

# Автоматическое дифференцирование

Семинар

ФКН ВШЭ

## Прямой режим

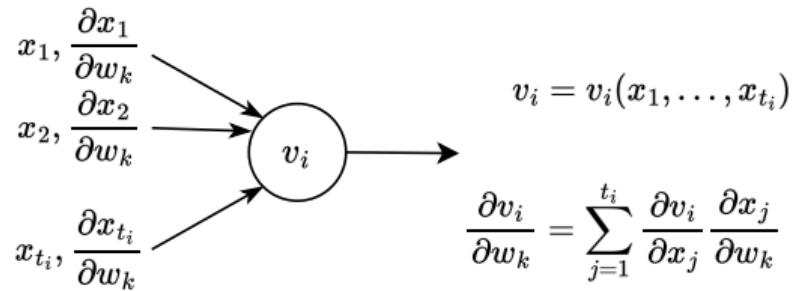


Figure 1: Иллюстрация прямого правила дифференцирования сложной функции для вычисления производной  $v_i$  по  $w_k$ .

- Использует прямое правило дифференцирования сложной функции

## Прямой режим

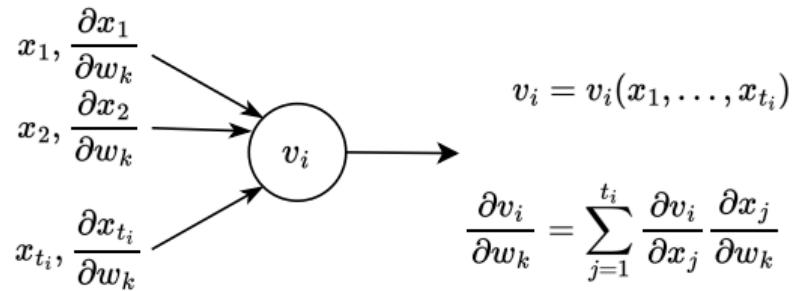


Figure 1: Иллюстрация прямого правила дифференцирования сложной функции для вычисления производной  $v_i$  по  $w_k$ .

- Использует прямое правило дифференцирования сложной функции
- Имеет сложность  $d \times \mathcal{O}(T)$  операций

## Обратный режим

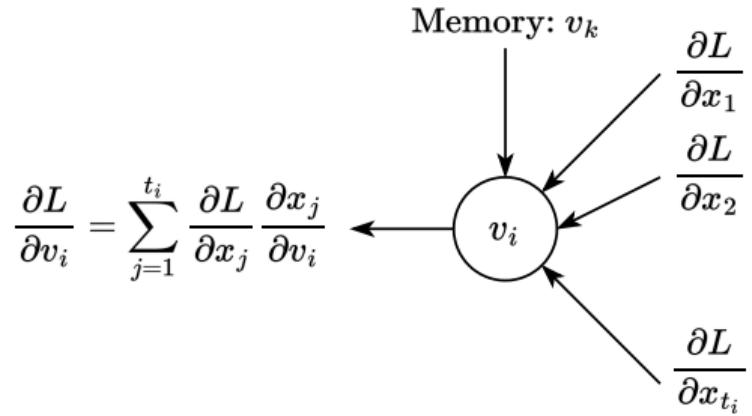


Figure 2: Иллюстрация обратного правила дифференцирования сложной функции для вычисления производной  $L$  по узлу  $v_i$ .

- Использует обратное правило дифференцирования сложной функции

## Обратный режим

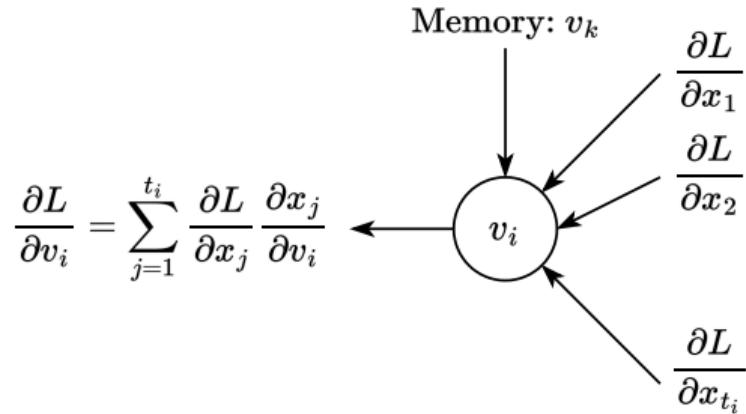


Figure 2: Иллюстрация обратного правила дифференцирования сложной функции для вычисления производной  $L$  по узлу  $v_i$ .

- Использует обратное правило дифференцирования сложной функции
- Сохраняет информацию из прямого прохода

## Обратный режим

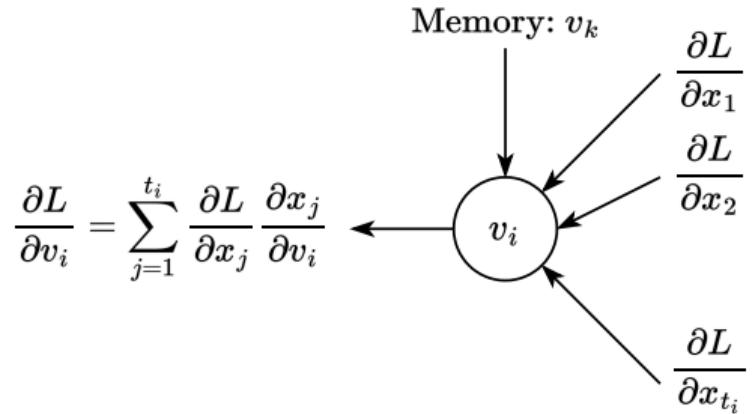


Figure 2: Иллюстрация обратного правила дифференцирования сложной функции для вычисления производной  $L$  по узлу  $v_i$ .

- Использует обратное правило дифференцирования сложной функции
- Сохраняет информацию из прямого прохода
- Имеет сложность  $\mathcal{O}(T)$  операций

## Простой пример

### Example

$$f(x_1, x_2) = x_1 \cdot x_2 + \sin x_1$$

Вычислим производные  $\frac{\partial f}{\partial x_i}$  с помощью прямого и обратного режимов.

## Простой пример

### Example

$$f(x_1, x_2) = x_1 \cdot x_2 + \sin x_1$$

Вычислим производные  $\frac{\partial f}{\partial x_i}$  с помощью прямого и обратного режимов.

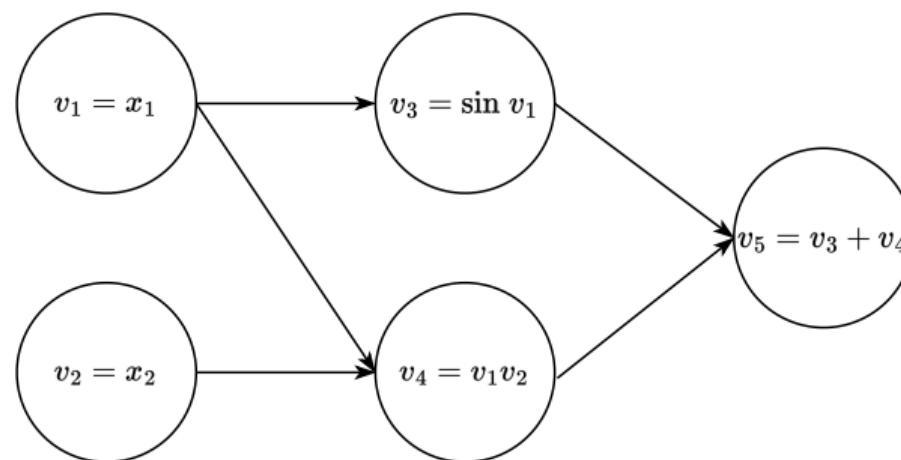


Figure 3: Иллюстрация вычислительного графа  $f(x_1, x_2)$ .

# Автоматическое дифференцирование с JAX

Пример №1

$$f(X) = \text{tr}(AX^{-1}B)$$

$$\nabla f = -X^{-T}A^TB^TX^{-T}$$

# Автоматическое дифференцирование с JAX

Пример №1

$$f(X) = \text{tr}(AX^{-1}B)$$

$$\nabla f = -X^{-T}A^TB^TX^{-T}$$

Пример №2

$$g(x) = \frac{1}{3}\|x\|_2^3$$

$$\nabla^2 g = \|x\|_2^{-1}xx^T + \|x\|_2 I_n$$

# Автоматическое дифференцирование с JAX

Пример №1

$$f(X) = \text{tr}(AX^{-1}B)$$

$$\nabla f = -X^{-T}A^TB^TX^{-T}$$

Пример №2

$$g(x) = \frac{1}{3}\|x\|_2^3$$

$$\nabla^2 g = \|x\|_2^{-1}xx^T + \|x\|_2 I_n$$

Вычислим градиенты и гессианы функций  $f$  и  $g$  

## Задача 1

### Question

Какой из режимов автоматического дифференцирования вы бы выбрали (прямой/обратный) для следующего вычислительного графа арифметических операций?

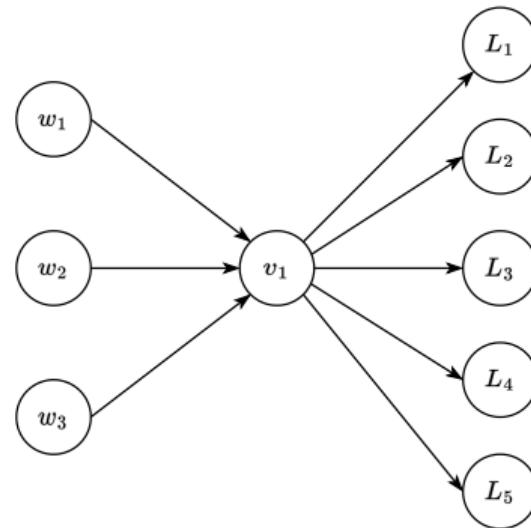


Figure 4: Какой режим вы бы выбрали для вычисления градиентов?

## Задача 2

Предположим, что у нас есть обратимая матрица  $A$  и вектор  $b$ , вектор  $x$  является решением линейной системы  $Ax = b$ , то есть можно записать аналитическое решение  $x = A^{-1}b$ .

### Question

Найдите производные  $\frac{\partial L}{\partial A}$ ,  $\frac{\partial L}{\partial b}$ .

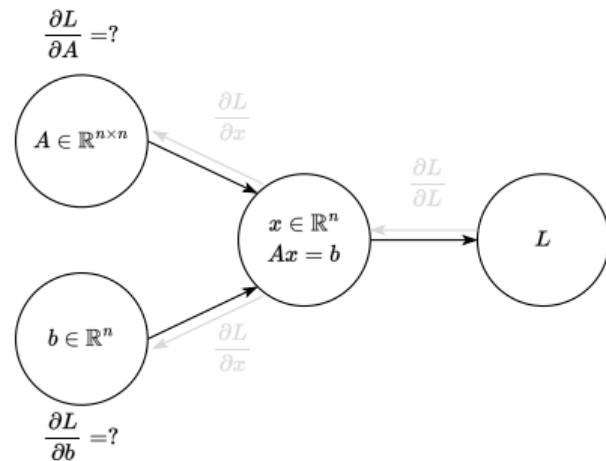
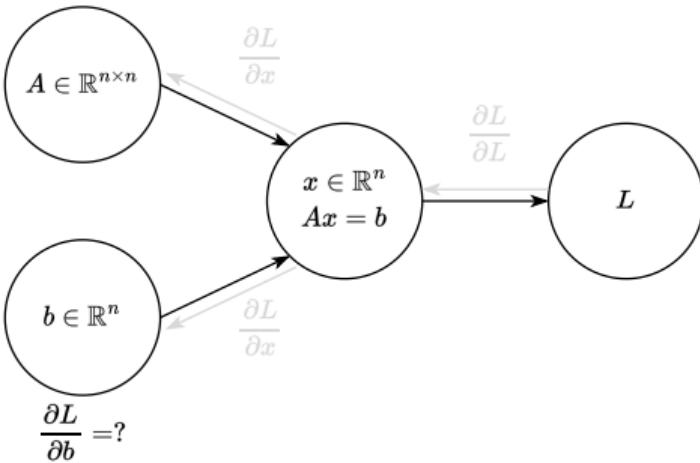


Figure 5:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$

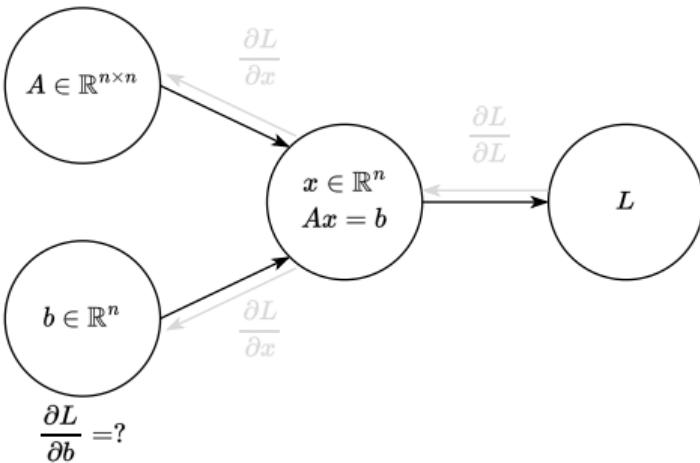


Предположим, что у нас есть обратимая матрица  $A$  и вектор  $b$ , вектор  $x$  является решением линейной системы  $Ax = b$ , то есть можно записать аналитическое решение  $x = A^{-1}b$ . В этом примере мы покажем, что вычисление всех производных  $\frac{\partial L}{\partial A}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial x}$ , то есть обратный проход, стоит примерно столько же, сколько и прямой проход.

Figure 6:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$



Предположим, что у нас есть обратимая матрица  $A$  и вектор  $b$ , вектор  $x$  является решением линейной системы  $Ax = b$ , то есть можно записать аналитическое решение  $x = A^{-1}b$ . В этом примере мы покажем, что вычисление всех производных  $\frac{\partial L}{\partial A}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial x}$ , то есть обратный проход, стоит примерно столько же, сколько и прямой проход.

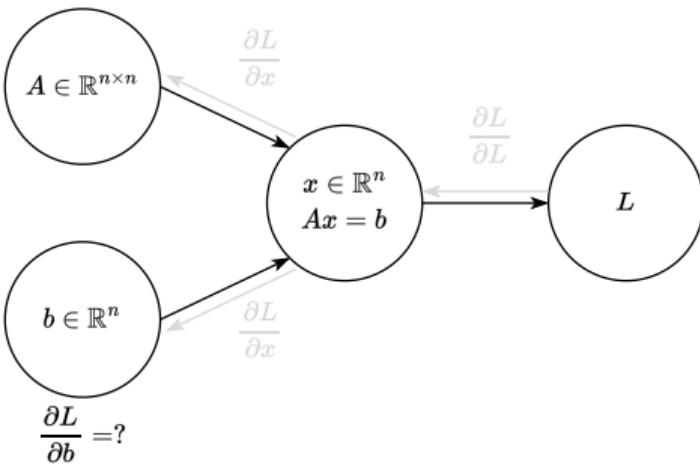
Известно, что дифференциал функции не зависит от параметризации:

$$dL = \left\langle \frac{\partial L}{\partial x}, dx \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Figure 6:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$



Предположим, что у нас есть обратимая матрица  $A$  и вектор  $b$ , вектор  $x$  является решением линейной системы  $Ax = b$ , то есть можно записать аналитическое решение  $x = A^{-1}b$ . В этом примере мы покажем, что вычисление всех производных  $\frac{\partial L}{\partial A}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial x}$ , то есть обратный проход, стоит примерно столько же, сколько и прямой проход.

Известно, что дифференциал функции не зависит от параметризации:

$$dL = \left\langle \frac{\partial L}{\partial x}, dx \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Учитывая линейную систему, имеем:

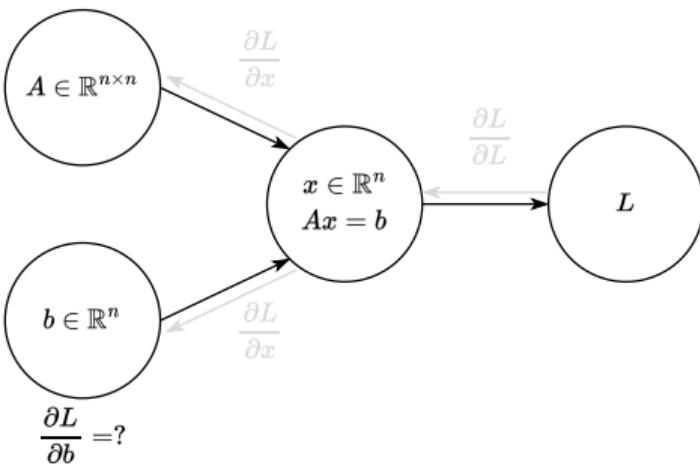
$$Ax = b$$

$$dAx + Adx = db \rightarrow dx = A^{-1}(db - dAx)$$

Figure 6:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$



Прямая подстановка даёт нам:

$$\left\langle \frac{\partial L}{\partial x}, A^{-1}(db - dAx) \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Figure 7:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$



Прямая подстановка даёт нам:

$$\left\langle \frac{\partial L}{\partial x}, A^{-1}(db - dAx) \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

$$\left\langle -A^{-T} \frac{\partial L}{\partial x} x^T, dA \right\rangle + \left\langle A^{-T} \frac{\partial L}{\partial x}, db \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Figure 7:  $x$  может быть найден как решение линейной системы

## Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$

$$A \in \mathbb{R}^{n \times n}$$

$$\frac{\partial L}{\partial x}$$

$$x \in \mathbb{R}^n$$
  
$$Ax = b$$

$$\frac{\partial L}{\partial L}$$

$$L$$

$$\frac{\partial L}{\partial x}$$

$$b \in \mathbb{R}^n$$

$$\frac{\partial L}{\partial b} = ?$$

Прямая подстановка даёт нам:

$$\left\langle \frac{\partial L}{\partial x}, A^{-1}(db - dAx) \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

$$\left\langle -A^{-T} \frac{\partial L}{\partial x} x^T, dA \right\rangle + \left\langle A^{-T} \frac{\partial L}{\partial x}, db \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Следовательно:

$$\frac{\partial L}{\partial A} = -A^{-T} \frac{\partial L}{\partial x} x^T \quad \frac{\partial L}{\partial b} = A^{-T} \frac{\partial L}{\partial x}$$

Figure 7:  $x$  может быть найден как решение линейной системы

# Распространение градиента через линейные наименьшие квадраты

$$\frac{\partial L}{\partial A} = ?$$



$$\frac{\partial L}{\partial b} = ?$$

Прямая подстановка даёт нам:

$$\left\langle \frac{\partial L}{\partial x}, A^{-1}(db - dAx) \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

$$\left\langle -A^{-T} \frac{\partial L}{\partial x} x^T, dA \right\rangle + \left\langle A^{-T} \frac{\partial L}{\partial b}, db \right\rangle = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle + \left\langle \frac{\partial L}{\partial b}, db \right\rangle$$

Следовательно:

$$\frac{\partial L}{\partial A} = -A^{-T} \frac{\partial L}{\partial x} x^T \quad \frac{\partial L}{\partial b} = A^{-T} \frac{\partial L}{\partial x}$$

Интересно, что наиболее вычислительно затратная часть здесь — это обращение матрицы, что совпадает со сложностью прямого прохода. Иногда даже можно сохранить сам результат, что делает обратный проход ещё дешевле.

Figure 7:  $x$  может быть найден как решение линейной системы

## Задача 3

Предположим, что у нас есть прямоугольная матрица  $W \in \mathbb{R}^{m \times n}$ , которая имеет сингулярное разложение:

$$W = U\Sigma V^T, \quad U^T U = I, \quad V^T V = I,$$
$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min(m,n)})$$

Регуляризатор  $R(W) = \text{tr}(\Sigma)$  в любой функции потерь поощряет решения с низким рангом.

### Question

Найдите производную  $\frac{\partial R}{\partial W}$ .

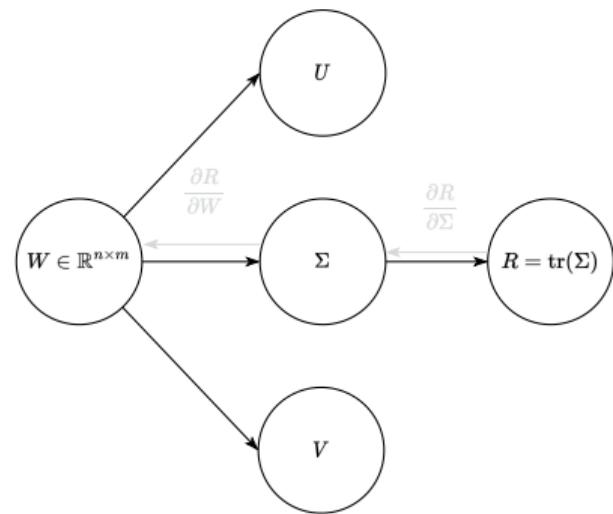
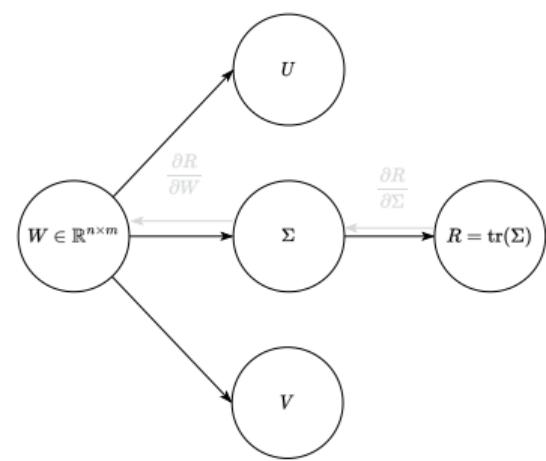


Figure 8: Вычислительный граф для сингулярного регуляризатора

# Распространение градиента через SVD



Предположим, что у нас есть прямоугольная матрица  $W \in \mathbb{R}^{m \times n}$ , которая имеет сингулярное разложение:

$$W = U\Sigma V^T, \quad U^T U = I, \quad V^T V = I, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min(m,n)})$$

1. Аналогично предыдущему примеру:

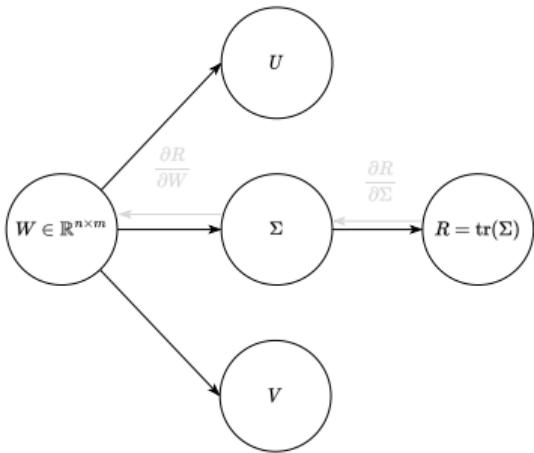
$$W = U\Sigma V^T$$

$$dW = dU\Sigma V^T + U d\Sigma V^T + U\Sigma dV^T$$

$$U^T dW V = U^T dU\Sigma V^T V + U^T U d\Sigma V^T V + U^T U\Sigma dV^T V$$

$$U^T dW V = U^T dU\Sigma + d\Sigma + \Sigma dV^T V$$

# Распространение градиента через SVD



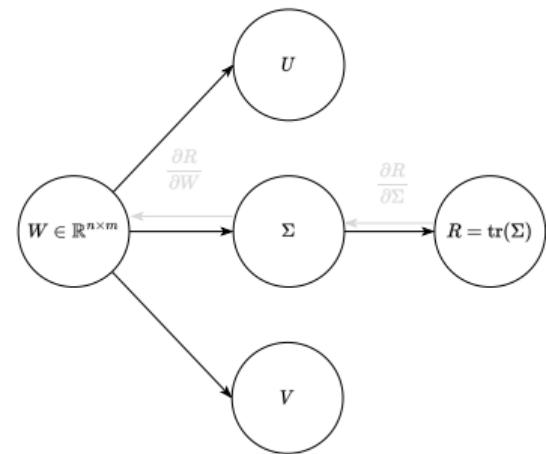
2. Заметим, что  $U^T U = I \rightarrow dU^T U + U^T dU = 0$ . Но также  $dU^T U = (U^T dU)^T$ , что означает, что матрица  $U^T dU$  антисимметрична:

$$(U^T dU)^T + U^T dU = 0 \rightarrow \text{diag}(U^T dU) = (0, \dots, 0)$$

Та же логика применима к матрице  $V$ :

$$\text{diag}(dV^T V) = (0, \dots, 0)$$

# Распространение градиента через SVD



2. Заметим, что  $U^T U = I \rightarrow dU^T U + U^T dU = 0$ . Но также  $dU^T U = (U^T dU)^T$ , что означает, что матрица  $U^T dU$  антисимметрична:

$$(U^T dU)^T + U^T dU = 0 \rightarrow \text{diag}(U^T dU) = (0, \dots, 0)$$

Та же логика применима к матрице  $V$ :

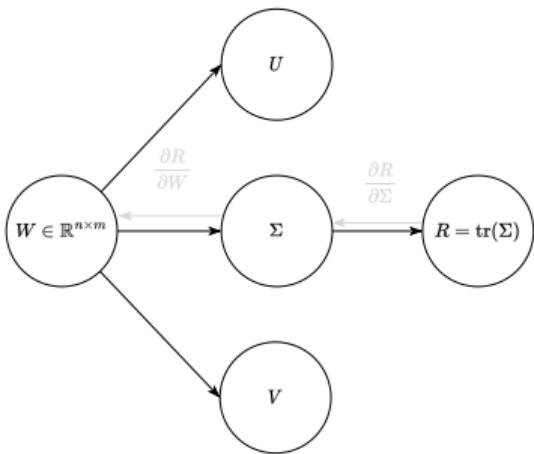
$$\text{diag}(dV^T V) = (0, \dots, 0)$$

3. При этом матрица  $d\Sigma$  диагональна, что означает (смотри пункт 1), что

$$\text{diag}(U^T dWV) = d\Sigma$$

Здесь с обеих сторон у нас диагональные матрицы.

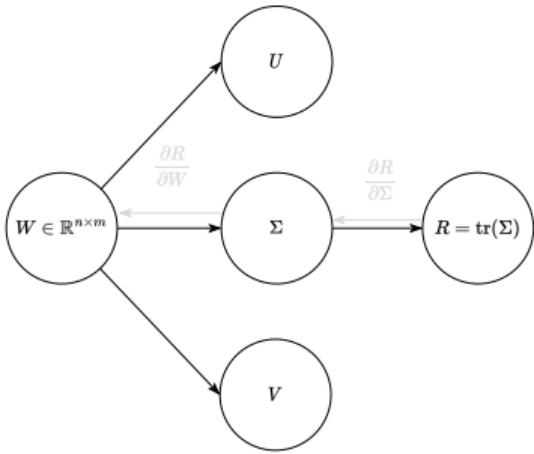
# Распространение градиента через SVD



4. Теперь мы можем разложить дифференциал функции потерь как функцию от  $\Sigma$  — такие задачи возникают в машинном обучении, когда нужно ограничить ранг матрицы:

$$\begin{aligned} dL &= \left\langle \frac{\partial L}{\partial \Sigma}, d\Sigma \right\rangle \\ &= \left\langle \frac{\partial L}{\partial \Sigma}, \text{diag}(U^T dWV) \right\rangle \\ &= \text{tr} \left( \frac{\partial L}{\partial \Sigma}^T \text{diag}(U^T dWV) \right) \end{aligned}$$

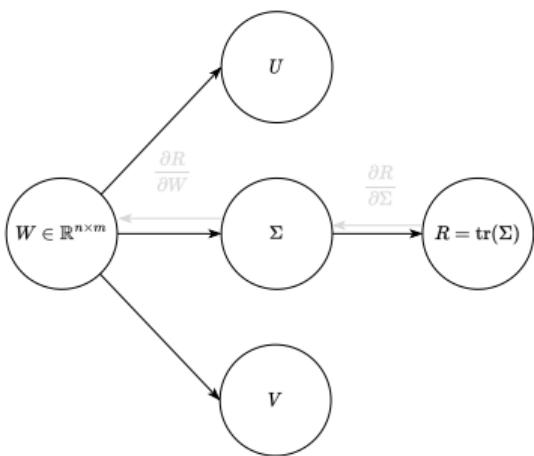
# Распространение градиента через SVD



5. Поскольку у нас диагональные матрицы внутри произведения, след диагональной части матрицы будет равен следу всей матрицы:

$$\begin{aligned} dL &= \text{tr} \left( \frac{\partial L^T}{\partial \Sigma} \text{diag}(U^T dWV) \right) \\ &= \text{tr} \left( \frac{\partial L^T}{\partial \Sigma} U^T dWV \right) \\ &= \left\langle \frac{\partial L}{\partial \Sigma}, U^T dWV \right\rangle \\ &= \left\langle U \frac{\partial L}{\partial \Sigma} V^T, dW \right\rangle \end{aligned}$$

# Распространение градиента через SVD



6. Наконец, используя другую параметризацию дифференциала

$$\left\langle U \frac{\partial L}{\partial \Sigma} V^T, dW \right\rangle = \left\langle \frac{\partial L}{\partial W}, dW \right\rangle$$

$$\frac{\partial L}{\partial W} = U \frac{\partial L}{\partial \Sigma} V^T,$$

Этот красивый результат позволяет связать градиенты  $\frac{\partial L}{\partial W}$  и  $\frac{\partial L}{\partial \Sigma}$ .

# Вычислительный эксперимент с JAX

Убедимся численно, что мы правильно вычислили производные в задачах 2 и 3 📸

## Архитектура прямого распространения

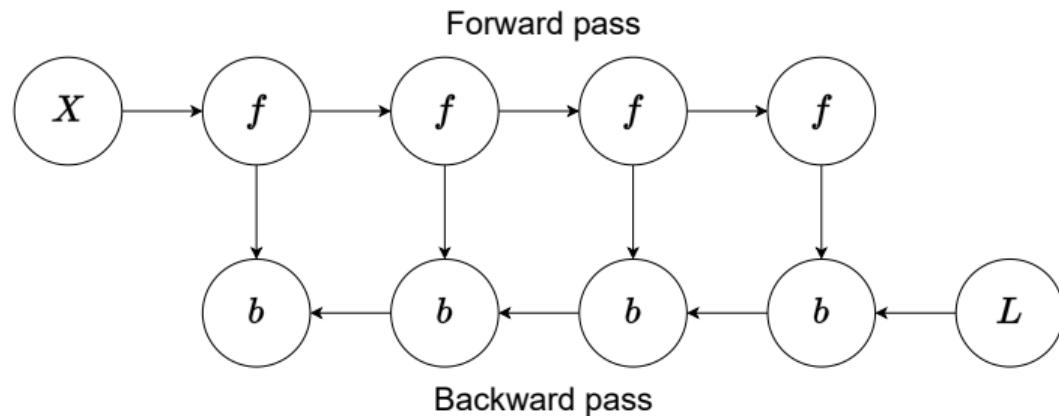


Figure 9: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Активации обозначены через  $f$ . Градиент функции потерь по активациям и параметрам обозначен через  $b$ .

## Архитектура прямого распространения

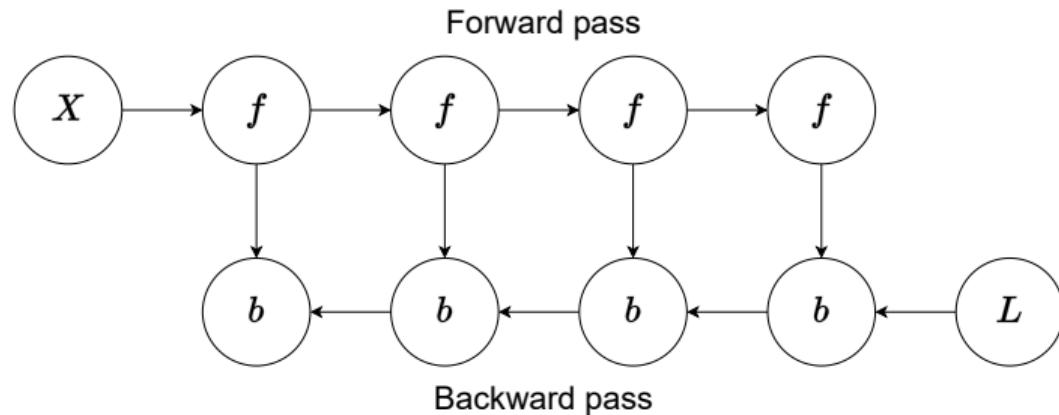


Figure 9: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Активации обозначены через  $f$ . Градиент функции потерь по активациям и параметрам обозначен через  $b$ .

! Important

Результаты, полученные для узлов  $f$ , необходимы для вычисления узлов  $b$ .

## Стандартное обратное распространение

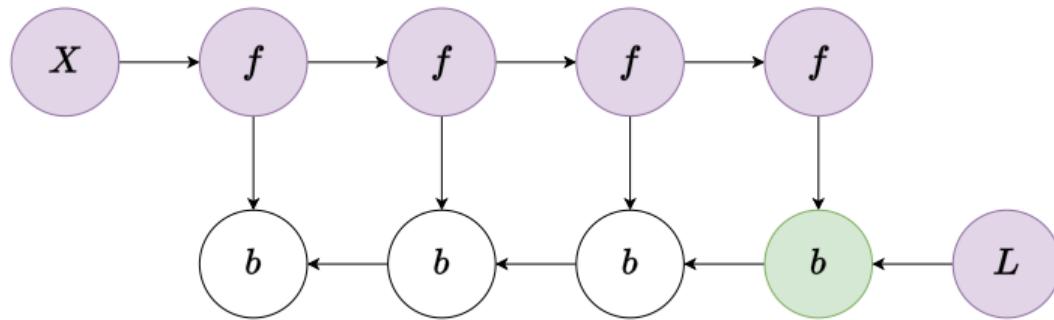


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

## Стандартное обратное распространение

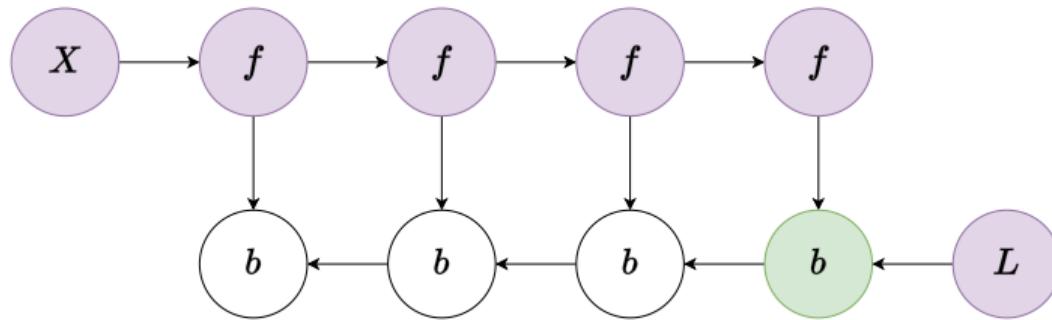


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Все активации  $f$  сохраняются в памяти после прямого прохода.

## Стандартное обратное распространение

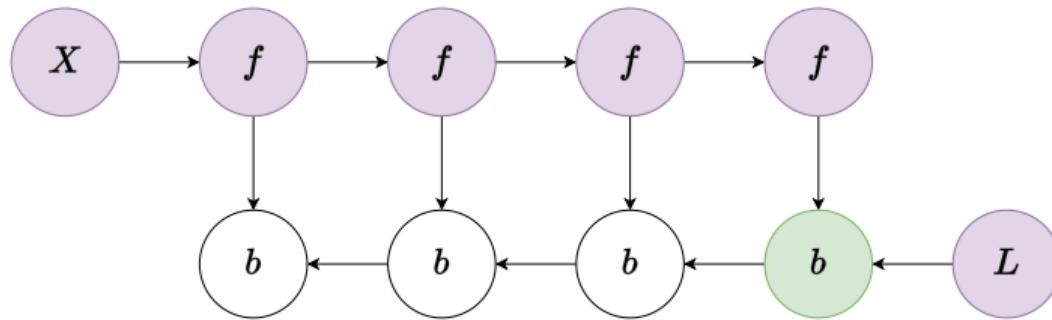


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Все активации  $f$  сохраняются в памяти после прямого прохода.

## Стандартное обратное распространение

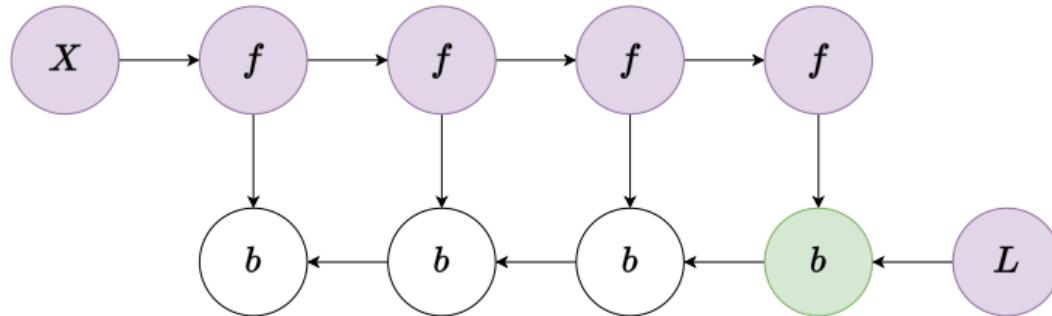


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Все активации  $f$  сохраняются в памяти после прямого прохода.
- Оптimalен по вычислениям: каждый узел вычисляется только один раз.

## Стандартное обратное распространение

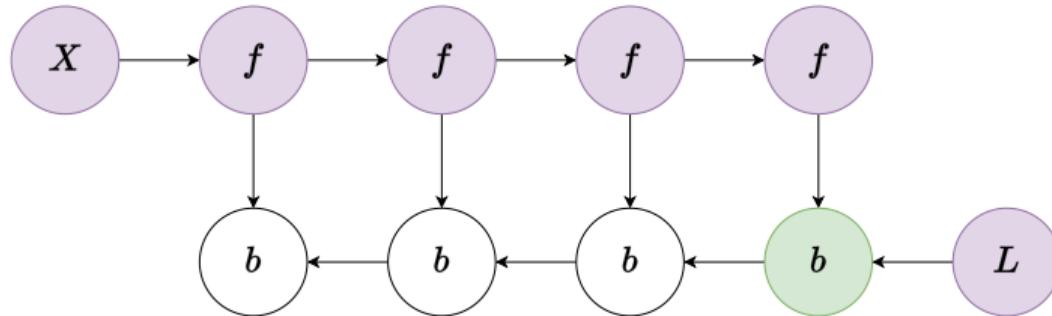


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Все активации  $f$  сохраняются в памяти после прямого прохода.
- Оптimalен по вычислениям: каждый узел вычисляется только один раз.

## Стандартное обратное распространение

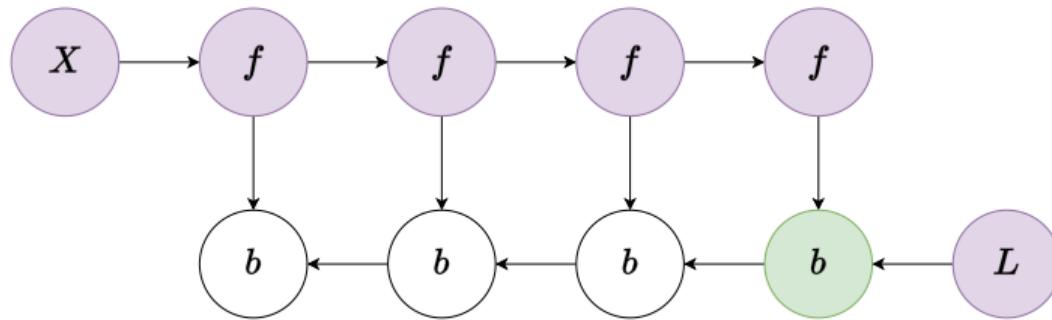


Figure 10: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Все активации  $f$  сохраняются в памяти после прямого прохода.
- Оптимален по вычислениям: каждый узел вычисляется только один раз.
- Высокое потребление памяти. Использование памяти растёт линейно с количеством слоёв в нейронной сети.

## Экономное по памяти обратное распространение

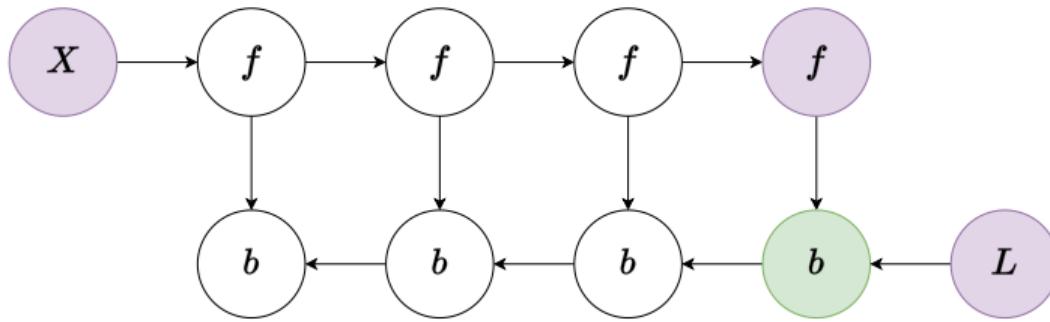


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями.  
Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

## Экономное по памяти обратное распространение

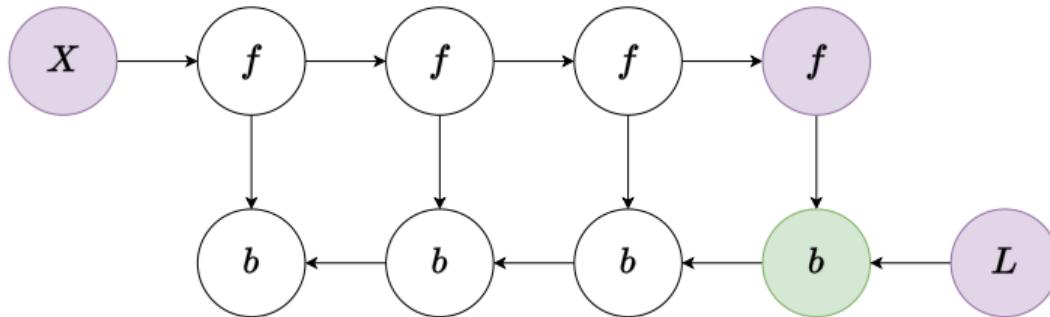


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями.  
Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Каждая активация  $f$  пересчитывается по мере необходимости.

## Экономное по памяти обратное распространение

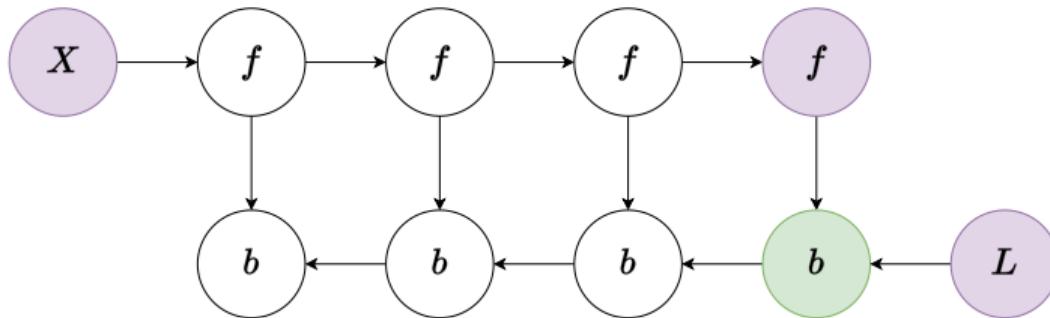


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями.  
Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Каждая активация  $f$  пересчитывается по мере необходимости.

## Экономное по памяти обратное распространение

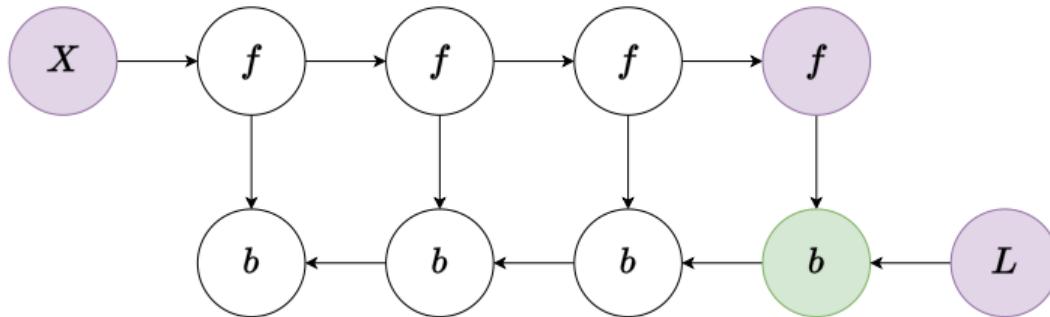


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Каждая активация  $f$  пересчитывается по мере необходимости.
- Оптимален по памяти: нет необходимости хранить все активации в памяти.

## Экономное по памяти обратное распространение

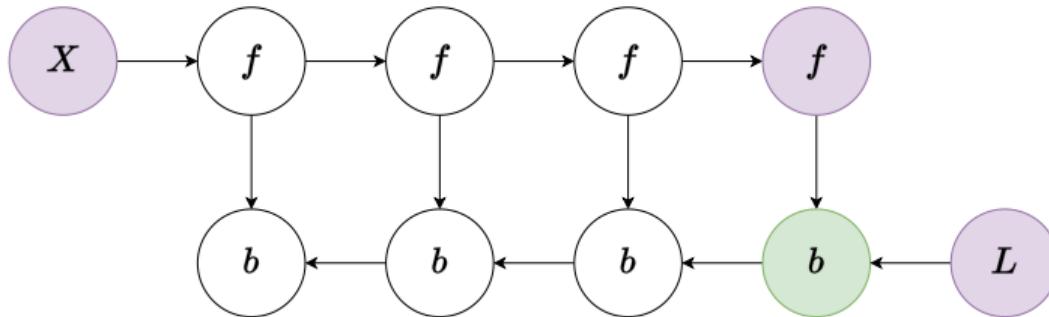


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Каждая активация  $f$  пересчитывается по мере необходимости.
- Оптимален по памяти: нет необходимости хранить все активации в памяти.

## Экономное по памяти обратное распространение

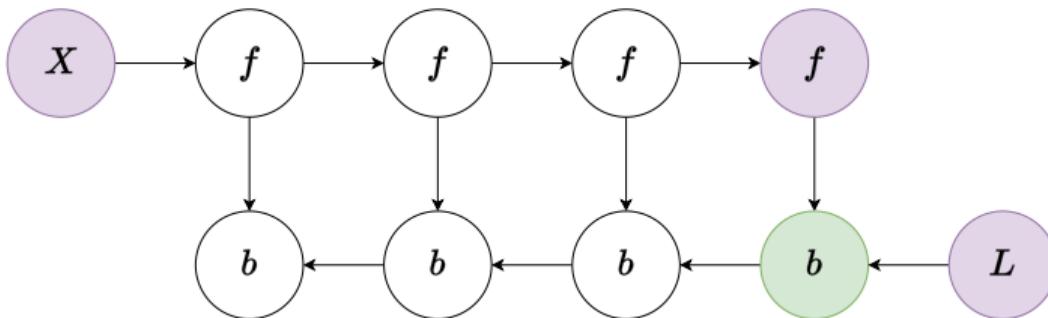


Figure 11: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Каждая активация  $f$  пересчитывается по мере необходимости.
- Оптимален по памяти: нет необходимости хранить все активации в памяти.
- Вычислительно неэффективен. Количество вычислений узлов растёт как  $n^2$ , тогда как в стандартном подходе — как  $n$ : каждый из  $n$  узлов пересчитывается порядка  $n$  раз.

## Обратное распространение с чекпоинтами

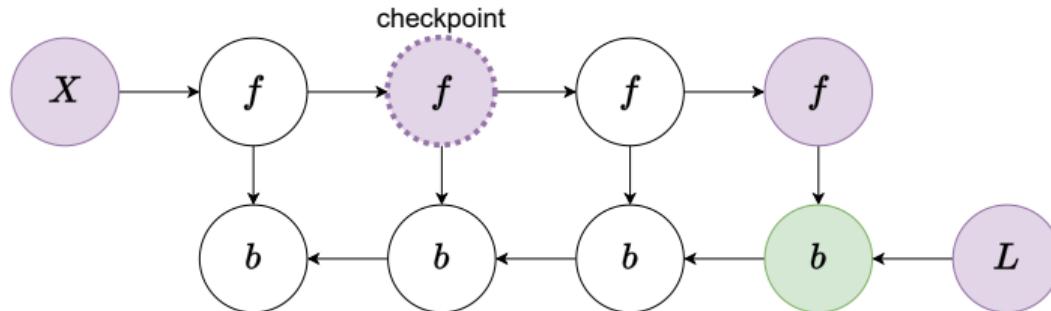


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

## Обратное распространение с чекпоинтами

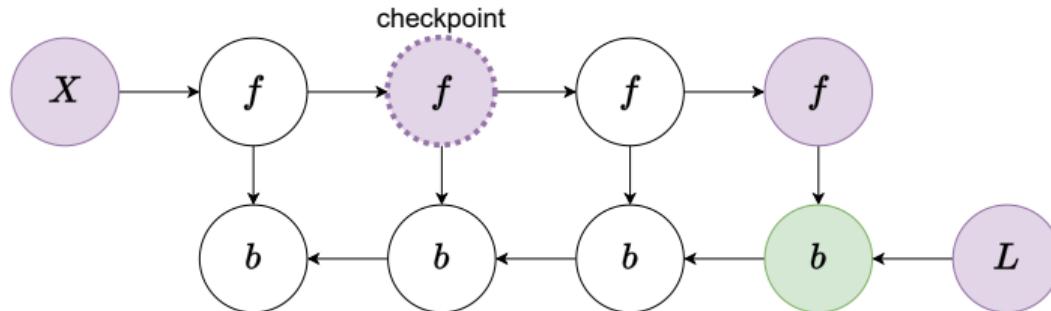


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Компромисс между **стандартным и экономным по памяти** подходами. Стратегия состоит в том, чтобы пометить подмножество активаций нейронной сети как чекпоинты, которые будут храниться в памяти.

## Обратное распространение с чекпоинтами

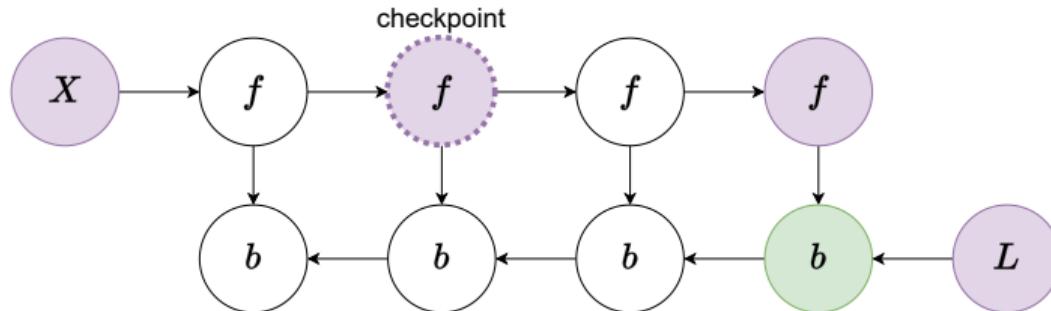


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Компромисс между **стандартным и экономным по памяти** подходами. Стратегия состоит в том, чтобы пометить подмножество активаций нейронной сети как чекпоинты, которые будут храниться в памяти.

## Обратное распространение с чекпоинтами

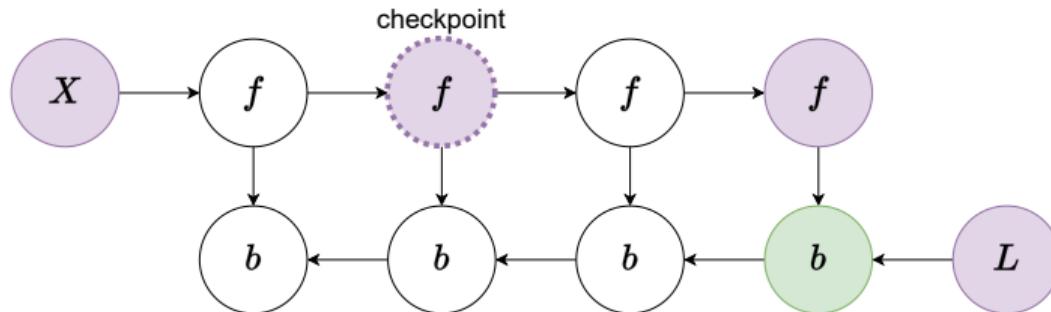


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Компромисс между **стандартным и экономным по памяти** подходами. Стратегия состоит в том, чтобы пометить подмножество активаций нейронной сети как чекпоинты, которые будут храниться в памяти.
- Более быстрый пересчёт активаций  $f$ . При вычислении узла  $b$  во время обратного прохода нужно пересчитать только узлы между этим  $b$  и последним предшествующим ему чекпоинтом.

## Обратное распространение с чекпоинтами

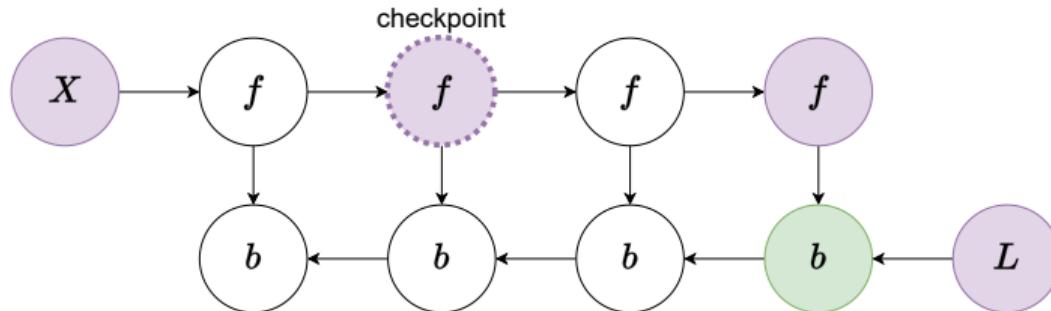


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Компромисс между **стандартным и экономным по памяти** подходами. Стратегия состоит в том, чтобы пометить подмножество активаций нейронной сети как чекпоинты, которые будут храниться в памяти.
- Более быстрый пересчёт активаций  $f$ . При вычислении узла  $b$  во время обратного прохода нужно пересчитать только узлы между этим  $b$  и последним предшествующим ему чекпоинтом.

## Обратное распространение с чекпоинтами

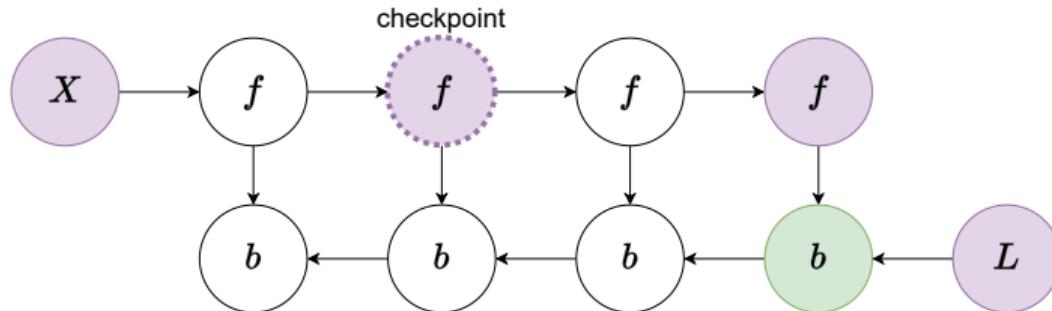


Figure 12: Вычислительный граф для получения градиентов простой нейронной сети прямого распространения с  $n$  слоями. Фиолетовым цветом обозначены узлы, хранящиеся в памяти.

- Компромисс между **стандартным и экономным по памяти** подходами. Стратегия состоит в том, чтобы пометить подмножество активаций нейронной сети как чекпоинты, которые будут храниться в памяти.
  - Более быстрый пересчёт активаций  $f$ . При вычислении узла  $b$  во время обратного прохода нужно пересчитать только узлы между этим  $b$  и последним предшествующим ему чекпоинтом.
  - Потребление памяти зависит от количества чекпоинтов. Эффективнее, чем **стандартный** подход.

## Визуализация чекпоинтинга градиента

Анимированная визуализация вышеописанных подходов 

Пример использования чекпоинтинга градиента 

## Оценка следа методом Хатчинсона<sup>1</sup>

Этот пример иллюстрирует оценку следа гессиана нейронной сети с использованием метода Хатчинсона — алгоритма, который позволяет получить такую оценку из произведений матрицы на вектор:

Пусть  $X \in \mathbb{R}^{d \times d}$  и  $v \in \mathbb{R}^d$  — случайный вектор такой, что  $\mathbb{E}[vv^T] = I$ . Тогда,

$$\text{tr}(X) = \mathbb{E}[v^T X v] = \frac{1}{V} \sum_{i=1}^V v_i^T X v_i.$$

Пример использования оценки следа методом Хатчинсона 

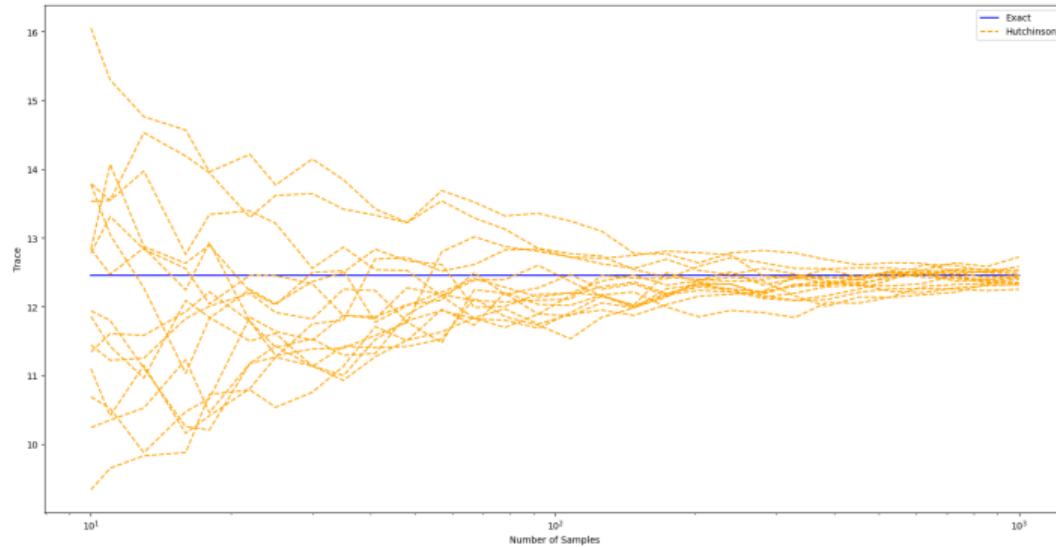


Figure 13: Несколько запусков оценки следа методом Хатчинсона с разными начальными значениями генератора случайных чисел.

<sup>1</sup>A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines - M.F. Hutchinson, 1990