

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



Системи штучного інтелекту

Логістична регресія

Практична робота #3

Виконав:
Ростислав Коваль
Група:
ІО-11мн
Курс:
1

31 жовтня 2021 р.

1 Логістична регресія

1.1 Письмове завдання

Покажіть, що похідна сигмоїди дорівнює цьому виразу:

$$\frac{d\hat{y}}{dx} = \frac{d\sigma(z)}{dx} = \sigma(x)(1 - \sigma(x)), \quad (1)$$

$$\text{де } \hat{y} = \sigma(x) = \frac{1}{1 + \exp(-x)}$$

Розв'язок:

$$\begin{aligned} \frac{d}{dx} \sigma(x) &= \frac{d}{dx} \left[\frac{1}{1 + e^{-x}} \right] \\ &= \frac{d}{dx} (1 + e^{-x})^{-1} \\ &= -(1 + e^{-x})^{-2} (-e^{-x}) \\ &= \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\ &= \frac{1}{1 + e^{-x}} \cdot \frac{(1 + e^{-x}) - 1}{1 + e^{-x}} \\ &= \frac{1}{1 + e^{-x}} \cdot \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \right) \\ &= \frac{1}{1 + e^{-x}} \cdot \left(1 - \frac{1}{1 + e^{-x}} \right) \\ &= \sigma(x) \cdot (1 - \sigma(x)) \end{aligned}$$

1.2 Завдання з програмування

Лістинг 1: Ініціалізація вагів та зсуву

```
1 def parameters_inititalization():
2     W = np.zeros((50310, 1))
3     b = 0
4     return W, b
```

Лістинг 2: Застосування нелінійної функції активації (сигмоїди) до лінійної комбінації вхідних ознак та ваг, включаючи зсув

```
1 def forwardPropagate(X, W, b):
2     z = np.dot(W.T, X) + b
3     y_hat = 1 / (1 + np.exp(-z))
4     return z, y_hat
```

Лістинг 3: Обчислення усередненої втрати на всьому навчальному наборі даних. Цільова функція

```

1 def cost(n, y_hat, y_true):
2     mult = np.multiply(y_true, np.log(y_hat)) + np.multiply((1 - y_true), np.log(1 - y_hat))
3     J = - np.sum(mult) / n
4     return J

```

Лістинг 4: Розрахунок градієнтів цільвої функції відносно ваг та зсуву

```

1 def backwardPropagate(n, X, y_hat, y_true):
2     dW = np.dot(X, (y_hat - y_true).T) / n
3     db = np.sum((y_hat - y_true)) / n
4     return dW, dbE

```

Лістинг 5: Оновлення вагів та зсувів

```

1 def update(alpha, dW, db, W, b):
2     W = W - alpha * dW
3     b = b - alpha * db
4     return W, b

```

Лістинг 6: Тестування

```

1 def testing():
2     W, b = parameters_inititalization()
3     for i in range(100):
4         z, y_hat = forwardPropagate(norm, W, b)
5         J = cost(1, y_hat, y_true)
6         dW, db = backwardPropagate(1, norm, y_hat, y_true)
7         W, b = update(0.003, dW, db, W, b)
8         print(J)
9 testing()

```

1.3 Результати експериментів

Деталі дослідження представлені у вигляді таблиць. Для $\alpha = 0.0001$:

iteration	J	b
0	0.6931471805599453	-5e-05
10	0.45559604431008077	-8.659300784757073e-05
20	0.32638987622128257	-0.00011444062508716522
30	0.25010896030122004	-0.00013656903215452325
40	0.20114230925923618	-0.00015478942782218698
50	0.16750704975587505	-0.00017021236234306852
60	0.14315815066741297	-0.00018355066214307788
70	0.12479714901307239	-0.00019528306853205837
80	0.11049714892926266	-0.0002057441802317694
90	0.09906654554958352	-0.0002151759365435022
100	0.08973278237094713	-0.0002237583929006694

Для $\alpha = 0.003$:

iteration	J	b
0	0.6931471805599453	-0.0015
10	6.883807061498679e-08	-0.0015000002065142049
20	6.883791440659648e-08	-0.0015000004130279409
30	6.883775808718385e-08	-0.0015000006195412079
40	6.883760176777122e-08	-0.001500000826054006
50	6.88374454483586e-08	-0.0015000010325663354
60	6.883728912894597e-08	-0.0015000012390781958
70	6.883713292055564e-08	-0.0015000014455895873
80	6.883697660114302e-08	-0.00150000165210051
90	6.883682028173039e-08	-0.0015000018586109637
100	6.883666396231776e-08	-0.0015000020651209486

З таблиці можна зробити висновок, що при більшій швидкості навчання значення цільової функції з кожною ітерацією стає значно меншим, що свідчить про здійснення навчання. Значення зсуву є більшим при $\alpha = 0.003$. Значення вагів представляються у вигляді великого масиву, тому недоречно вставляти його в таблицю порівняння. Їх значення змінюються при кожній ітерації.

1.4 Допомога

Не отримував допомоги від інших людей, використовував наступні матеріали: [1, 2, 3, 4].

1.5 Висновки

Упродовж виконання лабораторної роботи було спостережно, що значення цільової функції зменшується при кожній ітерації, при збільшенні швидкості навчання значення функції значно зменшується. Значення зсуву моделі незначно зменшується, можна вважати, значення вагів також змінюється.

Література

- [1] overleaf. Inserting images. overleaf. [Online]. Available: https://ru.overleaf.com/learn/latex/Inserting_Images
- [2] A. C. Quiros. Deep-learning-specialization-coursera. github. [Online]. Available: https://github.com/AdalbertoCq/Deep-Learning-Specialization-Coursera/blob/68b4b53ae2b88308e048558c49fa2192aa32fcd6/Neural%20Networks%20and%20Deep%20Learning/week2/logistic_regression.py
- [3] Kulbear. deep-learning-coursera. github. [Online]. Available: <https://github.com/Kulbear/deep-learning-coursera/blob/master/Neural%20Networks%20and%20Deep%20Learning/Logistic%20Regression%20with%20a%20Neural%20Network%20mindset.ipynb>
- [4] M. Percy. Derivative of sigmoid function. math.stackexchange.com. [Online]. Available: <https://math.stackexchange.com/questions/78575/derivative-of-sigmoid-function-sigma-x-frac11e-x>