Дифференцирование через решение оптимизационных задач для настройки гиперпараметров

Лебедь Федор Сергеевич, 517 группа ВМК

Московский Государственный Университет им. М.В. Ломоносова

21 мая 2021 г.

Постановка задачи

w – веса модели,

λ – вектор гиперпараметров,

$L_{tr}(w, \lambda)$ – функция ошибки на обучении,

$L_{te}(w, \lambda)$ – функция ошибки на тесте.

$$\begin{cases} L_{te}(w_*(\lambda), \lambda) \to \min\limits_{\lambda} \\ w_*(\lambda) = \arg\min\limits_{w} L_{tr}(w, \lambda) \end{cases}$$

Методы дифференцирования

Обозначим $f(x) : V \to W$.

Способы вычисления $\dfrac{\partial f}{\partial x}(x)$:

1. численное,
2. дифференциальное исчисление,
3. неявное.

Численное

- Считается по направлению d,
- необходимо выбирать ε,
- классическая формула,

$$\boxed{\frac{\partial f}{\partial x}(x) \cdot d} = \frac{f(x+\varepsilon d) - f(x-\varepsilon d)}{2\varepsilon} + O(\varepsilon^2)$$

$$\boxed{\frac{\partial f}{\partial x}(x) \cdot d} = \frac{Im(f(x+i\varepsilon d))}{\varepsilon} + O(\varepsilon^2),$$

- более устойчивая формула.

Дифференциальное исчисление

- Предпологается $f(x) = (f_n \circ \ldots \circ f_1)(x),\; f_k(x_k)$,
- Якобиан,

$$
\begin{aligned}
x_1 &= x, \\
x_{k+1} &= f_k(x_k), \\
\boxed{\frac{\partial f}{\partial x}(x)} &= \frac{\partial f_n}{\partial x_n}(x_n) \circ \ldots \circ \frac{\partial f_1}{\partial x_1}(x_1), \\
\boxed{\left(\frac{\partial f}{\partial x}(x)\right)^T} &= \left(\frac{\partial f_1}{\partial x_1}(x_1)\right)^T \circ \ldots \circ \left(\frac{\partial f_n}{\partial x_n}(x_n)\right)^T,
\end{aligned}
$$

- сопряженный Якобиан.

Неявное

- Предполагается $g(f(x), x) = 0$.

$$0 = \frac{\partial g}{\partial f}(f(x), x) \cdot \frac{\partial f}{\partial x}(x) \cdot dx + \frac{\partial g}{\partial x}(f(x), x) \cdot dx,$$

$$\frac{\partial f}{\partial x}(x) = -\left(\boxed{\frac{\partial g}{\partial f}(f(x), x)} \right)^{-1} \cdot \frac{\partial g}{\partial x}(f(x), x).$$

- Невырожденность требуется в «теореме о неявной функции».

Дифференцирование через методы оптимизации

Теория

$$\begin{cases} L_{te}(w_*(\lambda), \lambda) \\ w_*(\lambda) = \boxed{\arg\min_w L_{tr}(w, \lambda)} \end{cases}$$

Практика

$$\begin{cases} L_{te}(w_*(\lambda), \lambda) \\ w_*(\lambda) = \boxed{\begin{cases} \textit{l-BFGS}(L_{tr}, w_0, \lambda) \\ \textit{Adam}(L_{tr}, w_0, \lambda) \\ \dots \end{cases}} \end{cases}$$

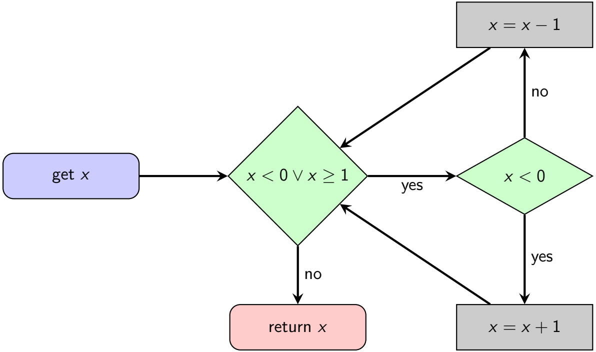
- w_0 – начальное приближение.

Дифференцирование алгоритмов

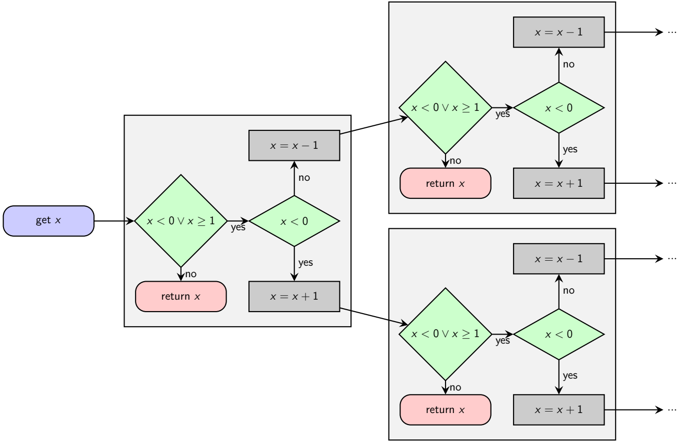
Алгоритм

```python

def mod1(x):
    while x < 0 or x >= 1:
        if x < 0:
            x += 1
        else:
            x -= 1
    return x

```

Алгоритмический граф



Вычислительный граф



1. Перенумеруем выходы return x.
2. Вычисляемые в них функции обозначим за $mod1_k(x)$.
3. Номер выхода return x для данного входа обозначим за $c(x)$.
4. $mod1(x) = mod1_{c(x)}(x)$.

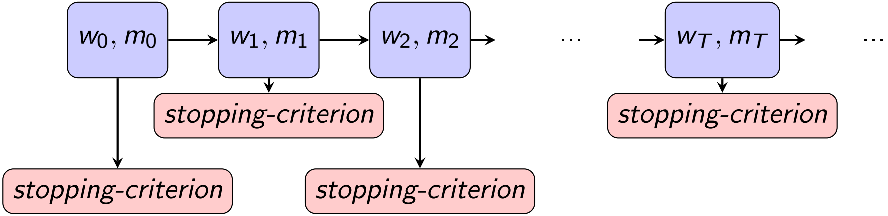
Дифференцирование методов оптимизации

Общий алгоритм оптимизации

1. w_0
2. $t \leftarrow 0$
3. $m_0 \leftarrow \textit{init-memory}(w_0)$
4. **while not** *stopping-criterion*
   - 4.1 $d_t \leftarrow \textit{get-direction}(m_{t-1})$
   - 4.2 $\alpha_t \leftarrow \textit{get-learning-rate}(w_{t-1}, m_{t-1})$
   - 4.3 $w_t \leftarrow w_{t-1} + \alpha_t d_t$
   - 4.4 $m_t \leftarrow \textit{update-memory}(m_{i-t}, w_t)$
   - 4.5 $t \leftarrow t + 1$

- Источники разрывности.
- *get-learning-rate* может быть:
  (a) фиксированным,
  (b) неточным оптимальным,
  (c) оптимальным.

Пример: *stopping-criterion* имеет вид $||\nabla_w L_{tr}(w, \lambda)|| < \varepsilon$.



Проблемы

- зависимость от w_0,
- аккумуляция шума в цепочке $\{w_t\}_t$

Проблема длинных цепочек:

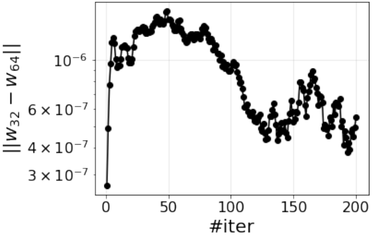


Рис.: Пример **схождения** истинной и посчитанной цепочек (ADAM)

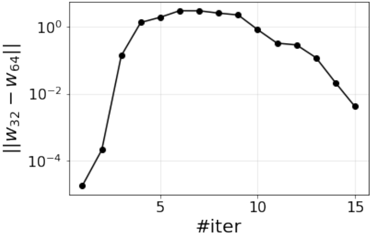


Рис.: Пример **расхождения** истинной и посчитанной цепочек (LBFGS)

Вычислительная сложность

Методы оптимизации последовательно строят цепочку $\{w_t, m_t\}_{t=0}^{T}$:

$$w_0, m_0 \longrightarrow w_1, m_1 \longrightarrow w_2, m_2 \rightarrow \quad \cdots \quad \rightarrow w_T, m_T$$

Back Propagation

- Время $O(T)$
- Память $O(T)$

$O(T)$ по памяти – плохо:

- 5ms на батч (Titan, resnet101, MNIST, model size=170Mb, bs=64)
- 200ms на загрузку из памяти (Samsung 870 EVO)

Инвертирование динамики

прямой Adam

1. $g_t \equiv \nabla_w L_{tr}(w_t, \lambda)$
2. w_0
3. $m_0 \leftarrow g_0$
4. $v_0 \leftarrow g_0 \odot g_0$
5. **for** t **in** $[1..T]$
   - 5.1 $w_t \leftarrow w_{t-1} - \alpha \dfrac{m_{t-1}}{\sqrt{v_{t-1}} + \varepsilon}$
   - 5.2 $m_t \leftarrow \gamma_1 m_{t-1} + (1-\gamma_1) g_t$
   - 5.3 $v_t \leftarrow \gamma_2 v_{t-1} + (1-\gamma_2) g_t \odot g_t$

инвертированный Adam

1. $g_t \equiv \nabla_w L_{tr}(w_t, \lambda)$
2. w_T, m_T, v_T
3. **for** t **in** $[T..1]$
   - 3.1 $m_{t-1} \leftarrow \dfrac{1}{\gamma_1} \left(m_t - (1-\gamma_1) g_t\right)$
   - 3.2 $v_{t-1} \leftarrow \dfrac{1}{\gamma_2} \left(v_t - (1-\gamma_2) g_t \odot g_t\right)$
   - 3.3 $w_{t-1} \leftarrow w_t + \alpha \dfrac{m_{t-1}}{\sqrt{v_{t-1}} + \varepsilon}$

Floating point arithmetic (float)

$$M = \textit{бит в мантиссе},$$
$$E = \textit{бит в экспоненте},$$
$$s_x \in -1, 1,$$
$$m_x \in \{0 ... 2^M - 1\},$$
$$e_x \in \{0 ... 2^E - 1\},$$
$$x = \boxed{s_x \frac{m_x}{2^M} 2^{e_x - 2^{E-1}}},$$

- сложная арифметика,
- потери при всех операциях.

Fixed point arithmetic (fp)

$$M = \textit{бит в мантиссе},$$
$$D = \textit{бит после запятой},$$
$$s_x \in -1, 1,$$
$$m_x \in \{0 ... 2^M - 1\},$$
$$x = \boxed{s_x \frac{m_x}{2^D}},$$

- простая арифметика,
- потери только при делении и умножении.

Пример операций для fixed point arithmetic:

$$n, d \in \mathbb{N},$$

$$\boxed{x = s_x \frac{m_x}{2^D}}, \boxed{y = s_y \frac{m_y}{2^D}},$$

$$s_x = s_y = 1,$$

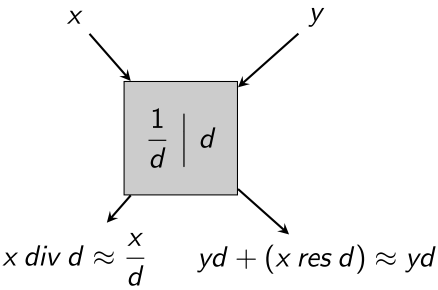
$$x + y = \underbrace{(m_x + m_y)}_{m_{x+y}} \frac{1}{2^D},$$

$$x - y = \underbrace{(m_x - m_y)}_{m_{x-y}} \frac{1}{2^D},$$

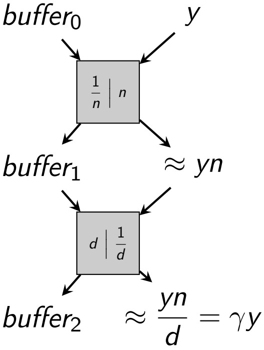
$$nx = \underbrace{(nm_x)}_{m_{nx}} \frac{1}{2^D},$$

$$\frac{x}{d} = \underbrace{(m_x \, div \, d)}_{m_{\frac{x}{d}}} \frac{1}{2^D},$$

I. Введем операцию обратимого деления в $\mathbb{N}$, $x, y \in \mathbb{N}$:



II. Введем понятие буфера $\gamma = \dfrac{n}{d}$:



Вычислительная сложность

- Время $O(T)$
- Память $O(T\sum_k \log_2(\frac{1}{\gamma_k}))$

Для $\gamma = 0.99$ и *float*64 память $= O(10^{-4}T)$.

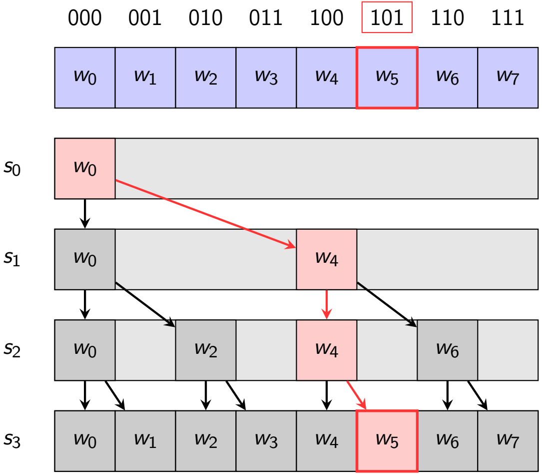
Древесное инвертирование

Хранить всю цепочку – накладно,

$$\boxed{w_0, m_0} \longrightarrow \boxed{w_1, m_1} \longrightarrow \boxed{w_2, m_2} \rightarrow \quad \cdots \quad \rightarrow \boxed{w_T, m_T}$$

будем хранить некоторые ее фрагменты $\{s_k\}_k$.

Схема фрагментов



- Шаг направо ($\rightarrow$):
  1. подняться до первого узла с левым потомком,
  2. спуститься 1 раз направо и остальные влево.
- Шаг налево ($\leftarrow$):
  1. подняться до первого узла с правым потомком,
  2. спуститься 1 раз влево и остальные направо.

Вычислительная сложность

- Время $O(T \log_2 T)$
- Память $O(\log_2 T)$

Итого

Алгоритм оптимизации гиперпараметров

1. w_0, m_0, λ
2. $t \leftarrow 0$
3. **while not** *stopping-criterion*
   3.1 **repeat** k **times**
      3.1.1 $w_{t+1}, m_{t+1}, \nabla_\lambda \leftarrow$ *diff-argmin* $(L_{te}, L_{tr}, w_t, m_t, \lambda)$
      3.1.2 update λ
   3.2 $t \leftarrow t + 1$

- Жадный алгоритм,
- *diff-argmin* – процедура подсчета $\dfrac{d}{d\lambda}L_{te}$ через развернутую оптимизацию.

Пример

LeNet на MNIST.

- 20 эпох оптимизации прямых параметров.
- 20 эпох оптимизации гипер-параметров (100 если деревом).

Метрика	без рег.	общая рег.	индивидуальная рег.
Cross-Entropy ($\downarrow$)	1.4723	1.4769	**1.4651**
1 - AUC ($\downarrow$)	0.0102	0.0135	**0.0017**

Таблица: Результаты оптимизации коэффициэнтов регуляризации

Неявное дифференцирование

$$\begin{cases} L_{te}(w_*(\lambda), \lambda) \to \min\limits_{\lambda} \\ w_*(\lambda) = \arg\min\limits_{w} L_{tr}(w, \lambda) \end{cases}$$

$$0 = \frac{\partial L_{tr}}{\partial w}(w_*, \lambda),$$

$$0 = \frac{\partial^2 L_{tr}}{\partial w \partial w}(w_*, \lambda) \cdot dw_* + \frac{\partial^2 L_{tr}}{\partial w \partial \lambda}(w_*, \lambda) \cdot d\lambda,$$

$$dw_* = -\Big(\underbrace{\frac{\partial^2 L_{tr}}{\partial w \partial w}(w_*, \lambda)}_{\succcurlyeq 0,\ \text{т.к. опт.}}\Big)^{-1} \cdot \frac{\partial^2 L_{tr}}{\partial w \partial \lambda}(w_*, \lambda) \cdot d\lambda,$$

$$
\begin{aligned}
dL_{te} &= \frac{\partial L_{te}}{\partial w}(w_*, \lambda) \cdot dw_* + \frac{\partial L_{te}}{\partial \lambda}(w_*, \lambda) \cdot d\lambda \\
&= \left(-\frac{\partial L_{te}}{\partial w}(w_*, \lambda) \cdot \Bigl(\underbrace{\frac{\partial^2 L_{tr}}{\partial w \partial w}(w_*, \lambda)}_{\succcurlyeq 0,\ \text{т.к. опт.}} \Bigr)^{-1} \cdot \frac{\partial^2 L_{tr}}{\partial w \partial \lambda}(w_*, \lambda) + \frac{\partial L_{te}}{\partial \lambda}(w_*, \lambda) \right) \cdot d\lambda
\end{aligned}
$$

$$
\nabla_\lambda = -\frac{\partial^2 L_{tr}}{\partial \lambda \partial w}(w_*, \lambda) \cdot \Bigl(\overbrace{\frac{\partial^2 L_{tr}}{\partial w \partial w}(w_*, \lambda)}^{\succcurlyeq 0,\ \text{т.к. опт.}} \Bigr)^{-1} \cdot \nabla_w L_{te}(w_*, \lambda) + \nabla_\lambda L_{te}(w_*, \lambda)
$$

- решаем методом сопряженных градиентов.

Вычислительная сложность

- Время $O(T) + O(CG)$
- Память $O(1)$

Проблемы

- множественные минимумы,
- сложность CG,
- вырожденность $\frac{\partial^2 L_{tr}}{\partial w \partial w}(w_*, \lambda)$,
- неприменимость к стохастическому случаю.

Итого

Алгоритм оптимизации гиперпараметров

1. w_0, m_0, λ
2. $t \leftarrow 0$
3. **while not** *stopping-criterion*
   - 3.1 $w_{t+1}, m_{t+1} \leftarrow argmin(L_{tr}, w_t, m_t, \lambda)$
   - 3.2 $\nabla_\lambda \leftarrow$ *diff-CG* $(L_{tr}, w_{t+1}, \lambda)$
   - 3.3 update λ
   - 3.4 $t \leftarrow t + 1$

- Честный алгоритм,
- *diff-CG* – процедура подсчета $\frac{d}{d\lambda} L_{te}$ путем неявного дифференцирования.

Пример

KLR на синтетическом датасете.

$$f(x) = b + \sum_{i=1}^{n} a_i K(x_i, x),$$

$$-y^T K a + \mathbb{1}^T \log(1 + \exp(Ka + b\mathbb{1})) + \frac{\lambda}{2} a^T K a \to \min_{a,b},$$

$$K = \left[K(x_i, x_j)\right]_{i,j}.$$

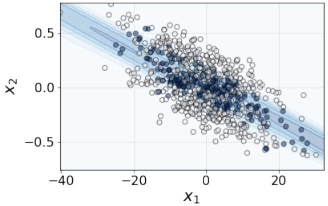


Рис.: Результат подбора ядровой функции

Метрика	без подбора ядра.	с подбором ядра.
Cross-Entropy (↓)	0.63	**0.44**
AUC (↑)	0.53	**0.84**

Таблица: Результаты оптимизации коэффициэнтов регуляризации

Заключение

- Дифференцирование через методы оптимизации для стохастических задач.
- Неявное дифференцирование для сильно выпуклых.