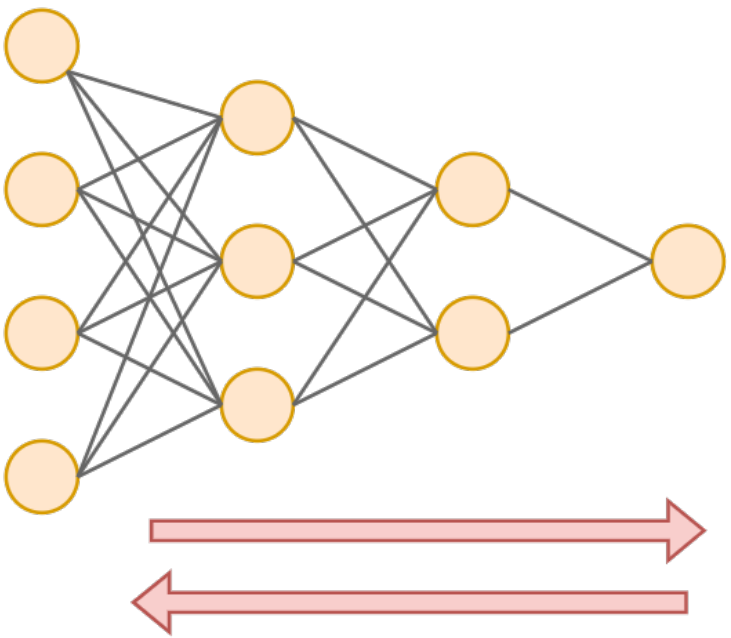
Автоматическое дифференцирование

Виктор Китов
victorkitov.github.io



Настройка весов сети

Итерация градиентной оптимизации:

$$w := w - \varepsilon \nabla \mathcal{L}(w)$$

$$\nabla \mathcal{L}(w) = \left[\frac{\partial \mathcal{L}(w)}{\partial w_1}; \frac{\partial \mathcal{L}(w)}{\partial w_2}; ... \frac{\partial \mathcal{L}(w)}{\partial w_K} \right]$$

- $\mathcal{L}(w)$ - по минибатчу объектов.
- $w = [w_1, w_2, ...w_K]$ обновляются все синхронно.
- Как эффективно вычислить $\nabla \mathcal{L}(w)$?

Содержание

Численная аппроксимация

- Вычисление $\nabla \mathcal{L}(w)$ через разностную аппроксимацию ($\delta_i = [0, ...0, \delta, 0, ...0]$)

$$\frac{\partial \mathcal{L}}{\partial w_i} \approx \frac{\mathcal{L}(w + \delta_i) - \mathcal{L}(w)}{\delta}$$

Численная аппроксимация

- Вычисление $\nabla \mathcal{L}(w)$ через разностную аппроксимацию ($\delta_i = [0, ...0, \delta, 0, ...0]$)

$$\frac{\partial \mathcal{L}}{\partial w_i} \approx \frac{\mathcal{L}(w + \delta_i) - \mathcal{L}(w)}{\delta}$$

$\oplus$: автоматический метод, нет риска ошибки

$\ominus$: приближённая. а не точная оценка

$\ominus$: имеет вычислительную сложность $O(K^2)$

- нужно посчитать K производных
- сложность вычисления каждой: $O(K)$

Вычисление напрямую

- Вычисление напрямую
  - $\oplus$: точный градиент
  - $\ominus$: громоздкие вычисления, риск ошибки

Вычисление напрямую

- Вычисление напрямую
  - $\oplus$: точный градиент
  - $\ominus$: громоздкие вычисления, риск ошибки
- Библиотеки символьного дифференцирования
  - $\oplus$: точный градиент
  - $\oplus$: автоматический метод, нет риска ошибки

Вычисление напрямую

- Вычисление напрямую
  $\oplus$: точный градиент
  $\ominus$: громоздкие вычисления, риск ошибки
- Библиотеки символьного дифференцирования
  $\oplus$: точный градиент
  $\oplus$: автоматический метод, нет риска ошибки

- Оба метода вычислительно неэффективны - повторное вычисление одинаковых слагаемых:

$$[A(w)B(w)C(w)]'$$
$$= A'(w)B(w)C(w) + A(w)B'(w)C(w) + A(w)B(w)C'(w)$$

Автоматическое дифференцирование

- Библиотеки автоматического дифференцирования вычисляют значение градиента в точке
  $\oplus$: автоматически, без риска ошибки
  $\oplus$: точный градиент
  $\oplus$: эффективно за $O(K)$
- Используют метод обратного распространения ошибки (backpropagation).
- Основные библиотеки: PyTorch, Tensorflow, JAX.

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B} \frac{\partial B}{\partial w}$$

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B}\frac{\partial B}{\partial w}$$

$$\mathcal{L}(w) = A(B_1(w), B_2(w), B_3(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B}\frac{\partial B}{\partial w}$$

$$\mathcal{L}(w) = A(B_1(w), B_2(w), B_3(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A}{\partial B_1}\frac{\partial B_1}{\partial w} + \frac{\partial A}{\partial B_2}\frac{\partial B_2}{\partial w} + \frac{\partial A}{\partial B_3}\frac{\partial B_3}{\partial w}$$

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B}\frac{\partial B}{\partial w}$$

$$\mathcal{L}(w) = A(B_1(w), B_2(w), B_3(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A}{\partial B_1}\frac{\partial B_1}{\partial w} + \frac{\partial A}{\partial B_2}\frac{\partial B_2}{\partial w} + \frac{\partial A}{\partial B_3}\frac{\partial B_3}{\partial w}$$

$$\mathcal{L}(w) = A(B(C(w))), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

Дифференцирование сложной функции

$$\mathcal{L}(w) = A(B(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B}\frac{\partial B}{\partial w}$$

$$\mathcal{L}(w) = A(B_1(w), B_2(w), B_3(w)), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A}{\partial B_1}\frac{\partial B_1}{\partial w} + \frac{\partial A}{\partial B_2}\frac{\partial B_2}{\partial w} + \frac{\partial A}{\partial B_3}\frac{\partial B_3}{\partial w}$$

$$\mathcal{L}(w) = A(B(C(w))), \quad \frac{\partial \mathcal{L}(w)}{\partial w} - ?$$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A}{\partial B}\frac{\partial B}{\partial C}\frac{\partial C}{\partial w}$$

Содержание

Обратное распространение ошибки

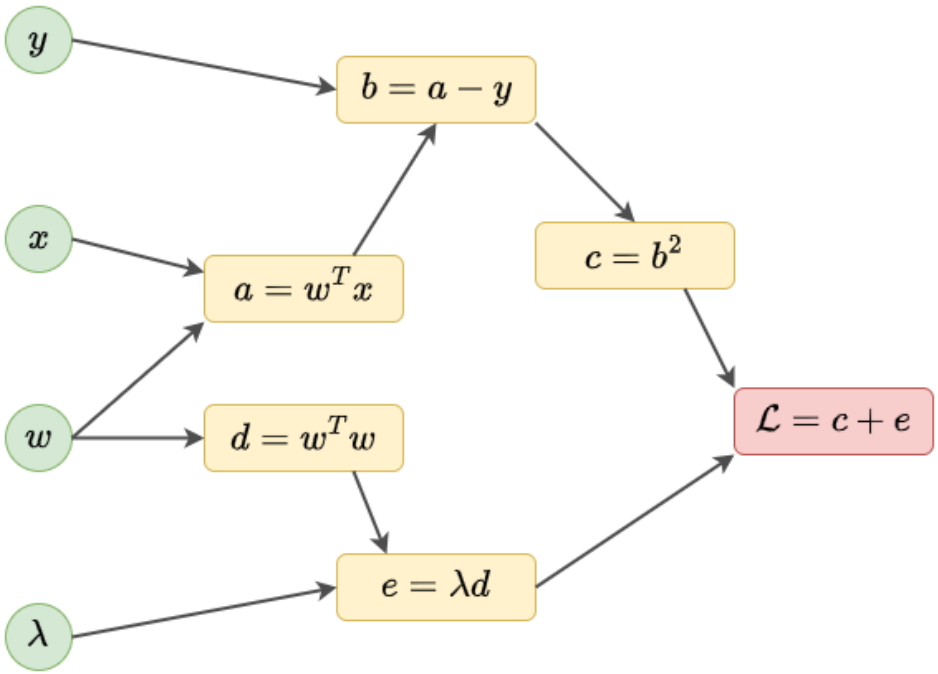
- Вычисление функции потерь $\mathcal{L}(w)$ декомпозируется в вычислительный граф из элементарных преобразований
  - по которым мы можем посчитать производную.
  - +, -, *, : , exp, log, sin, cos, ...

Обратное распространение ошибки

- Вычисление функции потерь $\mathcal{L}(w)$ декомпозируется в вычислительный граф из элементарных преобразований
  - по которым мы можем посчитать производную.
  - +, -, *, : , exp, log, sin, cos, ...
- Пример: $\hat{y} = \mathbf{w}^T\mathbf{x}, \quad \mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$

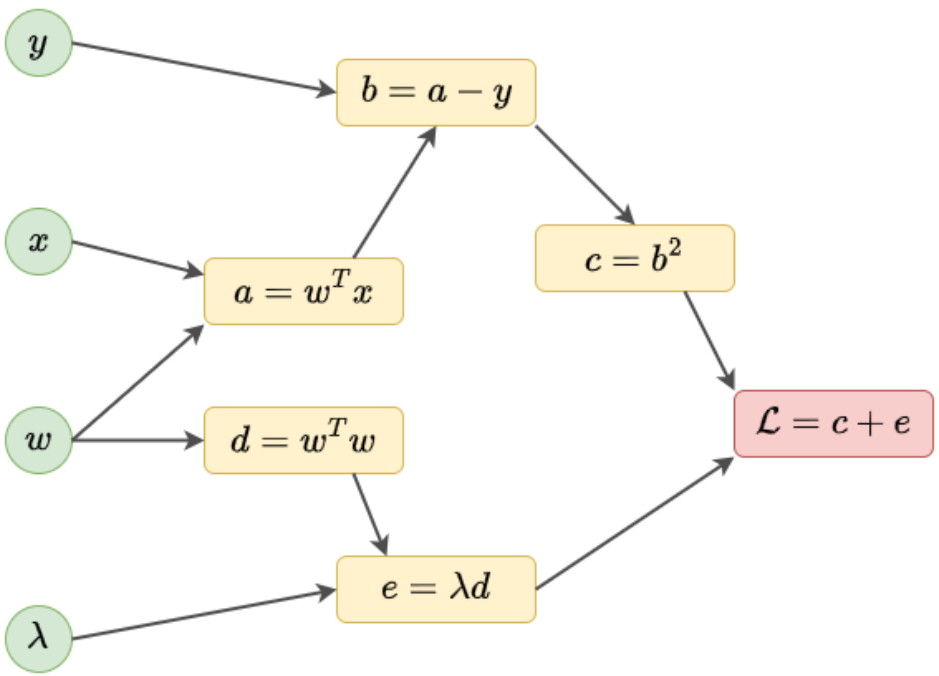
Обратное распространение ошибки

- Вычисление функции потерь $\mathcal{L}(w)$ декомпозируется в вычислительный граф из элементарных преобразований
  - по которым мы можем посчитать производную.
  - +, -, *, : , exp, log, sin, cos, ...
- Пример: $\hat{y} = \mathbf{w}^T\mathbf{x}, \quad \mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



Проходы вперёд и назад

- Вычисление $\nabla \mathcal{L}(w)$:
  - проход вперёд (forward pass)
  - проход назад (backward pass)

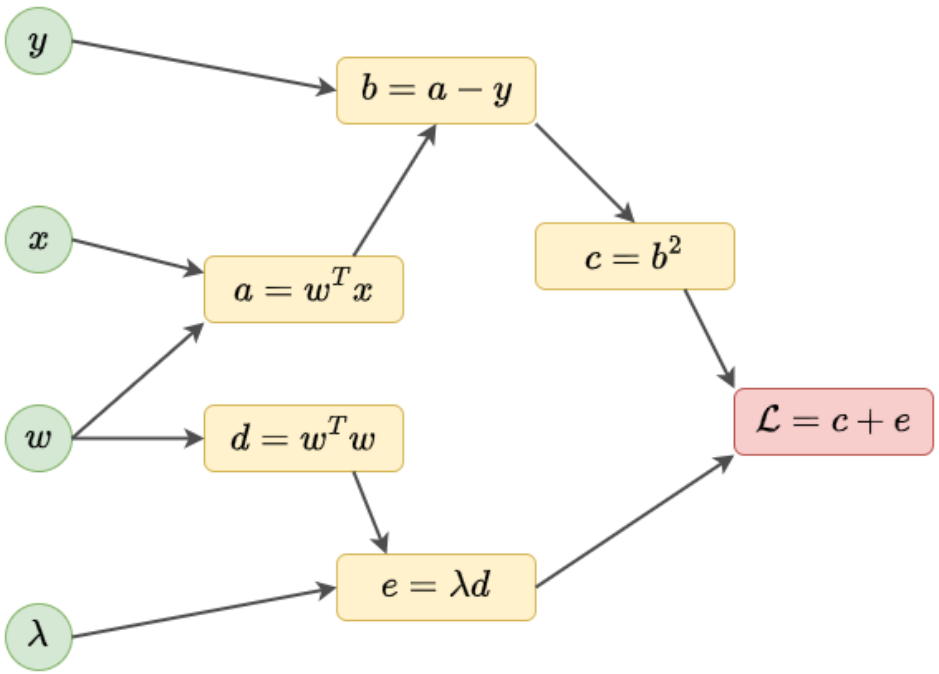


Проход вперёд

Проход вперёд: итеративно **слева-направо**

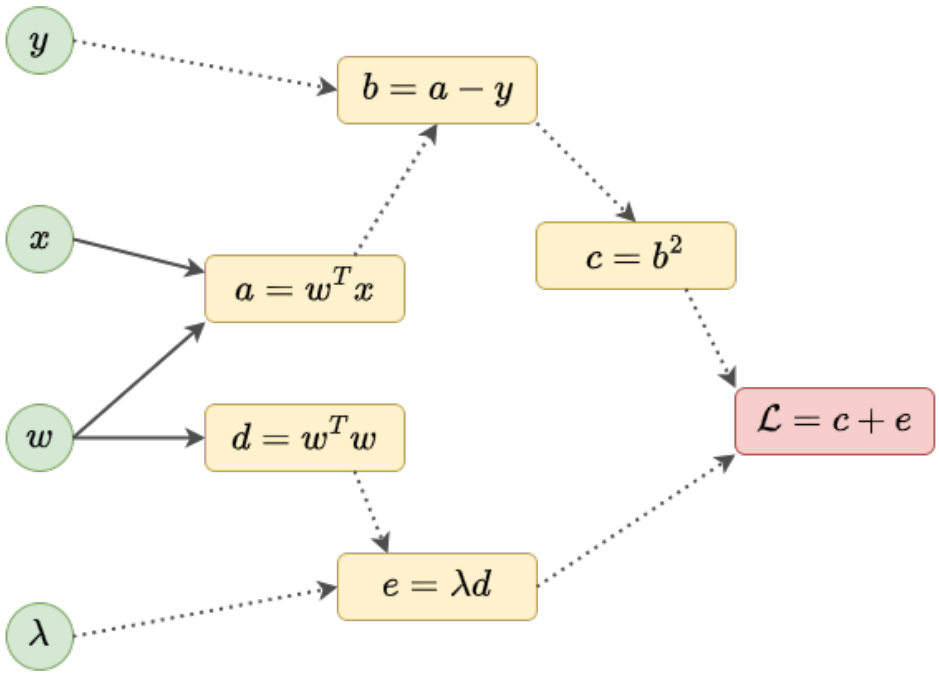
- вычисляются все промежуточные переменные
- на каждом шаге запоминаются:
  - промежуточные переменные
  - функциональные преобразования для их получения
    - реально важны только их производные

Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

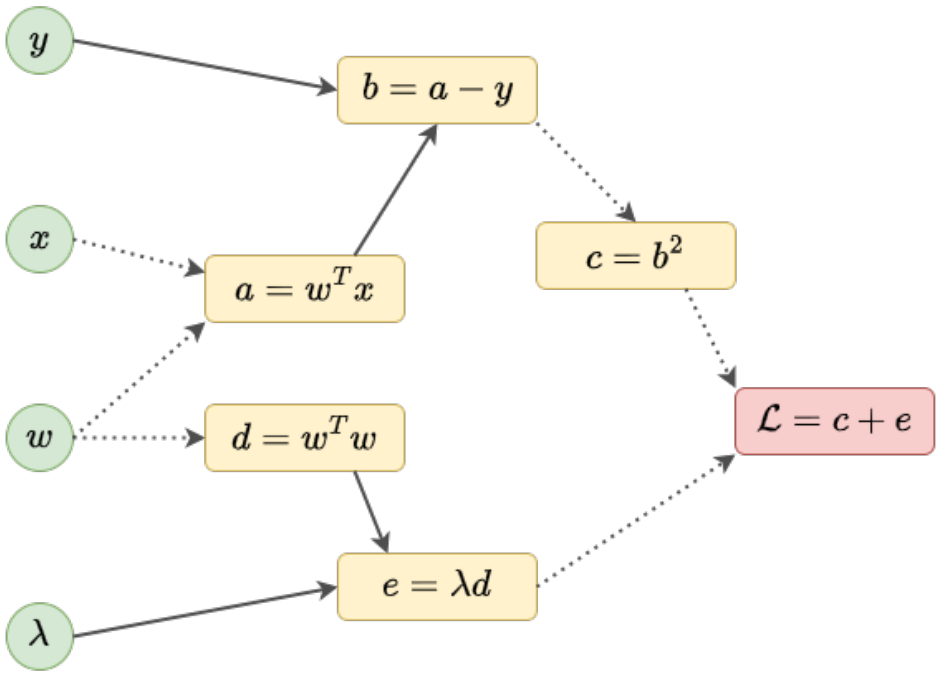
Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$a = 1 \cdot 3 + 2 \cdot 4 = 3 + 8 = 11; \quad d = 3 \cdot 3 + 4 \cdot 4 = 9 + 16 = 25$$

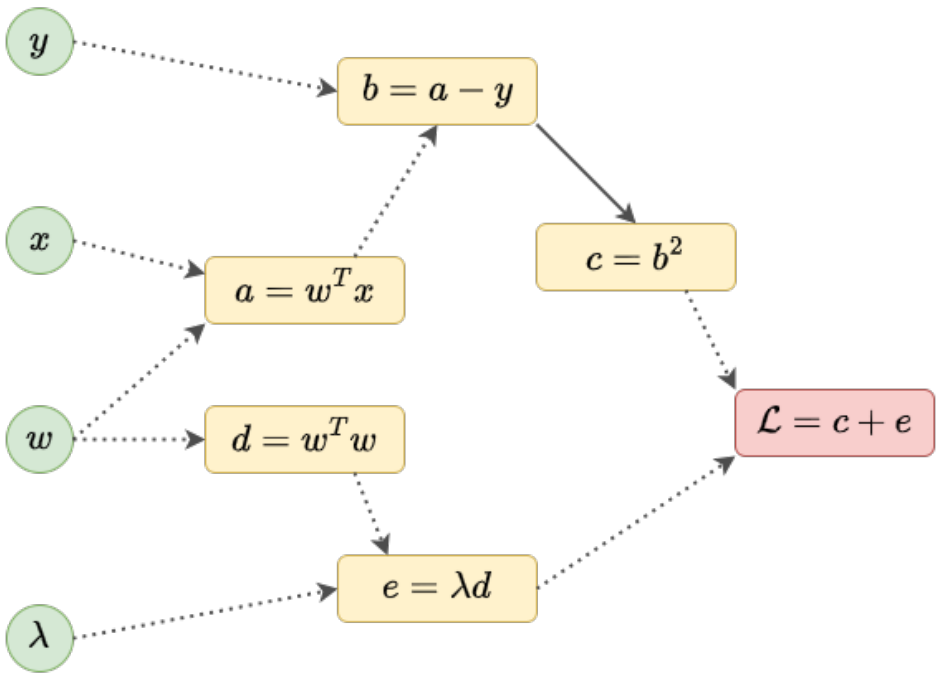
Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$b = a - y = 11; \quad e = \lambda d = 125$$

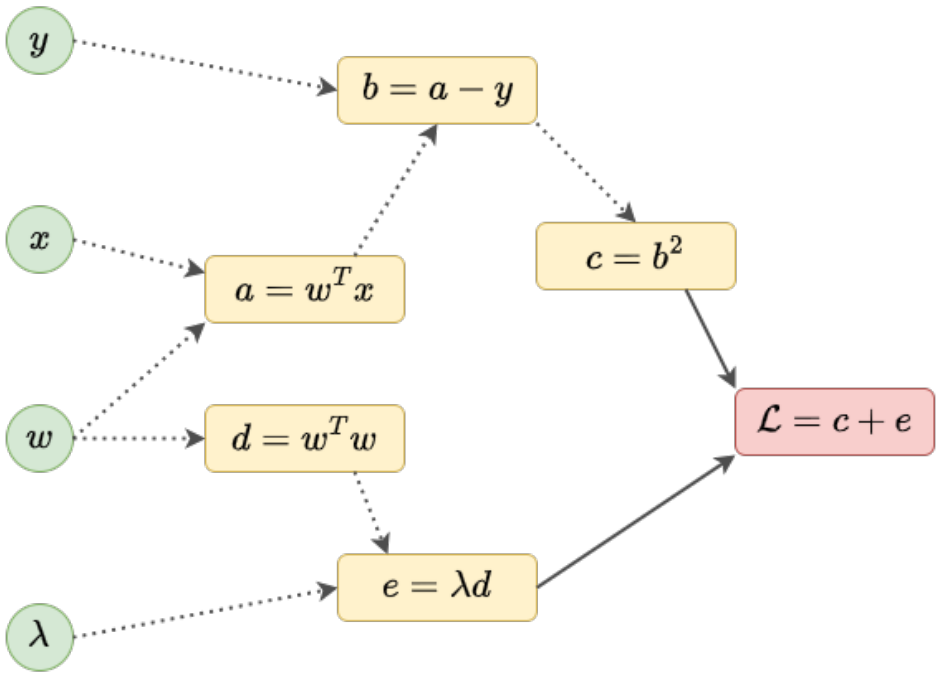
Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$c = b^2 = 11^2 = 121$$

Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$\mathcal{L} = c + e = 121 + 125 = 246$$

Проход назад

Проход назад: итеративно **справа-налево**

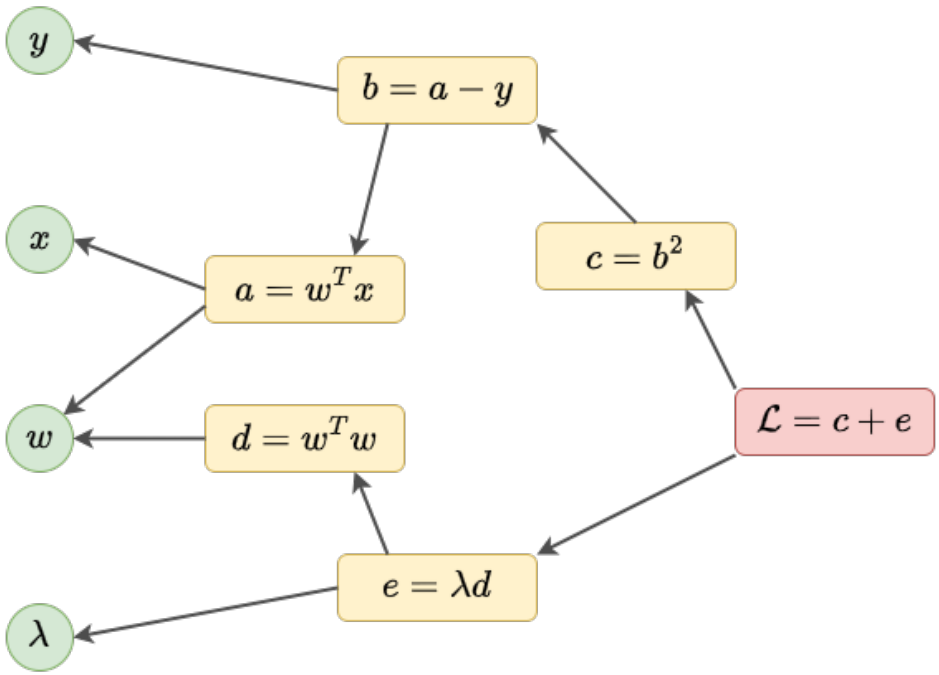
- вычисляются производные итога от предыдущих переменных графа как функции.
- подстановкой переменных получаем численные значения производных.
  - после получения чисел функции уже не нужны.

Используем:

$$\mathcal{L}(w) = A(B(w)): \qquad \frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A(B)}{\partial B}\frac{\partial B}{\partial w}$$

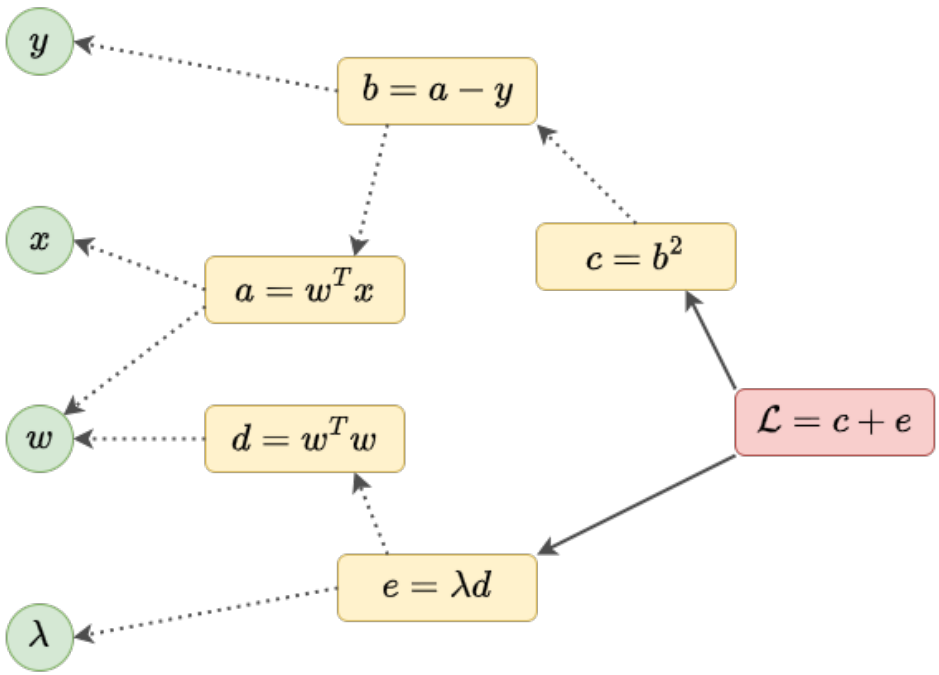
$$\mathcal{L}(w) = A(B_1(w), B_2(w)): \qquad \frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial A}{\partial B_1}\frac{\partial B_1}{\partial w} + \frac{\partial A}{\partial B_2}\frac{\partial B_2}{\partial w}$$

Проход назад: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$a = 11; \quad b = 11; \quad c = 121; \quad d = 25; \quad e = 125$$

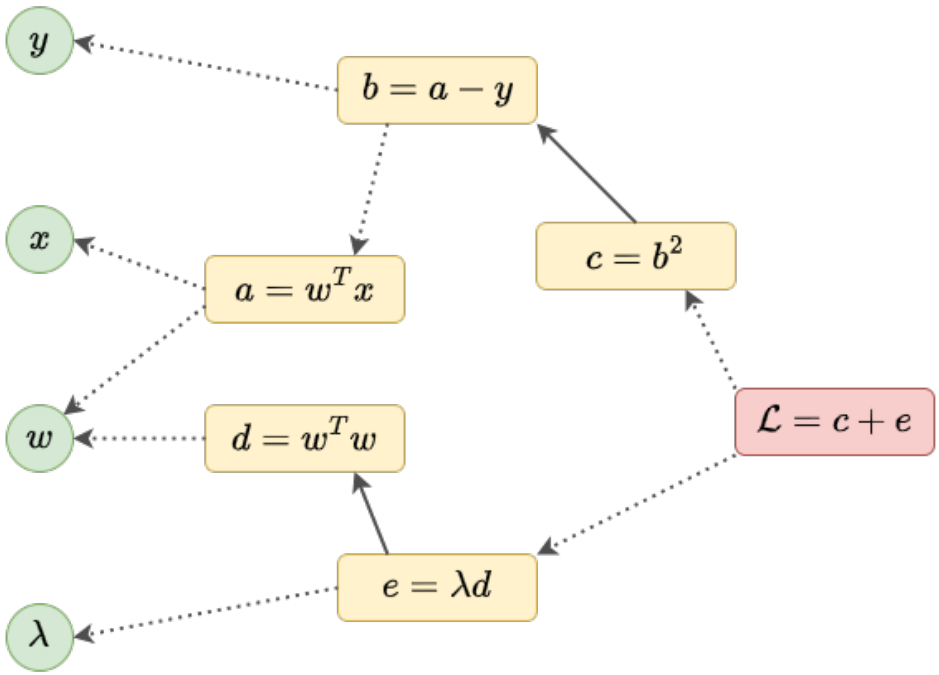
Проход назад: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$a = 11; \quad b = 11; \quad c = 121; \quad d = 25; \quad e = 125$$

$$\frac{\partial \mathcal{L}}{\partial c} = 1, \quad \frac{\partial \mathcal{L}}{\partial e} = 1$$

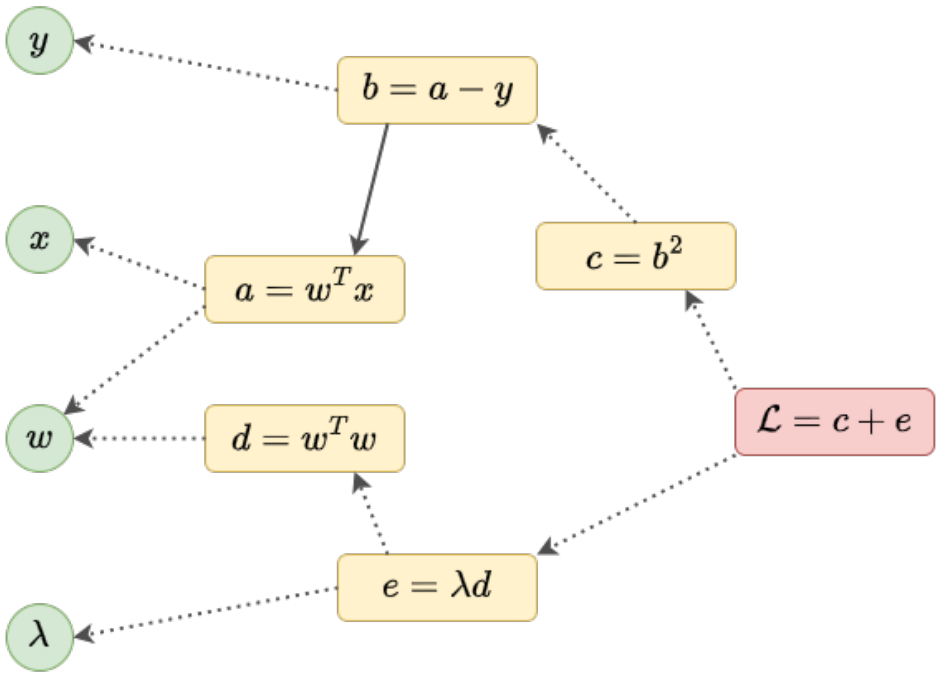
Проход назад: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$a = 11; \quad b = 11; \quad c = 121; \quad d = 25; \quad e = 125$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}(c)}{\partial b} = \frac{\partial \mathcal{L}}{\partial c}\frac{\partial c}{\partial b} = 1 \cdot 2b = 22; \quad \frac{\partial \mathcal{L}}{\partial d} = \frac{\partial \mathcal{L}(e)}{\partial d} = \frac{\partial \mathcal{L}}{\partial e}\frac{\partial e}{\partial d} = 1 \cdot \lambda = 5$$

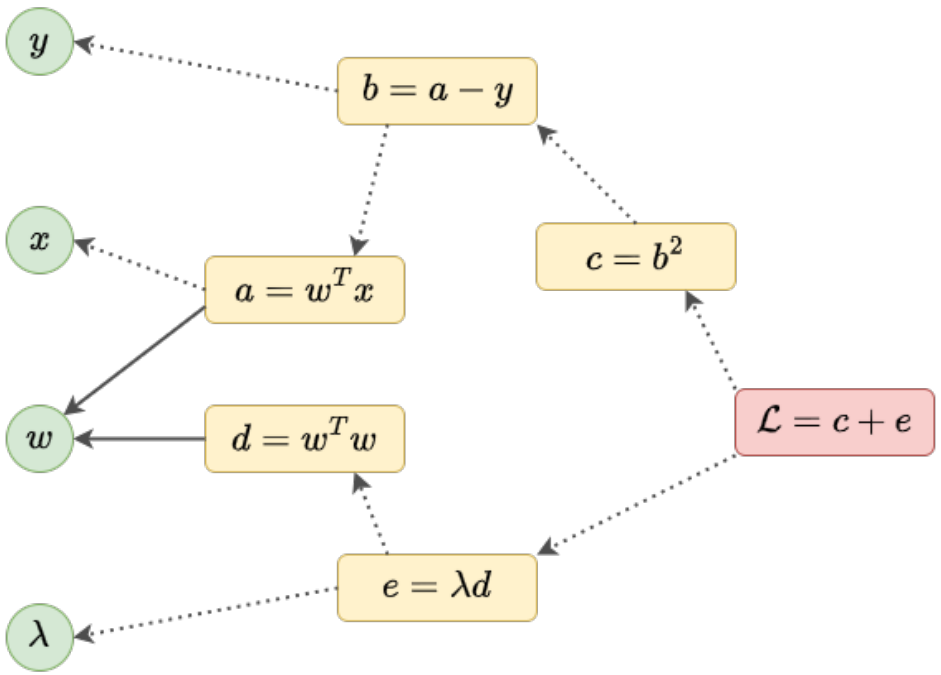
Проход назад: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$a = 11; \quad b = 11; \quad c = 121; \quad d = 25; \quad e = 125$$

$$\frac{\partial \mathcal{L}}{\partial a} = \frac{\partial \mathcal{L}(b)}{\partial a} = \frac{\partial \mathcal{L}}{\partial b}\frac{\partial b}{\partial a} = 22 \cdot 1 = 22$$

Проход назад: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



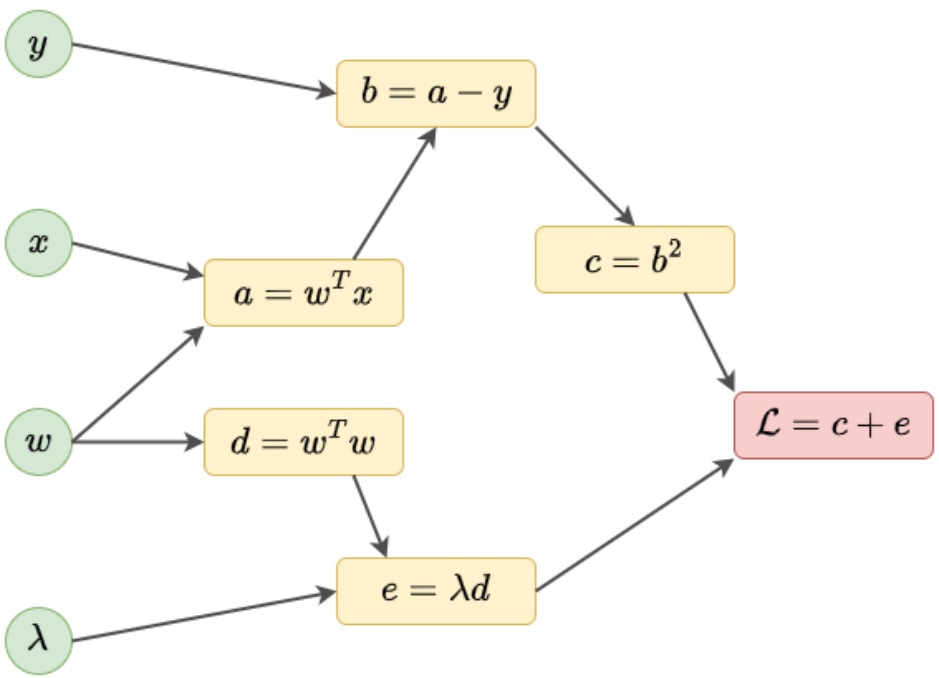
$$a = 11; \quad b = 11; \quad c = 121; \quad d = 25; \quad e = 125$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}(a,d)}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial a}\frac{\partial a}{\partial \mathbf{w}} + \frac{\partial \mathcal{L}}{\partial d}\frac{\partial d}{\partial \mathbf{w}} = 22 \cdot \mathbf{x} + 5 \cdot 2\mathbf{w}$$
$$= 22 \cdot [1,2] + 10 \cdot [3,4] = [22,44] + [30,40] = [52,84]$$

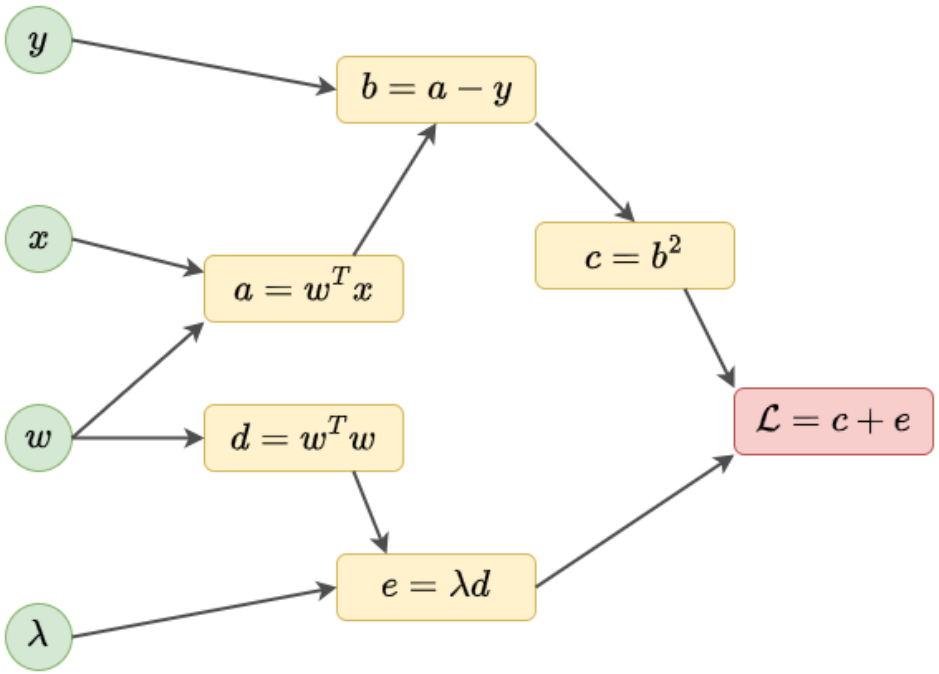
Содержание

Метод forward mode

- Метод forward mode основан только на проходе вперёд (forward pass)
  - эффективнее по памяти: не нужно хранить промежут. переменные & преобразования
- Вычисляются $\frac{\partial a}{\partial w}, \frac{\partial d}{\partial w}, \frac{\partial b}{\partial w}, \frac{\partial e}{\partial w}, \frac{\partial c}{\partial w}, \frac{\partial \mathcal{L}}{\partial w}$

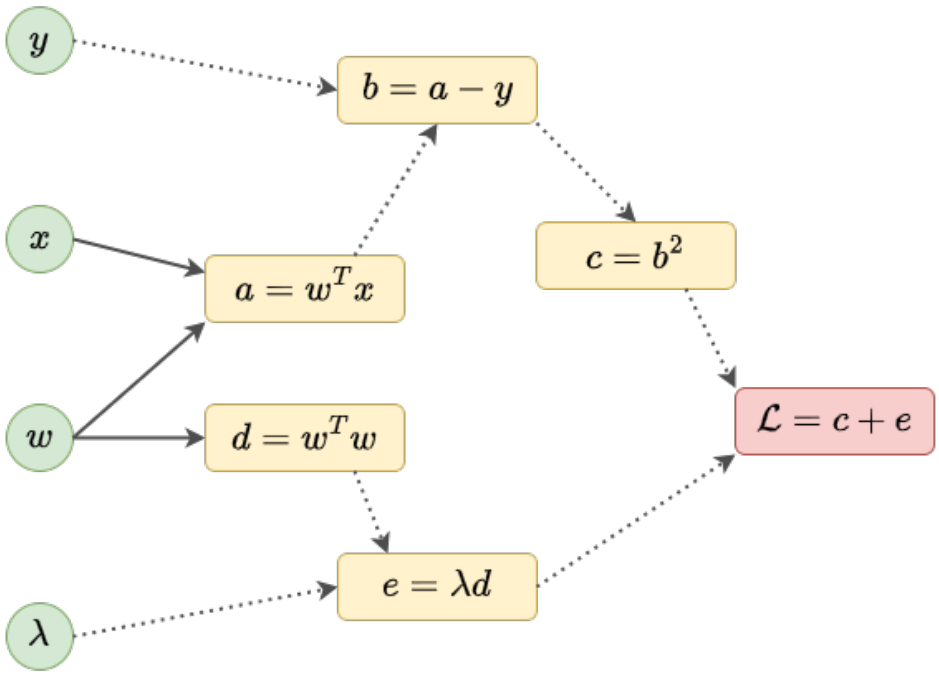


Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

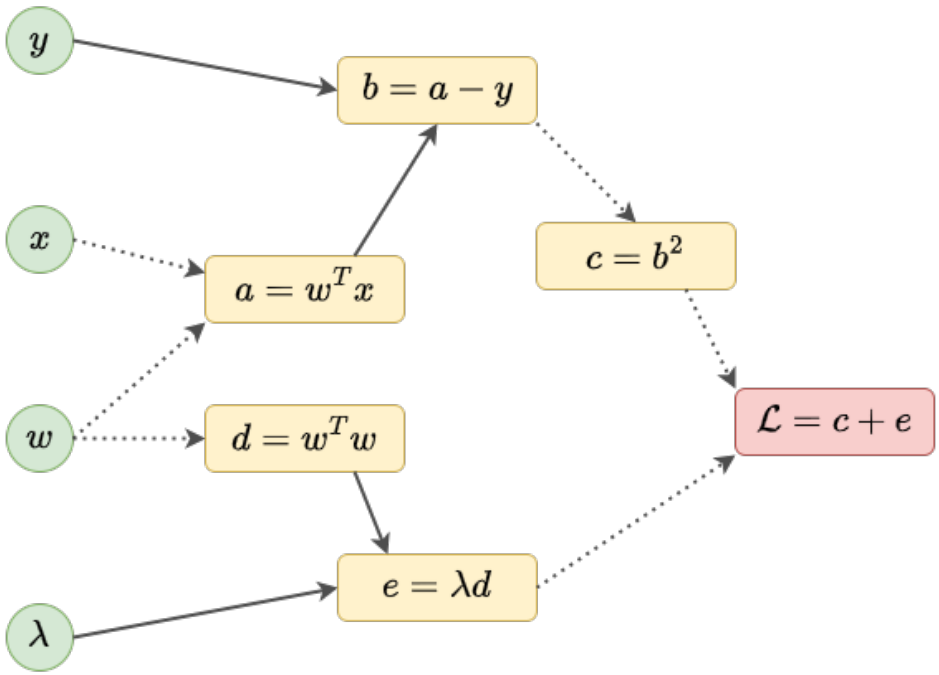
Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$\frac{\partial a}{\partial \mathbf{w}} = x = [1, 2]; \qquad \frac{\partial d}{\partial \mathbf{w}} = 2\mathbf{w} = [6, 8]$$

Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$

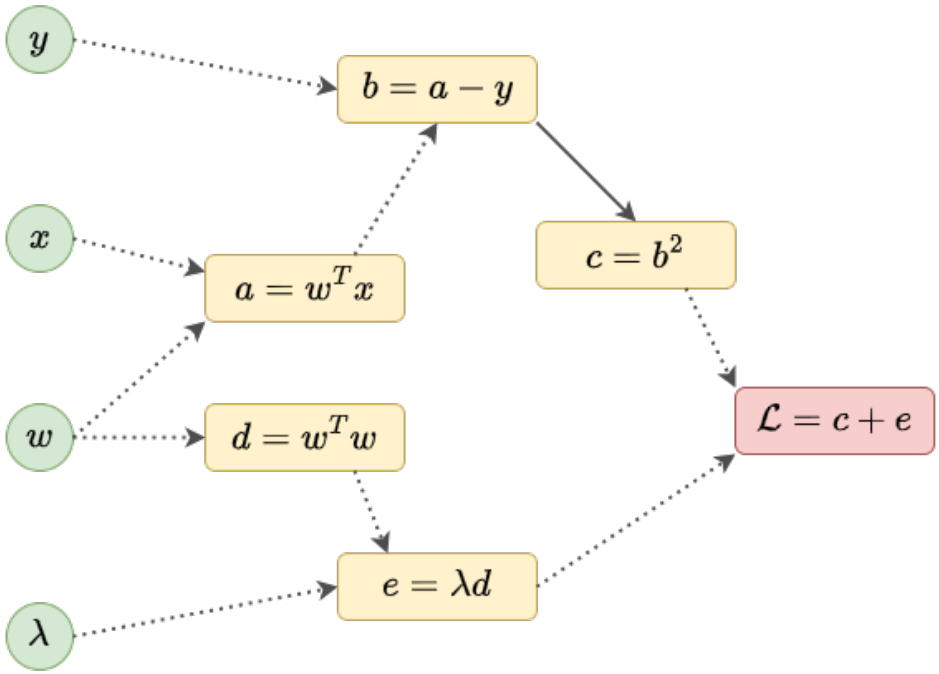


$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$\frac{\partial b}{\partial \mathbf{w}} = \frac{\partial b(a)}{\partial \mathbf{w}} = \frac{\partial b}{\partial a}\frac{\partial a}{\partial \mathbf{w}} = 1 \cdot [1, 2] = [1, 2];$$

$$\frac{\partial e}{\partial \mathbf{w}} = \frac{\partial e(d)}{\partial \mathbf{w}} = \frac{\partial e}{\partial d}\frac{\partial d}{\partial \mathbf{w}} = \lambda \cdot [6, 8] = [30, 40]$$

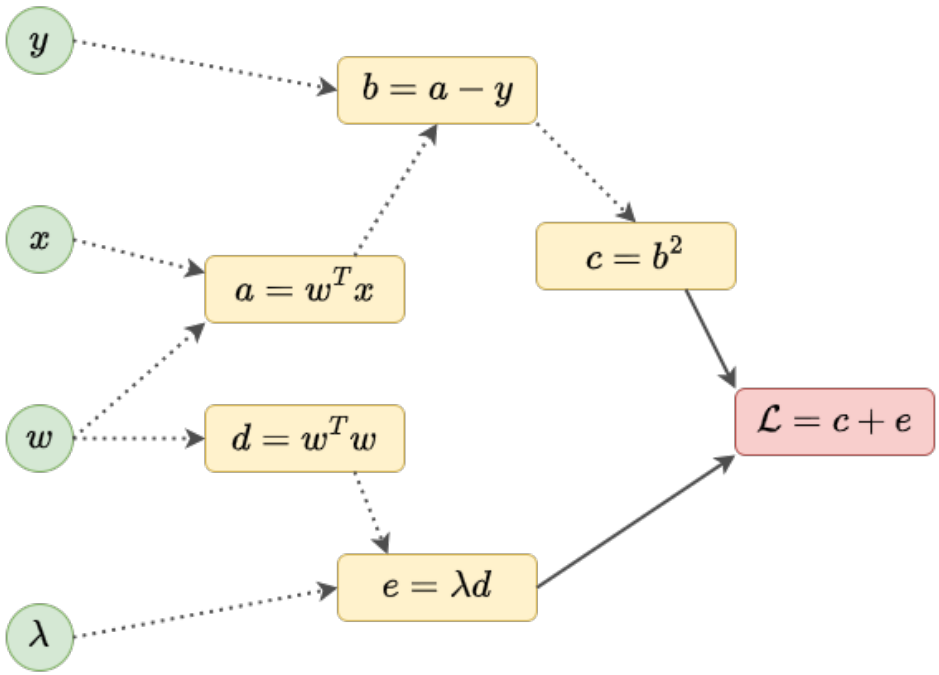
Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$



$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$\frac{\partial c}{\partial \mathbf{w}} = \frac{\partial c(b)}{\partial \mathbf{w}} = \frac{\partial c}{\partial b}\frac{\partial b}{\partial \mathbf{w}} = 2b \cdot [1, 2] = 22 \cdot [1, 2] = [22, 44]$$

Проход вперёд: $\mathcal{L}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2 + \lambda\mathbf{w}^T\mathbf{w}$

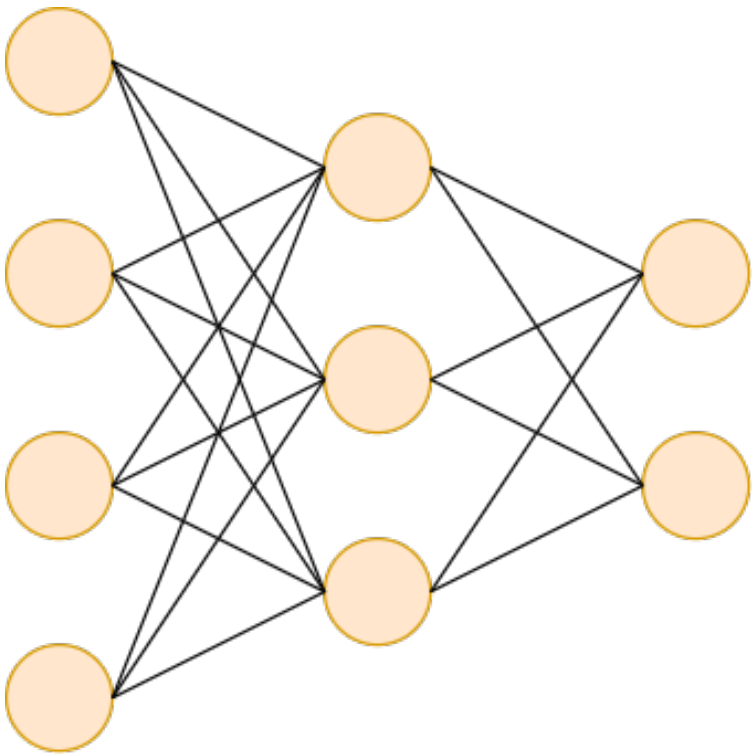


$$y = 0,\ \mathbf{x} = [1, 2],\ \mathbf{w} = [3, 4],\ \lambda = 5$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}(c, e)}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial c}\frac{\partial c}{\partial \mathbf{w}} + \frac{\partial \mathcal{L}}{\partial e}\frac{\partial e}{\partial \mathbf{w}} = 1 \cdot [22, 44] + 1 \cdot [30, 40] = [52, 84]$$

Эффективнее обратное распространение ошибки

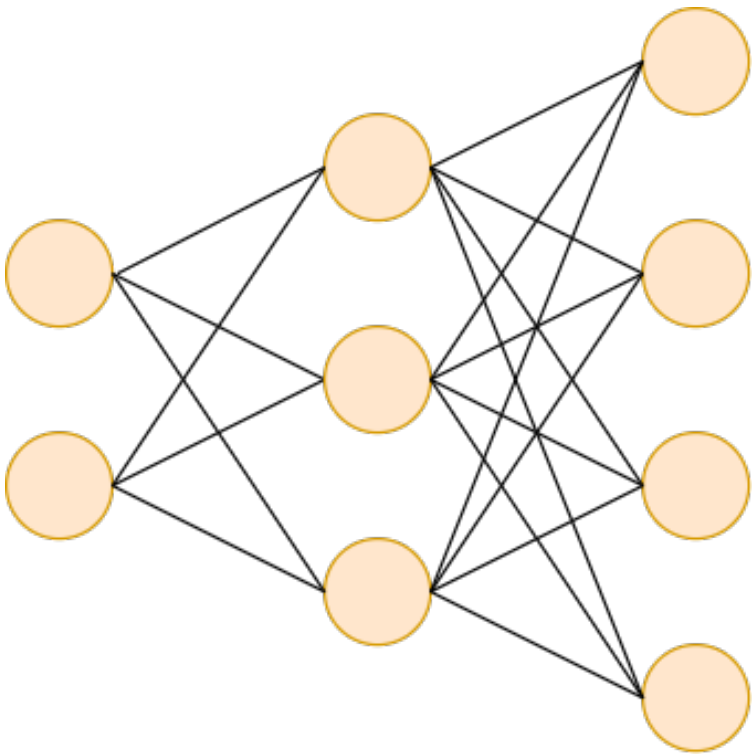
Сложность $\propto$ числу входов:



Настройка нейросетей, где выход $\mathcal{L} \in \mathbb{R}$, а весов много.

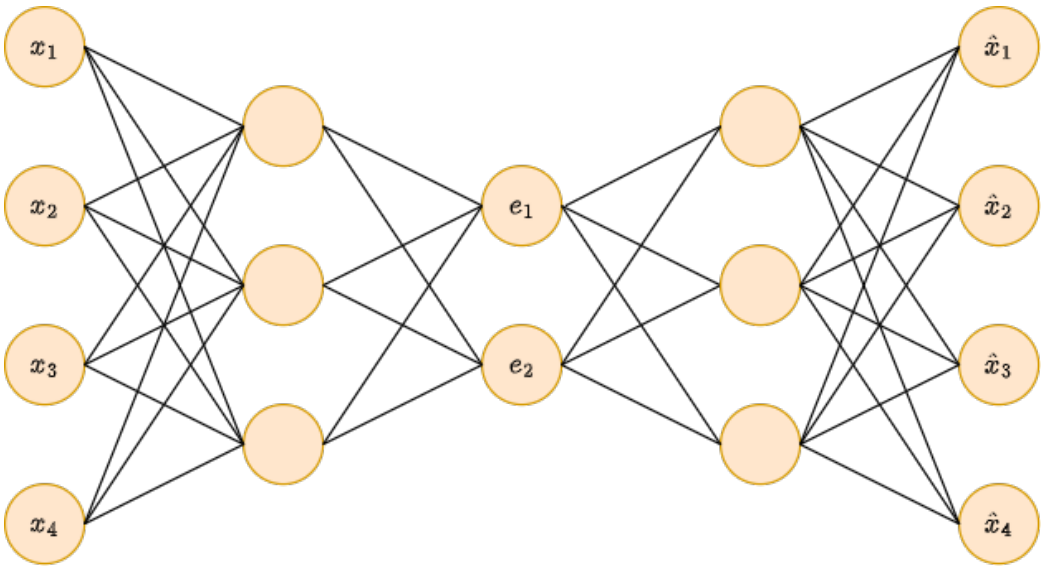
Эффективенее forward mode

Сложность $\propto$ числу выходов:



Чувствительность выходов к изменению отдельных пар-ров.

Эффективнее комбинация



$$\frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{x}} = \frac{\partial \hat{\mathbf{x}}(\mathbf{e})}{\partial \mathbf{x}} = \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial \mathbf{x}}$$

Эффективнее посчитать

- $\frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{e}}$ через обратное распространение ошибки
- $\frac{\partial \mathbf{e}}{\partial \mathbf{x}}$ - через forward mode.

Пример в PyTorch

$$\mathcal{L}(x,y) = x^2 + xy; \quad x = 2; \quad y = 3$$

$$\mathcal{L} = 10; \quad \frac{\partial \mathcal{L}}{\partial x} = 2x + y = 7; \quad \frac{\partial \mathcal{L}}{\partial y} = x = 2$$

```
import torch

# requires_grad=True - считать по ним градиенты
x = torch.tensor(2.0, requires_grad=True)
y = torch.tensor(3.0, requires_grad=True)
f = x * x + x * y

# обратное распространение ошибки:
f.backward() # по скаляру градиент сразу по всем входам

print("f(x,y) =", f.item())       # 10
print("df/dx =", x.grad.item())   # 7
print("df/dy =", y.grad.item())   # 2
```

Пример в PyTorch

$$\mathcal{L}(x,y) = x^2 + xy; \quad x = 2; \quad y = 3$$

$$\mathcal{L} = 10; \quad \frac{\partial \mathcal{L}}{\partial x} = 2x + y = 7; \quad \frac{\partial \mathcal{L}}{\partial y} = x = 2$$

```
import torch
from torch.autograd import forward_ad

x = torch.tensor(2.0)
y = torch.tensor(3.0)

with forward_ad.dual_level():  # dx/dx = 1, dy/dx = 0
    dual_x = forward_ad.make_dual(x, torch.tensor(1.0))
    dual_y = forward_ad.make_dual(y, torch.tensor(0.0))
    f = dual_x * dual_x + dual_x * dual_y
    dual_f = forward_ad.unpack_dual(f)

print("f(x,y) =", dual_f.primal.item())  # 10
print("df/dx =", dual_f.tangent.item())  # 7
```

Пример в PyTorch

$$\mathcal{L}(x,y) = x^2 + xy; \quad x = 2; \quad y = 3$$

$$\mathcal{L} = 10; \quad \frac{\partial \mathcal{L}}{\partial x} = 2x + y = 7; \quad \frac{\partial \mathcal{L}}{\partial y} = x = 2$$

```
import torch
from torch.autograd import forward_ad

x = torch.tensor(2.0)
y = torch.tensor(3.0)

with forward_ad.dual_level():  # dx/dx = 0, dy/dx = 1
    dual_x = forward_ad.make_dual(x, torch.tensor(0.0))
    dual_y = forward_ad.make_dual(y, torch.tensor(1.0))
    f = dual_x * dual_x + dual_x * dual_y
    dual_f = forward_ad.unpack_dual(f)

print("f(x,y) =", dual_f.primal.item())  # 10
print("df/dx =", dual_f.tangent.item())  # 2
```

Заключение

- **Автоматическое дифференцирование** - эффективный расчёт градиентов за линейное по числу связей время.
  - Библиотеки: PyTorch, TensorFlow, JAX.
- Методы:
  - **обратное распространение ошибки**
    - проход вперёд и назад
    - сложность $\propto$ числу выходов
    - применение: настройка нейросетей
  - **forward mode**
    - только проход вперёд
    - сложность $\propto$ числу входов
    - эффективнее по памяти
    - применение: зависимость изменений от параметра