

Data science

Лекция 5. Глубокое обучение и нейросети

2025/2026 учебный год

Доцент кафедры МО&МО, Махно В.В.

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



От линейных моделей к нейронным сетям

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.

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**.

Нейронные сети

Нейронная сеть — это **модель**, которая умеет **извлекать сложные зависимости** из данных (устройство мозга, состоит из **искусственных нейронов**, соединённых между собой и обучающихся на примерах).

Нейросети умеют:

- распознавать изображения
- переводить текст
- генерировать музыку
- водить машины

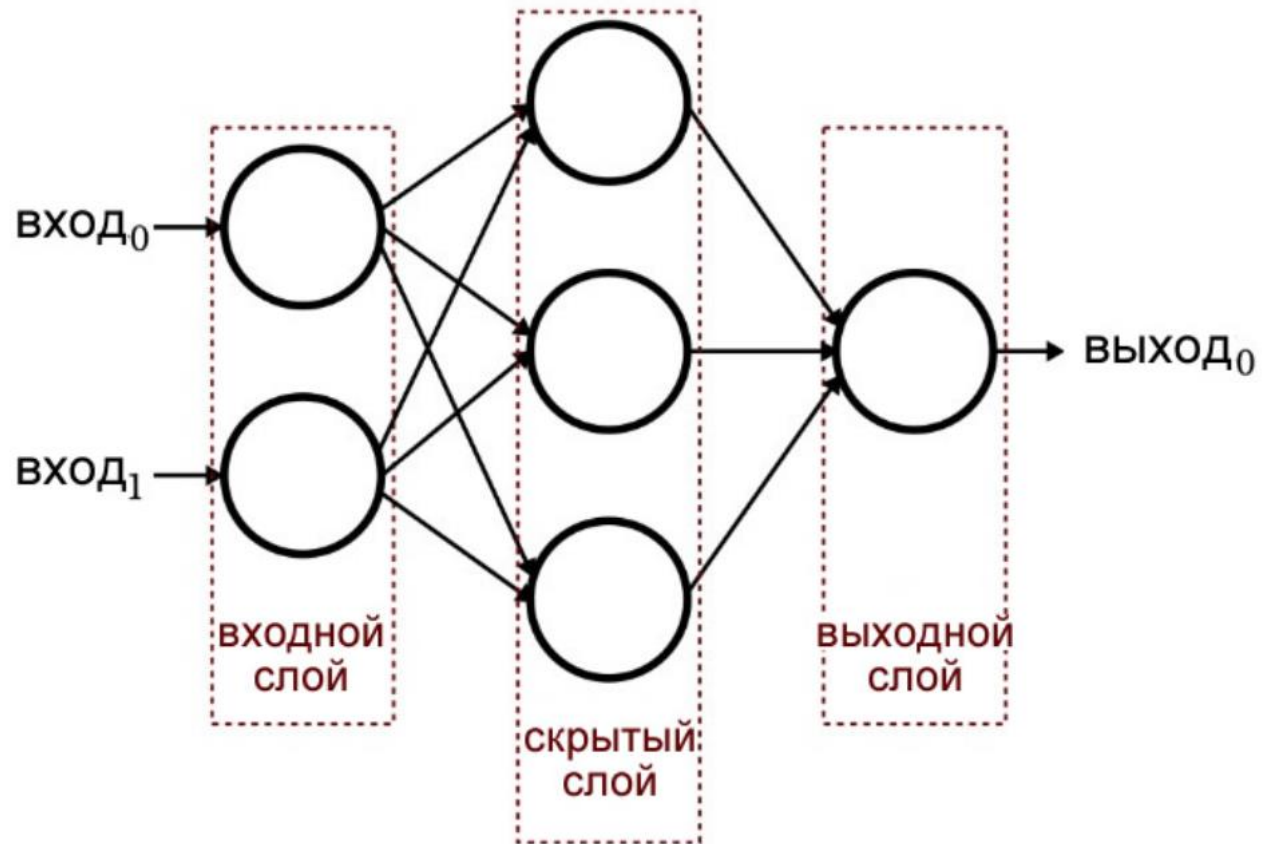
Нейросеть — это **слои нейронов**, через которые проходят данные.

Входные данные → Скрытые слои → Выход

Нейронные сети

Каждый **нейрон**:

- получает на вход числа (признаки),
- умножает их на веса,
- применяет функцию активации,
- передаёт результат дальше.



Компоненты нейронной сети

Элемент	Что делает
Нейрон	Базовая вычислительная единица
Вес (weight)	Показывает "важность" признака
Функция активации	Добавляет нелинейность (ReLU, Sigmoid и т.д.)
Слои	Состоит из множества нейронов
Обратное распространение (backprop)	Механизм обучения сети

ОБУЧЕНИЕ НС

Данные проходят через сеть → получаем предсказание

Считается **ошибка (loss)**

Ошибка **распространяется назад** по сети

Весы корректируются с помощью **градиентного спуска**

Повторяем много раз → сеть "обучается"

Нейронные сети

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model = Sequential([
    Dense(8, activation='relu', input_shape=(4,)),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32)
```

Недостаток	Что это значит
Сложность	Трудно интерпретировать результат
Объём	Требуют много данных
Вычисления	Нужны GPU и много ресурсов
Риск переобучения	Без регуляризации могут "запомнить", а не обобщить

Backpropagation (обратное распространение ошибки)

- *Нейросеть делает предсказание → сравнивает с правильным ответом → считает ошибку → и распространяет эту ошибку назад, чтобы обновить веса сети и сделать предсказание лучше в следующий раз.*

Этапы работы Backpropagation

Прямой проход (Forward Pass)

- Данные проходят через сеть:
вход → скрытые слои → выход
Получаем предсказание \hat{y} .

Вычисление ошибки (Loss)

- Сравниваем \hat{y} с правильным y :
Например, используя **MSE** или **binary cross-entropy**.

$$\text{Loss} = \frac{1}{2}(y - \hat{y})^2$$

Обратный проход (Backpropagation)

Вычисляется влияние каждого веса на ошибку, используя правила дифференцирования (цепное правило).
То есть: как нужно изменить вес w , чтобы ошибка уменьшилась?

$$\frac{\partial \text{Loss}}{\partial w}$$

Backpropagation (обратное распространение ошибки)

Обновление весов (Gradient Descent)

Используем **градиентный спуск**:

$$w := w - \eta \cdot \frac{\partial \text{Loss}}{\partial w}$$

где:

w — вес

η (learning rate) — насколько сильно менять вес

Backpropagation (обратное распространение ошибки)

Каждый нейрон знает, насколько он "виноват" в ошибке и может **подстроить свои веса**, чтобы в следующий раз ошибка была меньше

Пример

Допустим, сеть предсказала $\hat{y} = 0.8$, а правильный ответ $y = 1.0$.

Ошибка: 0.2

Сеть "понимает", что нужно выдать значение ближе к 1.0

Сеть начинает:

- двигаться **по направлению, где ошибка уменьшается** (градиент)
- **обновлять веса** по всем слоям **от конца к началу**

Почему это важно?

Без Backpropagation нейросети не учились бы

Это основа **всех современных моделей**, включая GPT, CNN, GAN и др.

Работает в связке с **градиентным спуском**

Виды нейросетей и какие задачи они решают

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