

Тест начат среда, 10 января 2024, 23:34

Состояние Завершены

Завершен среда, 10 января 2024, 23:56

Прошло времени 21 мин. 29 сек.

Оценка 7,00 из 7,00 (100%)

### Вопрос 1

Выполнен

Баллов: 1,00 из 1,00

Вы обучаете одномерную логистическую регрессию  $\hat{y}(x) = \frac{1}{1+e^{-wx-b}}$ , то есть  $w \in \mathbb{R}$  - это скаляр (число).

$x$  -- единственный признак входного объекта,  $y(x) \in \{0, 1\}$  -- настоящий класс объекта  $x$ ,  $0 < \hat{y}(x) < 1$  - предсказанная вероятность того, что  $x$  принадлежит к классу 1.

В качестве функции потерь Вы используете бинарную кросс-энтропию  $BCE(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$ .

Найдите **в общем виде** производную функции потерь по  $w$   $\frac{\partial BCE(\hat{y}, y)}{\partial w}$  и запишите в ответ её формулу. Для обозначений используйте латинские буквы  $y, x, w, b$  в нижнем регистре.

Ответ должен компилироваться в SymPy. Вам могут понадобиться операции деления  $/$ , умножения  $*$ , сложения  $+$ , вычитания  $-$  и взятия экспоненты  $e^z \exp(z)$ .

Пример ответа:

`exp(x * w + b) - y`

Для отладки своего выражения можете использовать следующий фрагмент кода (этот код в ответ вставлять не надо).

```
import sympy.parsing.sympy_parser

sample_expr_str = '<your solution here>'
sample_expr = sympy.parsing.sympy_parser.parse_expr(sample_expr_str)
sample_value = sample_expr.evalf(subs=dict(x=0.5, y=1, w=4, b=1))
print(sample_value)
```

Если Вы решили не считать производную самостоятельно, а используете для нахождения производной библиотеку SymPy, то упростите результат (метод `.simplify()`), иначе правильный ответ может не быть принят.

`x*(-y*exp(b + w*x) - y + exp(b + w*x))/(exp(b + w*x) + 1)`

$$\frac{x \cdot ((-y) \cdot \exp(b + w \cdot x) - y + \exp(b + w \cdot x))}{\exp(b + w \cdot x) + 1}$$

✔ Верный ответ, так держать!

## Вопрос 2

Выполнен

Баллов: 1,00 из 1,00

Теперь найдите в общем виде производную функции потерь по  $b \frac{\partial BCE(\hat{y}, y)}{\partial b}$  и запишите в ответ её формулу.

Пример ответа:

`exp(x * w + b) - y`

Для отладки своего выражения можете использовать следующий фрагмент кода (этот код в ответ вставлять не надо).

```
import sympy.parsing.sympy_parser

sample_expr_str = '<your solution here>'
sample_expr = sympy.parsing.sympy_parser.parse_expr(sample_expr_str)
sample_value = sample_expr.evalf(subs=dict(x=0.5, y=1, w=4, b=1))
print(sample_value)
```

Если Вы решили не считать производную самостоятельно, а используете для нахождения производной библиотеку SymPy, то упростите результат (метод `.simplify()`), иначе правильный ответ может не быть принят.

`-1/(exp(b + w*x) + 1) - y + 1`

$$\frac{-1}{\exp(b+w \cdot x)+1} - y + 1$$

✓ Верный ответ, так держать!

## Вопрос 3

Выполнен

Баллов: 1,00 из 1,00

Теперь добавьте L2-регуляризацию  $Loss(\hat{y}, y) = BCE(\hat{y}, y) + c(w^2 + b^2)$ , где  $c > 0$  - коэффициент регуляризации.

Задание то же, что и на шаге 2: найдите в общем виде производную функции потерь по  $w \frac{\partial Loss(\hat{y}, y)}{\partial w}$  и запишите в ответ её формулу. Для обозначений используйте латинские буквы  $y, x, w, b, c$  в нижнем регистре.

Ответ должен компилироваться в Sympy

```
import sympy.parsing.sympy_parser

sample_expr_str = '<your solution here>'
sample_expr = sympy.parsing.sympy_parser.parse_expr(sample_expr_str)
sample_value = sample_expr.evalf(subs=dict(x=0.5, y=1, w=4, b=1, c=1))
print(sample_value)
```

`x * (1/(1+exp(-w*x-b))-y)+2*c*w`

$$x \cdot \left( \frac{1}{1+\exp((-w) \cdot x - b)} - y \right) + 2 \cdot c \cdot w$$

✓ Верный ответ, так держать!

**Вопрос 4**

Выполнен

Баллов: 1,00 из 1,00

Используя формулу производной с предыдущего шага, запишите формулу для обновления веса  $w$  с помощью [стохастического градиентного спуска](#) (размер минибатча равен 1). Для обозначения скорости обучения (learning rate) используйте маленькую латинскую букву  $t$ .

$$w - t \cdot (x \cdot (1 / (1 + \exp(-w \cdot x - b)) - y) + 2 \cdot c \cdot w)$$

$$w - t \cdot \left( x \cdot \left( \frac{1}{1 + \exp((-w) \cdot x - b)} - y \right) + 2 \cdot c \cdot w \right)$$

✓ Верный ответ, так держать!

**Вопрос 5**

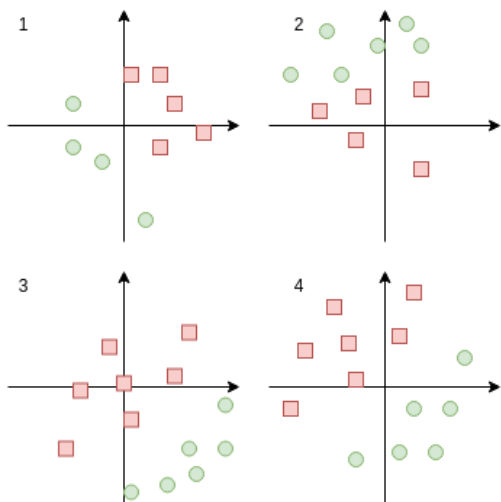
Выполнен

Баллов: 1,00 из 1,00

Мы предлагаем Вам поразмышлять над разницей в мощности двух вариантов логистической регрессии:

- $\hat{y}_1(x) = \sigma(wx + b)$
- $\hat{y}_2(x) = \sigma(wx + b)$ , где  $b = 0$

Выберите варианты ответа (один или несколько), соответствующие датасетам, которые могут быть успешно обработаны  $\hat{y}_1(x)$  и не могут быть обработаны  $\hat{y}_2(x)$ .



Выберите один или несколько ответов:

- ☐ a. 4
- ☒ b. 3
- ☐ c. 1
- ☒ d. 2

Ваш ответ верный.

**Вопрос 6**

Верно

Баллов: 1,00 из 1,00

Напишите функцию для вычисления точечной взаимной информации двух случайных событий.

$$pmi(a, b) = \log \frac{p(a, b)}{p(a)p(b)}$$

На вход функция получает два массива из 0 и 1 одинаковой длины - реализации случайных событий. 1 - событие произошло, 0 - не произошло.

В результате функция должна вернуть вещественное число - точечную взаимную информацию **событий**.

При расчёте следует использовать натуральный логарифм (`numpy.log`).

**Ответ:** (штрафной режим: 0 %)

Сброс ответа

```
1 import sys
2 import numpy as np
3
4
5 def parse_array(s):
6     return np.array([int(s.strip()) for s in s.strip().split(' ')])
7
8 def read_array():
9     return parse_array(sys.stdin.readline())
10
11 def calculate_pmi(a, b):
12     # Calculate joint probability
13     joint_prob = np.sum((a == 1) & (b == 1)) / len(a)
14
15     # Calculate marginal probabilities
16     marginal_prob_a = np.sum(a == 1) / len(a)
17     marginal_prob_b = np.sum(b == 1) / len(b)
18
19     # Calculate PMI
20     pmi = np.log(joint_prob / (marginal_prob_a * marginal_prob_b))
21
22     return pmi
23
24 a = read_array()
25 b = read_array()
26 pmi_value = calculate_pmi(a, b)
27
28 print('{:.6f}'.format(pmi_value))
29
30
```

Прошли все тесты! ✓

Верно

Баллы за эту попытку: 1,00/1,00.

**Вопрос 7**

Выполнен

Баллов: 1,00 из 1,00

Найдите количество слов, которые встречаются менее, чем в 10 из 10000 документов, если предполагать, что вероятность встретить слово в документе распределена по Ципфу с параметром  $s = 2$ , количество слов в словаре  $N = 1000$ . Ранги нумеруются с 1.

Ответ: 

Ваш ответ верный.

[◀ 2.2 Создаём нейросеть для работы с текстом](#)[2.4 Семинар: классификация новостных текстов ▶](#)