COURSE NAME: Hillel - Machine Learning. Lesson 3. (06.03.2023 19:15)
LECTURER: Yevhen Krasnokutsky
STUDENT: Maryna Lysenko

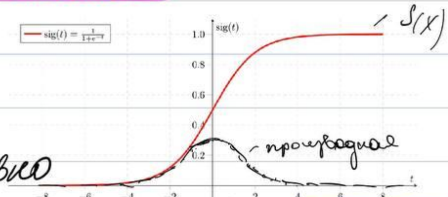
HW 3 REPORT “Computation Graph. Derivative, gradient, learning rate, gradient descent”

Завдання 1.

Note 6 Mar 2023
6 Mar 2023, 21:33

$S(x)$ – Sigmoid function

$$S(x) = \frac{1}{1+e^{-x}}$$



e = Euler's number

Сигмоїда - неперервно диференційована монотонно ↑, нелінійна S-подібна ф-ція; логістична крива

- Вона обмежена горизонтальними асимптотами $y=1$, $y=0$ дає (нормалізац)
- Характерний гладкий градієнт
- Для $x>2$ та $x<-2$ „приближається до однієї з асимптот

Task 1. Take the analytical derivative of $S(x)$

$$y = S(x) = \frac{1}{1+e^{-x}}$$

$$y' = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{d}{dx}\left(1+e^{-x}\right)^{-1} =$$

$$\left[\frac{1}{u(x)}\right]' = u(x)^{-1\,\prime} = -\frac{u'(x)}{u(x)^2} = -u(x)^{-2}\cdot u'(x)$$

$$= -\frac{(1+e^{-x})'}{(1+e^{-x})^2} = -\frac{\overset{=0}{1'}+(e^{-x})'}{(1+e^{-x})^2} = -\frac{e^{-x}\cdot(-x)'}{(1+e^{-x})^2} = \frac{e^{-x}\cdot x'}{(1+e^{-x})^2} =$$

$$= \boxed{\frac{e^{-x}}{(1+e^{-x})^2}} = \underbrace{\frac{1}{1+e^{-x}}}_{=S(x)} \cdot \frac{e^{-x}+1-1}{1+e^{-x}} =$$

$$= S(x)\cdot\left(\cancel{\frac{e^{-x}+1}{1+e^{-x}}} - \underbrace{\frac{1}{1+e^{-x}}}_{=S(x)}\right) = \boxed{S(x)\cdot(1-S(x))}$$

Завдання 2. Experiments with demo code (gradient descent)

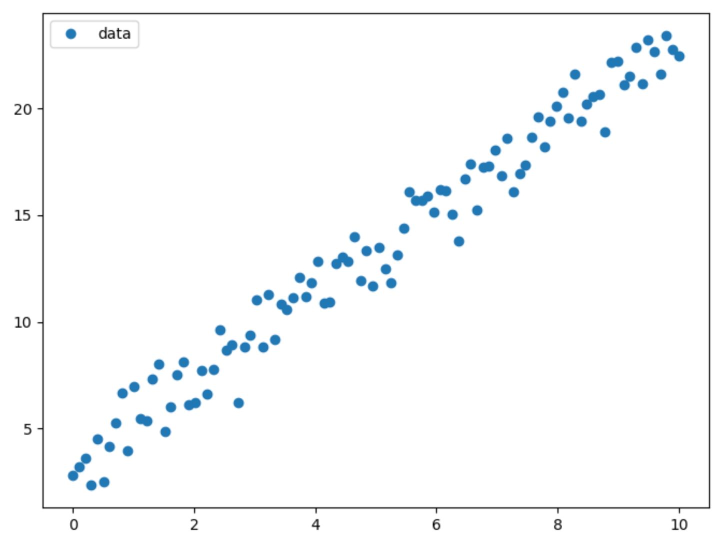
- Vary learning rate
- Vary epochs
- Plot MSE over training (over epochs for specific learning rate)

Introduction

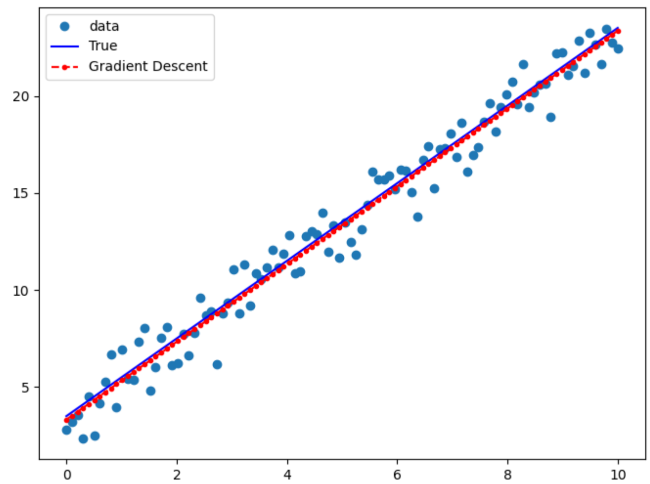
Ця лабораторна робота включає в себе розробку простої лінійної регресії методом градієнтного спуску.

Для експериментів було згенеровано випадкові дані, що відповідають лінійній залежності між змінною X та змінною Y з деяким додатковим шумом. Конкретніше, змінна X є рівномірно розподіленою на відрізку від 0 до 10 з 100 рівномірно розподіленими значеннями. Змінна Y визначається формулою $y = 2 * x + 3.5$ з додаванням гаусового шуму. Шум було згенеровано за допомогою випадкової величини з нормальним розподілом з середнім значенням рівним 0 та стандартним відхиленням рівним 1.

Отже, експериментальні дані відповідають лінійній залежності з невеликим додатковим шумом, що робить їх придатними для застосування методів лінійної регресії.



Далі застосуємо метод градієнтного спуску (learning rate = `0.01,` epochs = 1000): для побудови моделі лінійної регресії. до набору даних, який ми згенерували раніше.



У кінці виведемо на екран параметри отриманої моделі лінійної регресії

Params:

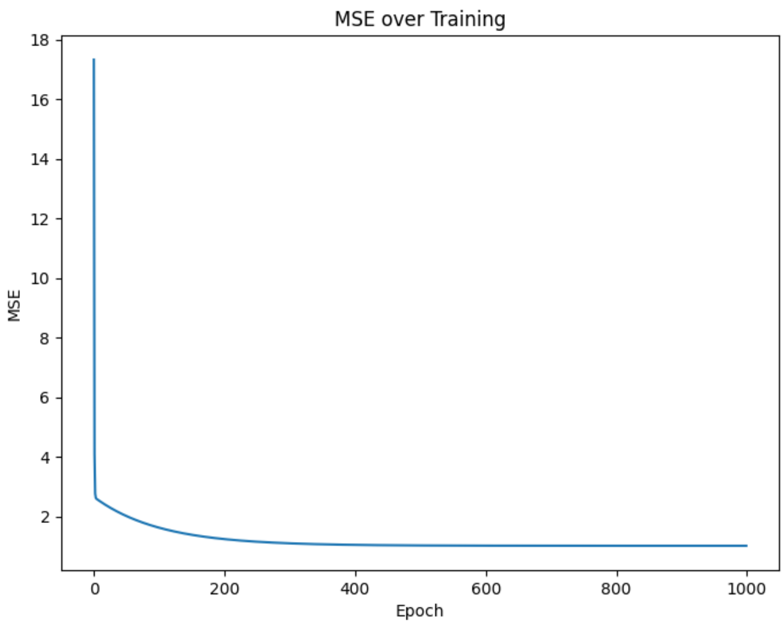
m = 2.003222509810484

b = 3.3248920388471914

Final MSE: 1.0241824830839787

Виведемо динаміку значень MSE (Mean Squared Error), які були обчислені на кожній епосі під час тренування лінійної регресії методом градієнтного спуску для можливості оцінити, наскільки точно модель навчається на даних і які гіперпараметри (наприклад, learning rate) працюють найкраще. (Чим нижче значення MSE, тим краще модель підлаштовується під даний набір даних, а отже, тим краще модель може передбачити відповіді на нових даних. В данній лабораторній роботі ми використовуємо MSE для візуалізації процесу навчання та знаходження оптимальних гіперпараметрів для моделі лінійної регресії.)

(learning rate = `0.01,` epochs = 1000):



Десь з 400 епохи MSE змінюється мінімально.

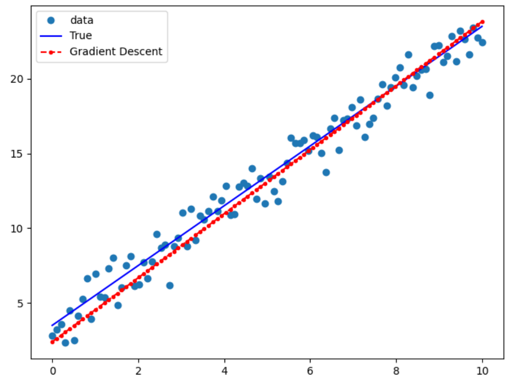
Results

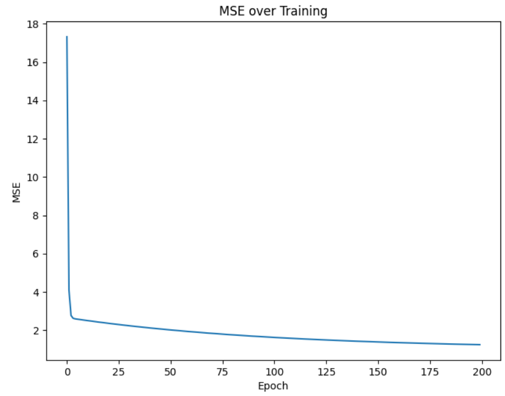
y = a * x + b, a = 2, b = 3.5

Експеримент 0: learning rate = `0.01,` epochs = 1000, Params: m = 2.003222509810484, b = 3.3248920388471914, Final MSE: 1.0241824830839787, Десь з 400 епохи MSE змінюється вже не так швидко. Початкові параметри моделі y = 2x + 3.5 були досить точно відновлені під час використання цих параметрів

Експеримент 1: Залишаю параметри з експеримента 0, але зменшую кількість епох до 200.

Params: m = 2.142493733590308. b = 2.3986413449255664 Final MSE: 1.250220311528887 .





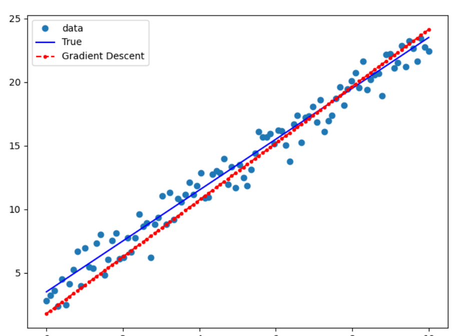
При зменшенні кількості епох до 200 модель не встигла зійтися і вийшла найгірша якість передбачення.

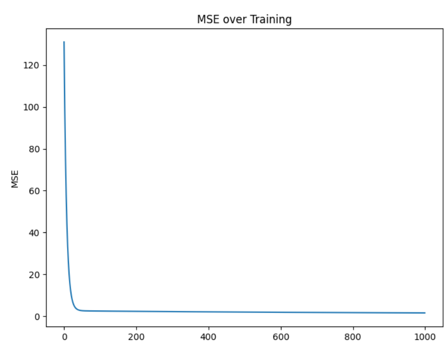
Експеримент 2: У експерименті 2 було повернуто до початкових умов, але , але learning rate = 0.001. Результаты Params:

m = 2.234267465666658

b = 1.7882820710393608

Final MSE: 1.6372142601852175





Модель недотренована.

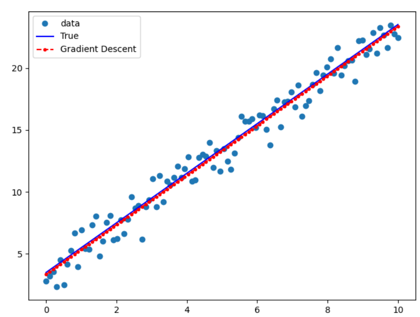
Експеримент 3: У експерименті 3 було повернуто до початкових умов, але збільшено кількість епох до 2000.

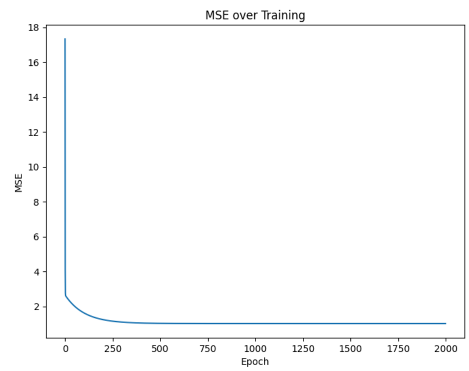
Params:

m = 2.000591763571137

b = 3.3423883346618903

Final MSE: 1.0241037070031567





Отриманий результат був незначно точнішим, ніж найкращий результат попередніх експериментів.

Експеримент 4: У експерименті 4 з learning rate = 0.1 код побудови моделі викинув помилку. Помилка виникла через переповнення (overflow) при обчисленні середньоквадратичної помилки (mean squared error). Також була попереджена некоректна операція множення зі значеннями NaN (not a number).

Щоб вирішити цю проблему, можна спробувати зменшити значення learning rate (наприклад, lr=0.01) або зменшити кількість епох для тренування моделі.

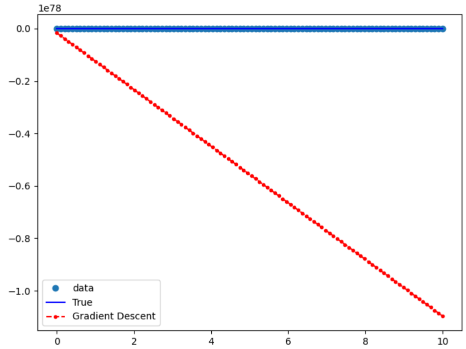
Експеримент 5: с learning rate = 0.1 и уменьшаю количество эпох до 100.

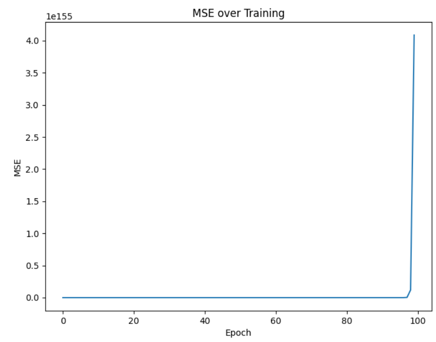
Params:

m = -1.0801217418175042e+77

b = -1.624073026895325e+76

Final MSE: 4.0865746532122445e+155





При використанні learning rate = 0.1 та зменшенні кількості епох до 100 отримано найгіршу модель з вкрай низькою якістю передбачення.

Conclusions

Порівнюючи експерименти, виявляється, що при настроюванні гіперпараметрів для лінійної регресії необхідно знаходити баланс між швидкістю навчання моделі та її точністю. В результаті експериментів було показано, що оптимальний learning rate для цього завдання дорівнює 0.01, а кількість епох тренування має бути достатньою для збіжності моделі.

Завдання 3.

Task 3. Make one forward and backward steps for $L = (2a+b)(c-d)$

a, b, c, d are arbitrary numbers

Forward steps:

7 $a = 3, \quad b = 4, \quad c = 5, \; d = 6$

$L = (2 \cdot 3 + 4)(5-6) = -10$

Backward steps:

$$\frac{\partial L}{\partial a} = (c-d)(2) = 2(c-d)$$

$$\frac{\partial L}{\partial b} = (c-d) \cdot (1) = c-d$$

$$\frac{\partial L}{\partial c} = (2a+b) \cdot 1 = 2a+b$$

$$\frac{\partial L}{\partial d} = (2a+b) \cdot (-1) = -(2a+b)$$