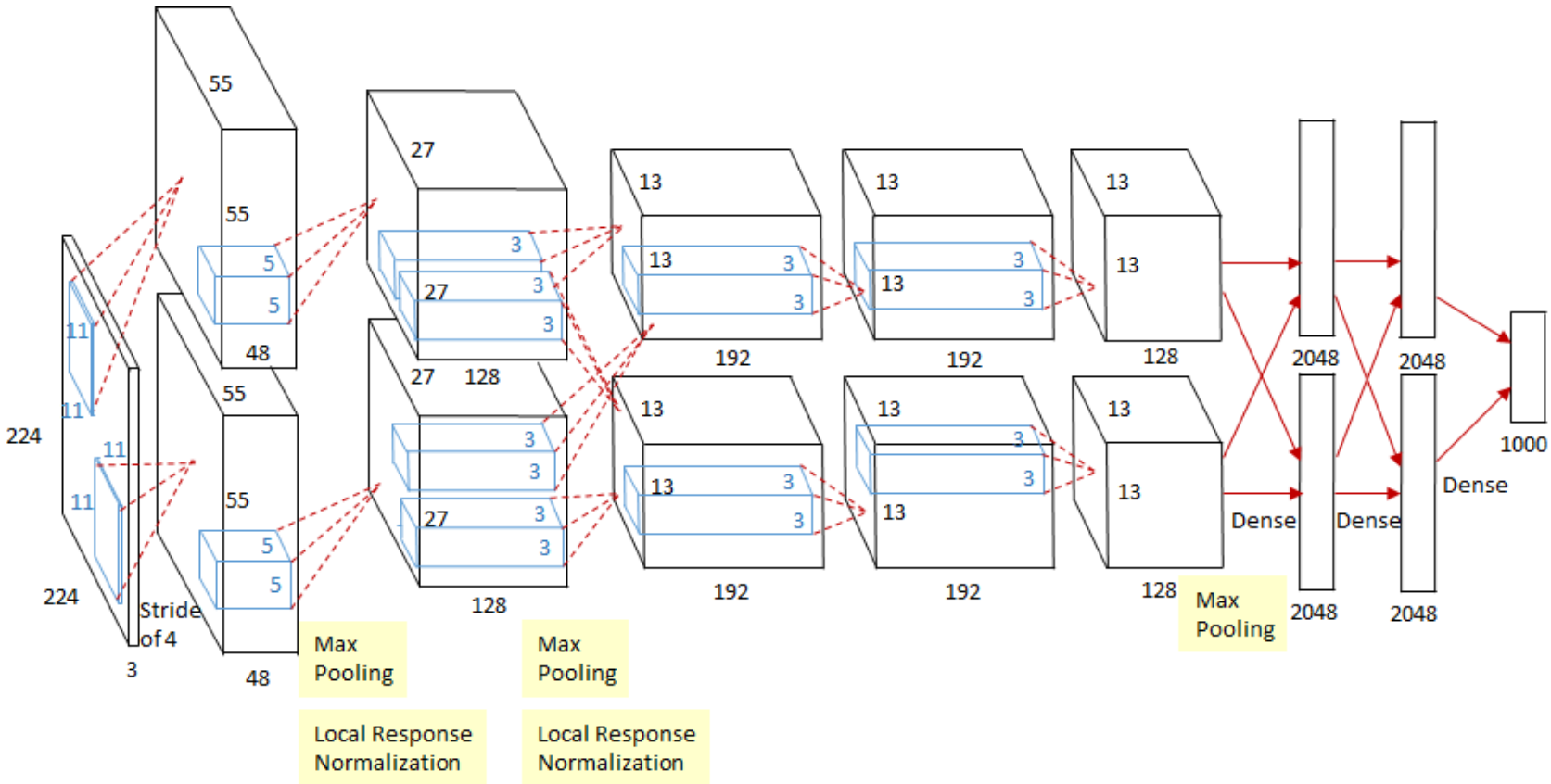
- Перемешивать данные;
- Указывать число потоков;
- Изменять размер батча;

Для чего нужно разбиение данных на батчи?

- Ускоряет обучение за счет векторизации;
- Стабилизирует обновление градиентов.

Модель

Модель — конструктор, который собирается из слоев.
Например, AlexNet:



Loss

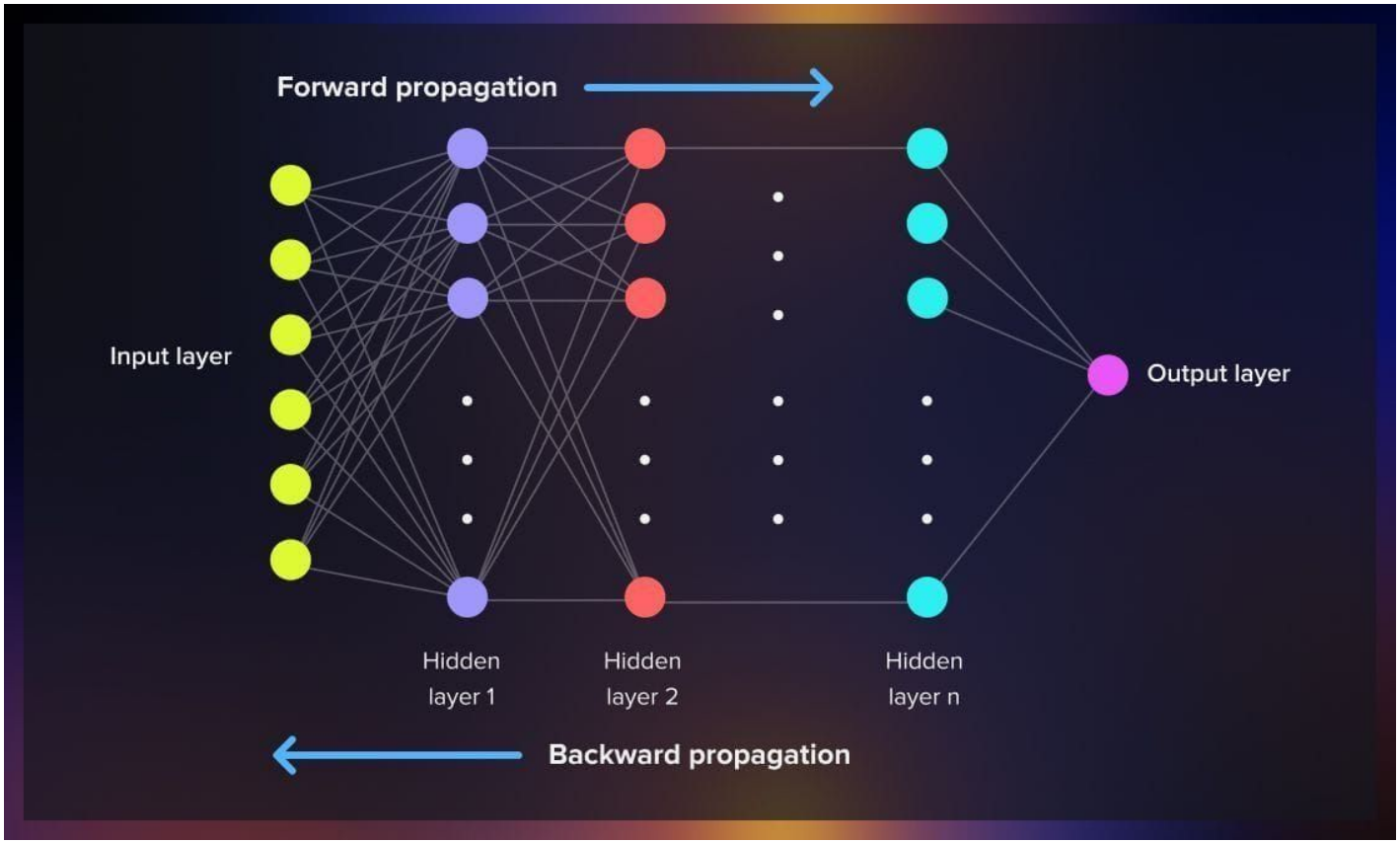
Классификация (Cross-entropy):

$$\mathcal{L}_{\text{ce}}(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log \hat{y}_i.$$

Регрессия (MSE):

$$\mathcal{L}_{\text{mse}}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Backpropogation



Оптимизатор

Алгоритм 1 Adam

Вход: шаг $D_i > 0$, параметры сглаживания $\beta_1 = 0.9$ и $\beta_2 = 0.99$, стартовая точка $x^0 \in \mathbb{R}^d$, сглаженная сумма квадратов градиентов $G_i^{-1} = 0$, сглаженная сумма градиентов $v^{-1} = 0$, параметр сглаживания $\varepsilon =$ 1e-8, количество итераций K

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: Вычислить $g^k \in \partial f(x^k)$
3: Вычислить $v^k = \beta_1 v^{k-1} + (1-\beta_1) g^k$
4: Вычислить $\hat{v}^k = v^k / \left(1 - \beta_1^{k+1}\right)$
5: Для каждой координаты: $G_i^k = \beta_2 G_i^{k-1} + (1-\beta_2)(g_i^k)^2$
6: Вычислить $\hat{G}^k = G^k / \left(1 - \beta_2^{k+1}\right)$
7: Для каждой координаты: $x_i^{k+1} = x_i^k - \frac{D_i}{\sqrt{\hat{G}_i^k + \varepsilon}} \hat{v}_i^k$
8: **end for**

Выход: $\frac{1}{K} \sum_{k=0}^{K} x^k$

Инференс

Есть ли отличие между состоянием модели во время обучения и во время инференса?

Инференс

Есть ли отличие между состоянием модели во время обучения и во время инференса?
Да, есть. Мы хотим, чтобы некоторые слои работали по-разному в разных режимах. Но об этом уже в следующей лекции . . .

Конец

Спасибо за внимание!