# Transfer Learning

# Большая модель для маленькой задачи

Большие модели могут выучить много информации
=> логично использовать их для всех задач

Большие модели требуют много данных для обучения
На маленьких задачах они переобучаются

[Как решить эту проблему?]

# Transfer learning

**Transfer learning** – перенос знаний обученной модели на новую задачу.

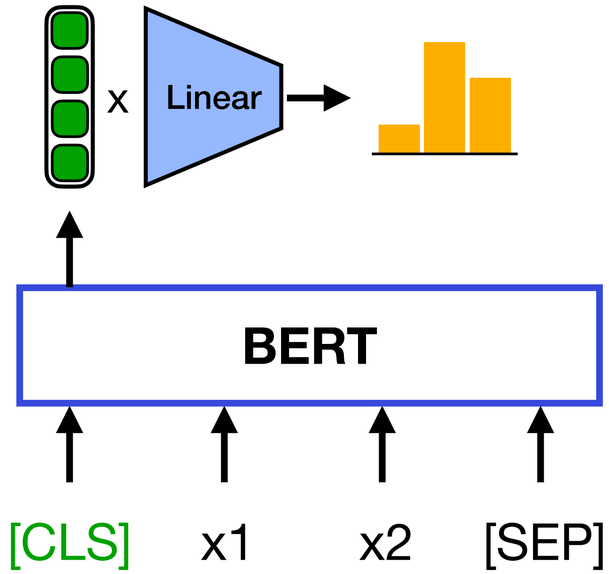
Обучение разбивается на две стадии:

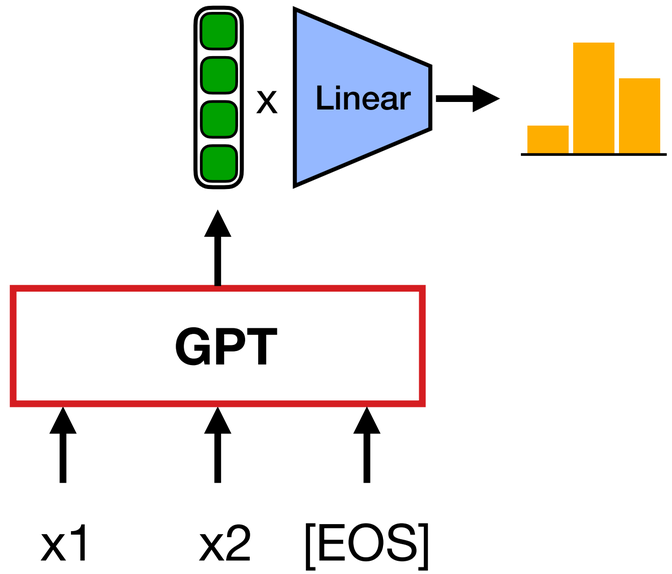
- **Предобучение** – self-supervised обучение на куче данных
- **Дообучение** на небольшую *downstream* задачу

# Модификации BERT и GPT

**Классификация текста:**

- У обеих моделей меняется последний линейный слой
- У BERT берется выход [CLS] токена
- У GPT – выход последнего токена

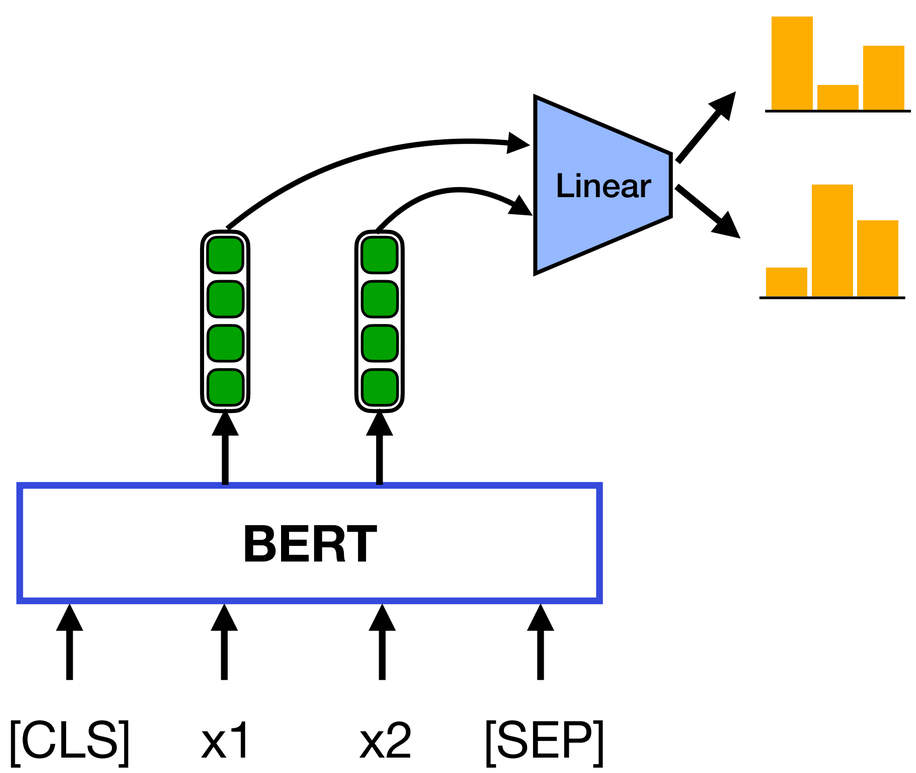


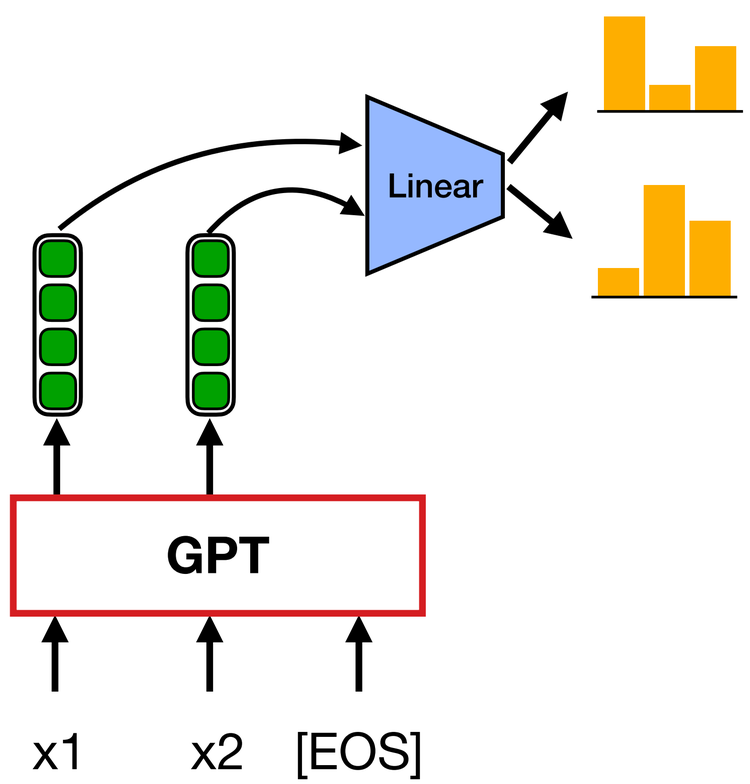



# Модификации BERT и GPT

**Классификация токенов:**

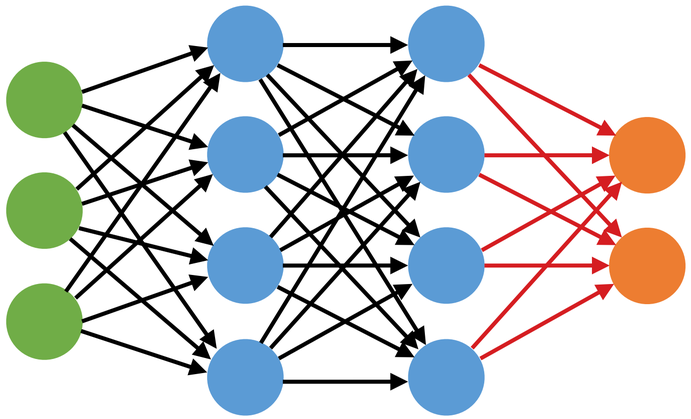
- Так же меняется последний линейный слой
- Берутся выходы каждого токена
- Лучше использовать **BERT**



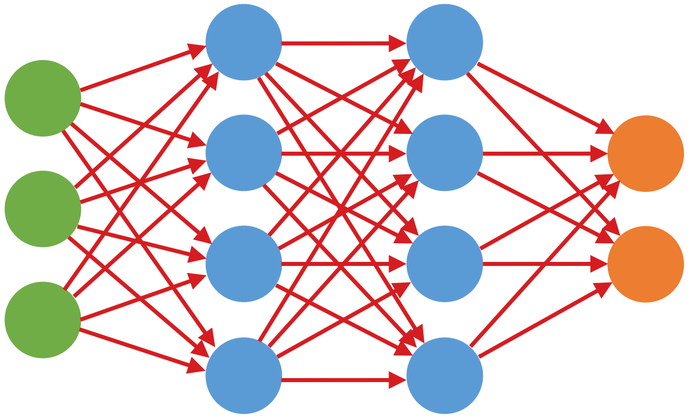
# Способы дообучения

Обучение головы
(линейный пробинг)



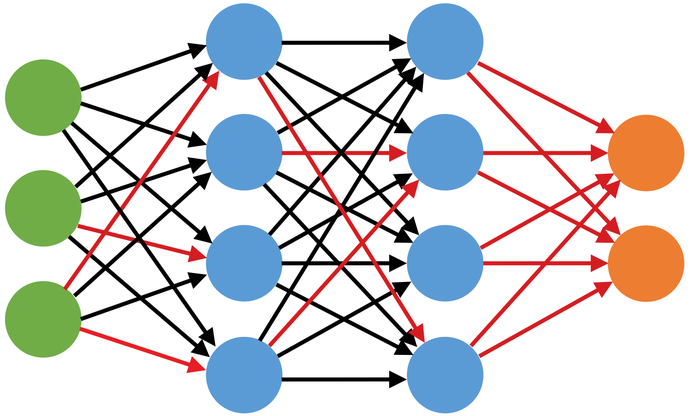
Обучается только последний слой

Fine-tuning



Обучается вся модель

Parameter Efficient
Fine-tuning



Обучается небольшой набор весов

# Пробинг vs Fine-tuning

**Downstream** задача – небольшая по объему данных задача, на которую дообучается модель (напр., классификация новостей).

# Пробинг vs Fine-tuning

**Downstream** задача – небольшая по объему данных задача, на которую дообучается модель (напр., классификация новостей).

**Пробинг**

- Самый простой и быстрый способ дообучения
- Не меняем информацию, извлекаемую моделью

**Fine-tuning**

- Самый долгий, но эффективный способ
- Обучает модель извлекать важную для **downstream** задачи информацию

# Сравнение с точки зрения ID и OOD

**ID** (In Distribution) – данные, лежащие в распределении **downstream** задачи (на чем обучаем)

- Новости издания Известия

**OOD** (Out Of Distribution) – данные, лежащие вне распределения **downstream** задачи (другие данные для задачи такого вида)

- Новости из Одноклассники.ru
- Новости Russia Today (RT)

# Сравнение с точки зрения ID и OOD

**ID** (In Distribution) – данные, лежащие в распределении **downstream** задачи (на чем обучаем)

- Новости издания Известия

**OOD** (Out Of Distribution) – данные, лежащие вне распределения **downstream** задачи (другие данные для задачи такого вида)

- Новости из Одноклассники.ru
- Новости Russia Today (RT)

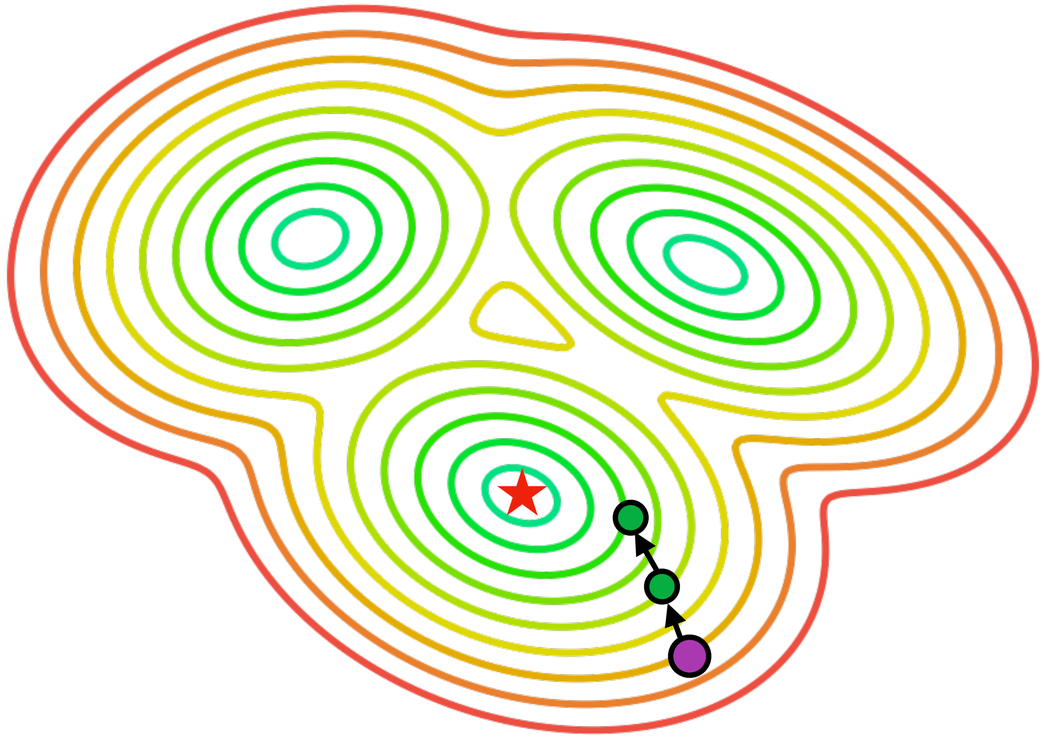
| | Fine-tuning | Пробинг |
|---|---|---|
| ID test | 85.1% | 82.9% |
| OOD test | 59.3% | 66.2% |

Точность классификации

# Fine-tuning портит модель

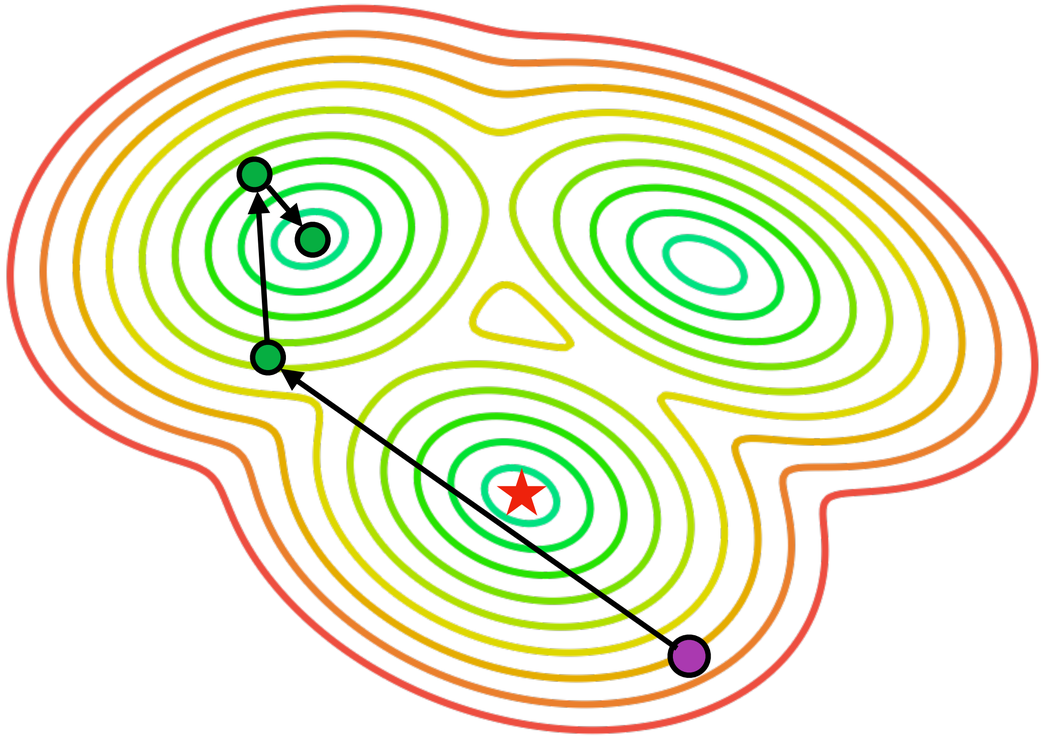
Из-за случайной инициализации головы градиенты слишком сильно сдвигают веса модели на первых итерациях. Это приводит к переобучению.

Пробинг



Недостаточно близко подходим к оптимуму

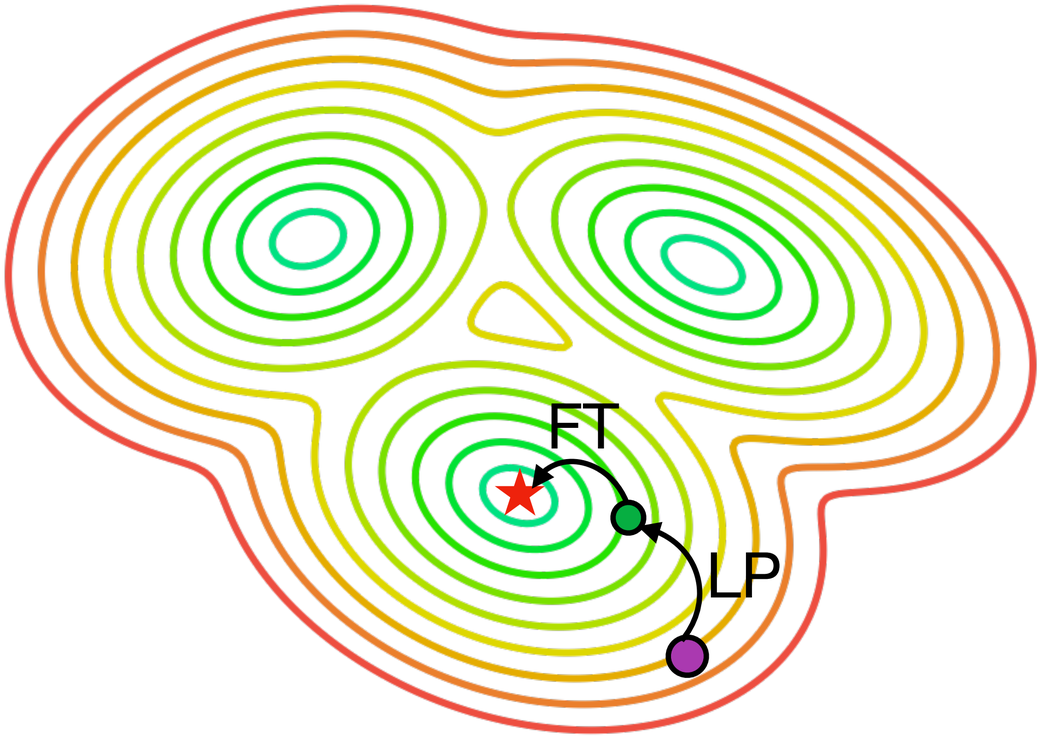
Fine-tuning



Близко подходим к оптимуму, но сильно меняем исходные параметры модели

# Пробинг → Fine-tuning

Обучим сначала голову, а затем дообучим веса всей модели с помощью fine-tuning.



# Пробинг → Fine-tuning

Качество вырастает как на OOD, так и на ID.

| | Fine-tuning | Пробинг | Пробинг → Fine-tuning |
|---|---|---|---|
| ID test | 85.1% | 82.9% | **85.7%** |
| OOD test | 59.3% | 66.2% | **68.9%** |

Точность классификации

https://arxiv.org/abs/2202.10054

# План

- Что такое Transfer Learning
- Способы дообучения моделей

- **Parameter-Efficient Fine-tuning**
  - Prompt-tuning
  - Adaptars
  - LoRa

# Parameter-Efficient Fine-tuning

- Способ дообучения моделей с изменением минимального числа параметров
- Используется для обучения LLM, которые слишком затратно обучать целиком
- При дообучении на несколько разных задач можно хранить только набор измененных параметров для каждой задачи

# Few-shot и Zero-shot для GPT

- GPT-3 обучалась на 45 тб данных. Это дает ей интересные свойства.

- Модель можно применять для новой задачи, показав ей примеры решения в промпте.

```
Переведи с русского на английский:
стол => table
сыр => cheese
дом =>
```

Это называется **Few-shot** режим.

# Few-shot и Zero-shot для GPT

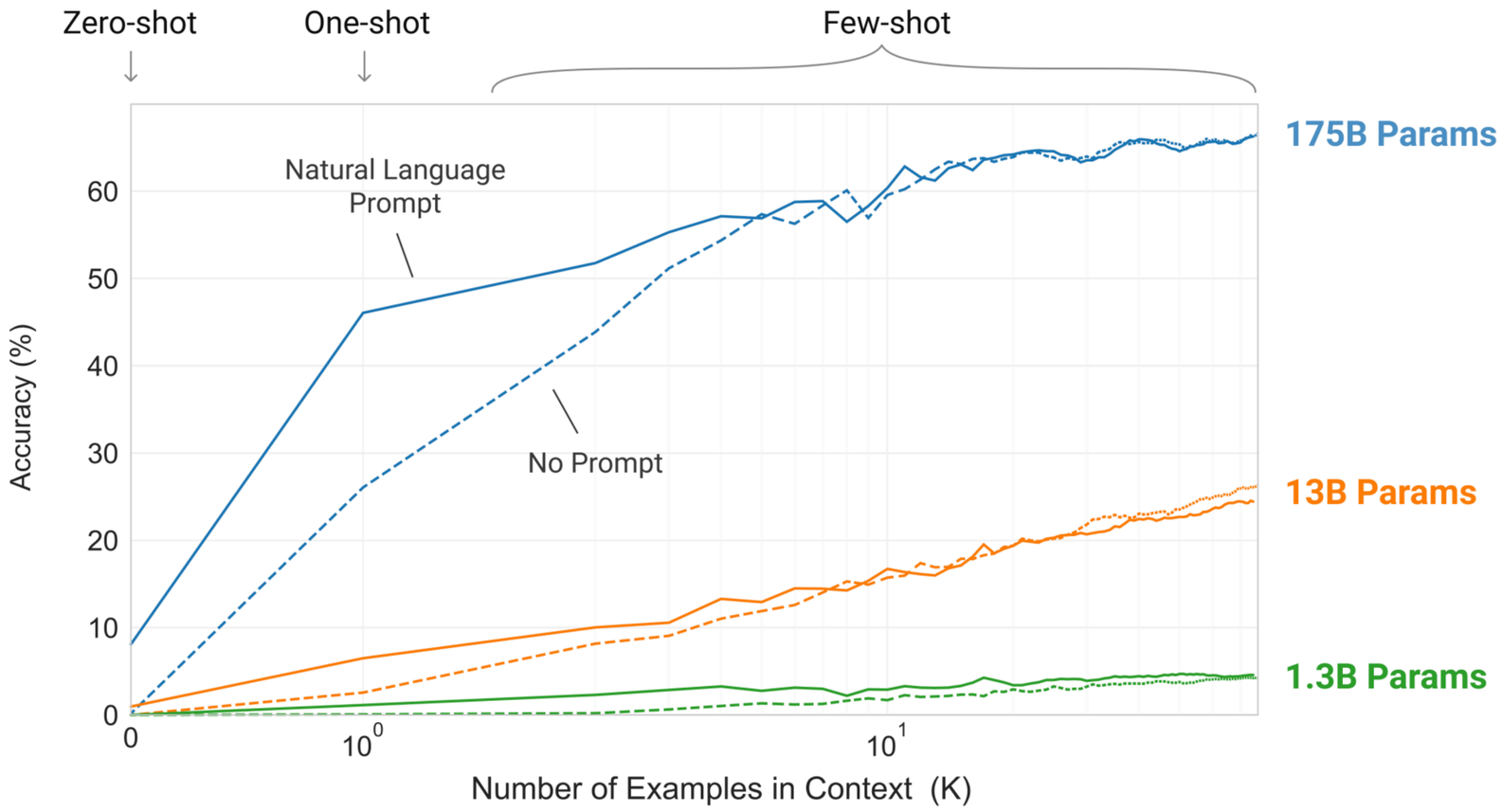
Можно вообще не показывать правильных решений

```
Переведи с русского на английский:
дом =>
```

Это называется **Zero-shot** режим.

# Few-shot и Zero-shot для GPT

Чем больше модель, тем лучше она работает в таком режиме.



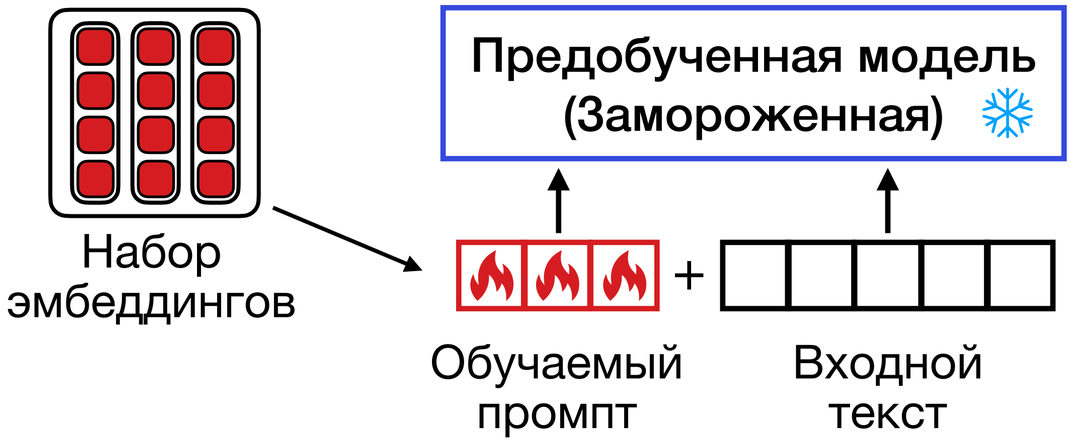

Однако результат может быть так себе из-за того, что

- Модель не училась решать эту задачу
- Промпт недостаточно хорош

# Prompt Tuning

**Идея:** Попробуем автоматически подобрать наиболее подходящий промпт

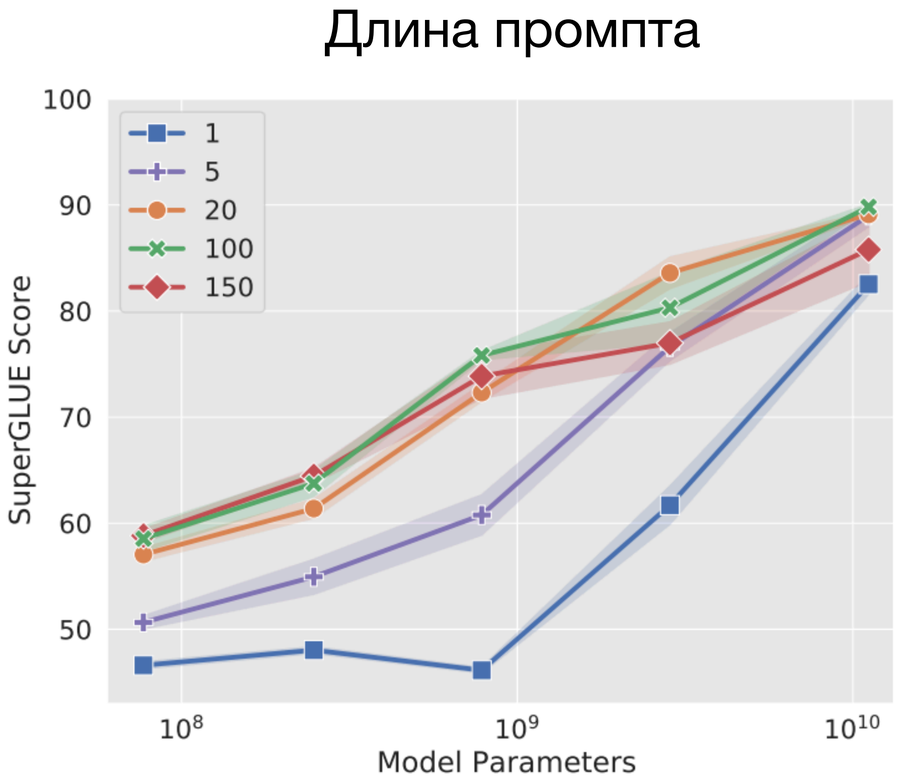
- Инициализируем эмбеддинги промпта случайно, задавая только их число
- Можно инициализировать эмбеддингами текстового промпта
- Для задачи классификации нужно дополнительно обучить голову

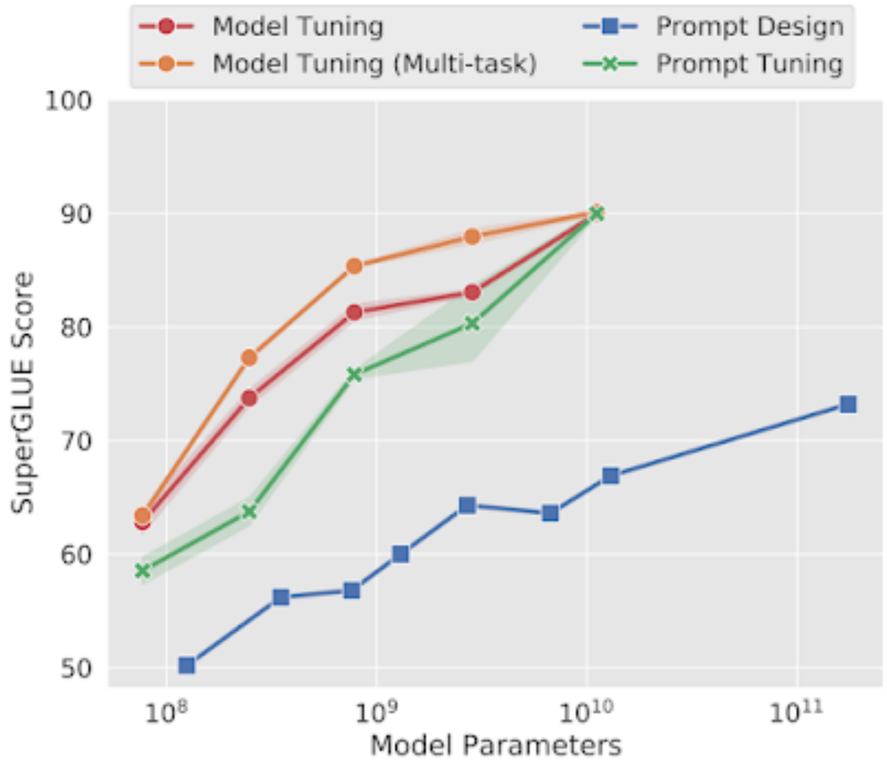



https://arxiv.org/abs/2104.08691

# Prompt Tuning

- Длина промпта напрямую влияет на качество модели
- Однако чем больше модель, тем меньше разница в качестве



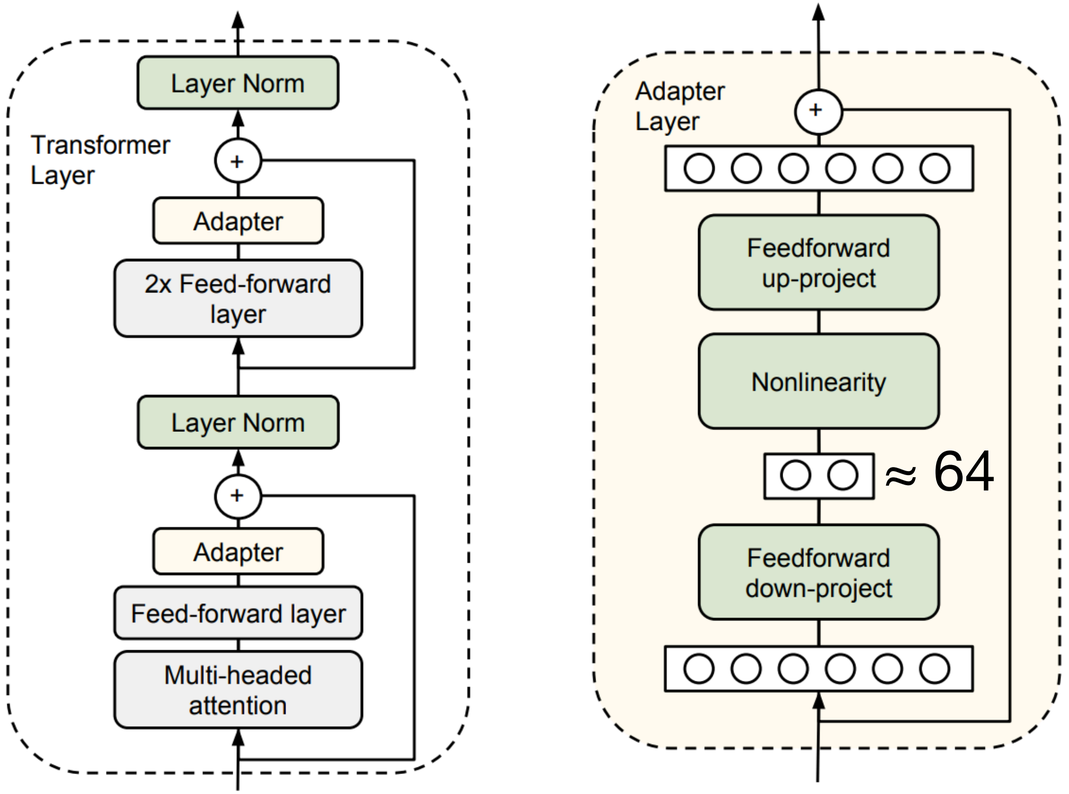



# Prompt Tuning: Недостатки

- Дополнение обучаемого промпта ограничивает максимальную длину и замедляет модель
- Prompt Tuning учится очень нестабильно и качество меняется немонотонно при увеличении размеров промпта и модели

# Adapters

- После каждого слоя внимания и FFN добавляется **небольшой** обучаемый адаптер
- Адаптер имеет skip-connection и два полносвязный слоя с **понижением** размерности (уменьшает число параметров)
- Обучаются только адаптеры, нормализации и голова
- Такое дообучение по качеству достигает fine-tuning




https://arxiv.org/abs/1902.00751

# Adapters: Недостатки

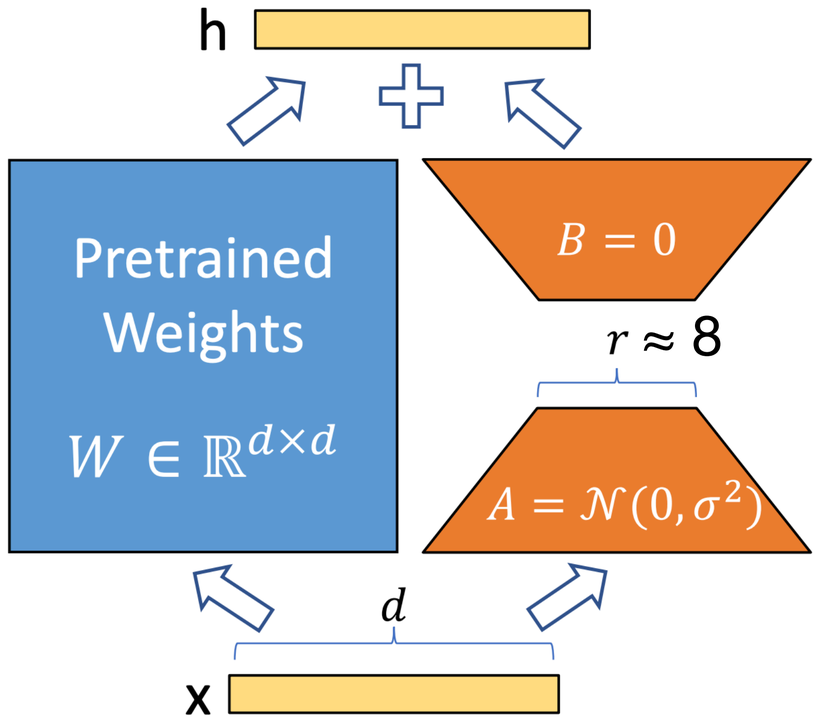
- Базовая версия адаптера добавляет довольно много параметров по сравнению с Prompt Tuning

- Адаптеры добавляют дополнительные слои, которые нельзя считать параллельно
- Это замедляет модель. Особенно для малых размеров батча.

# LoRA
Low-Rank Adaptation

- **Идея:** для адаптации модели к новой задаче нужно сдвинуть веса в сторону антиградиента

$$W' = W + \delta W$$




https://arxiv.org/abs/2106.09685

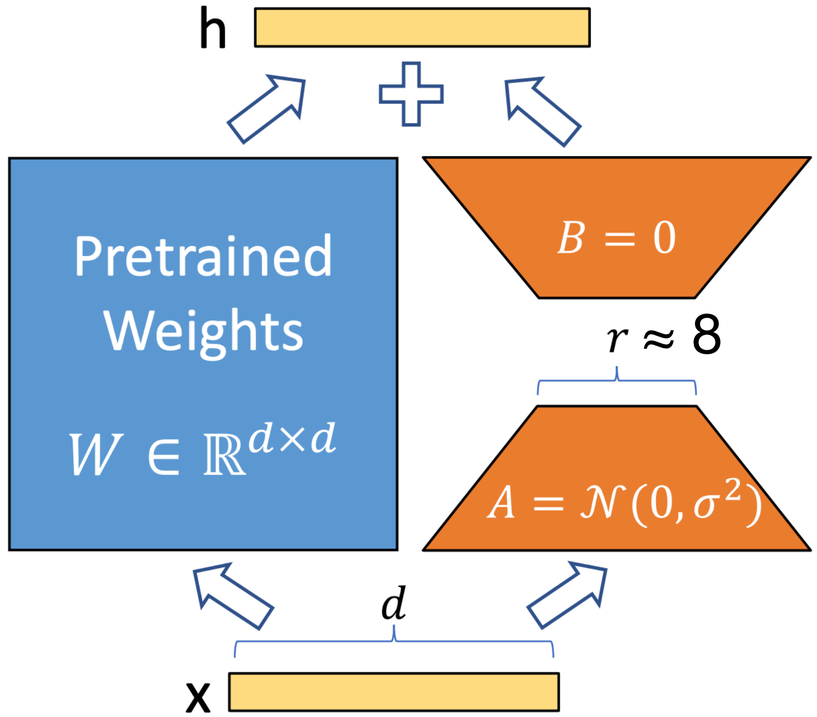
# LoRA
Low-Rank Adaptation

- **Идея:** для адаптации модели к новой задаче нужно сдвинуть веса в сторону антиградиента

$$W' = W + \delta W$$

- Приблизим $\delta W$ произведением обучаемых матриц $AB$

$$W' = W + AB$$




https://arxiv.org/abs/2106.09685

# LoRA
Low-Rank Adaptation

- **Идея:** для адаптации модели к новой задаче нужно сдвинуть веса в сторону антиградиента

$$W' = W + \delta W$$

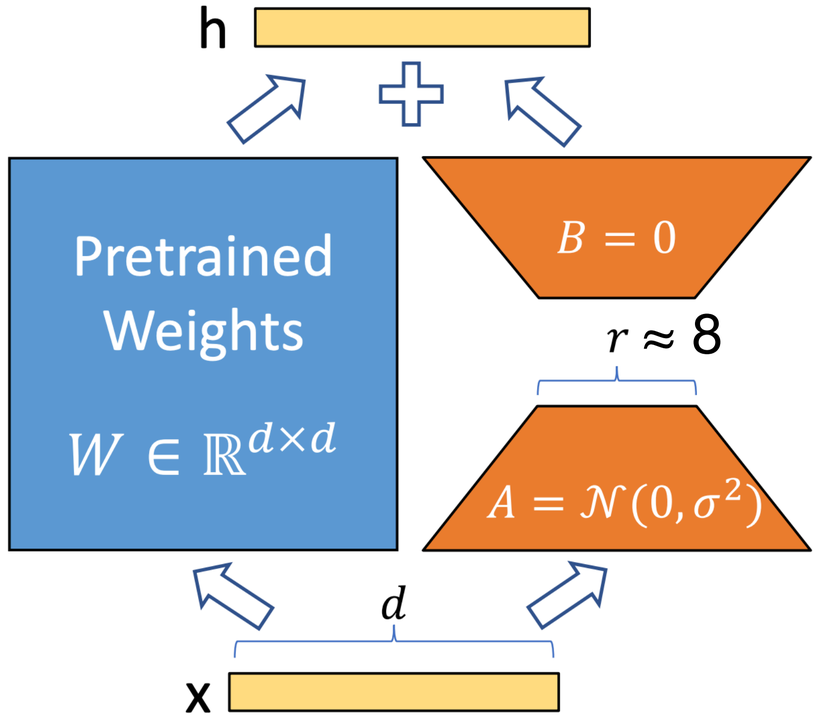
- Приблизим $\delta W$ произведением обучаемых матриц $AB$

$$W' = W + AB$$

- Так изменяются только матрицы $W_q$ и $W_v$ механизма внимания

- Так добавляется очень мало параметров и добавка может считаться параллельно с основным блоком

- Наиболее популярный способ PEFT




https://arxiv.org/abs/2106.09685

# BitFit

Bias-terms Fine-tuning

**Идея:** сдвиги имеют очень мало параметров, будем обучать только их

## Attention

$$\mathbf{Q}^{m,\ell}(\mathbf{x}) = \mathbf{W}_q^{m,\ell}\mathbf{x} + \mathbf{b}_q^{m,\ell}$$
$$\mathbf{K}^{m,\ell}(\mathbf{x}) = \mathbf{W}_k^{m,\ell}\mathbf{x} + \mathbf{b}_k^{m,\ell}$$
$$\mathbf{V}^{m,\ell}(\mathbf{x}) = \mathbf{W}_v^{m,\ell}\mathbf{x} + \mathbf{b}_v^{m,\ell}$$

$$\mathbf{h}_1^\ell = att\left(\mathbf{Q}^{1,\ell}, \mathbf{K}^{1,\ell}, \mathbf{V}^{1,\ell}, .., \mathbf{Q}^{m,\ell}, \mathbf{K}^{m,\ell}, \mathbf{V}^{m,l}\right)$$

## Feed Forward Network

$$\mathbf{h}_2^\ell = \text{Dropout}\left(\mathbf{W}_{m_1}^\ell \cdot \mathbf{h}_1^\ell + \mathbf{b}_{m_1}^\ell\right)$$
$$\mathbf{h}_3^\ell = \mathbf{g}_{LN_1}^\ell \odot \frac{(\mathbf{h}_2^\ell + \mathbf{x}) - \mu}{\sigma} + \mathbf{b}_{LN_1}^\ell$$
$$\mathbf{h}_4^\ell = \text{GELU}\left(\mathbf{W}_{m_2}^\ell \cdot \mathbf{h}_3^\ell + \mathbf{b}_{m_2}^\ell\right)$$
$$\mathbf{h}_5^\ell = \text{Dropout}\left(\mathbf{W}_{m_3}^\ell \cdot \mathbf{h}_4^\ell + \mathbf{b}_{m_3}^\ell\right)$$
$$\text{out}^\ell = \mathbf{g}_{LN_2}^\ell \odot \frac{(\mathbf{h}_5^\ell + \mathbf{h}_3^\ell) - \mu}{\sigma} + \mathbf{b}_{LN_2}^\ell$$

https://arxiv.org/abs/2106.10199

# Сравнение методов

Пусть у нас есть трансформер с 350M параметров и 24 слоями.

| **Метод** | **Число параметров** |
|---|---|
| Prompt tuning (длина: 20) | 15 тыс (0.006%) |
| Adapters (размерность 64) | 6M (2%) |
| LoRA (размерность 8) | 0.8M (0.22%) |
| BitFit | 0.32M (0.09%) |

Все три метода показывают сравнимое качество, однако LoRA наиболее стабильна и используется чаще остальных