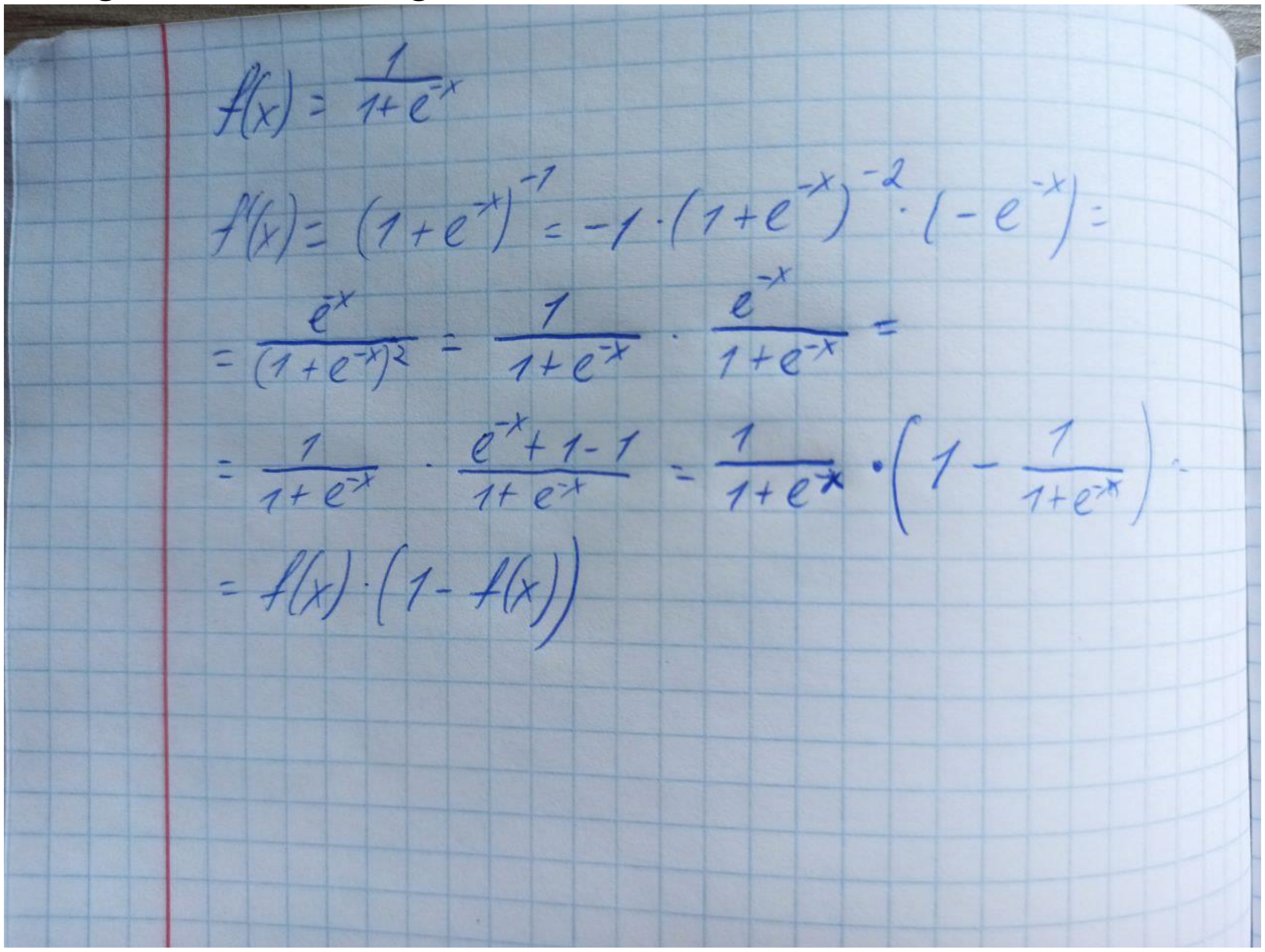
Report on the completion of the third task.

1. **The purpose of the task.**
   The main goal of the project is to take analytical derivative of sigmoid function; investigate how do the different learning rates and epochs affect the performance of gradient descent; make one forward and backward steps for L = (2a+b)(c-d)

2. Taking the derivative of sigmoid function.



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

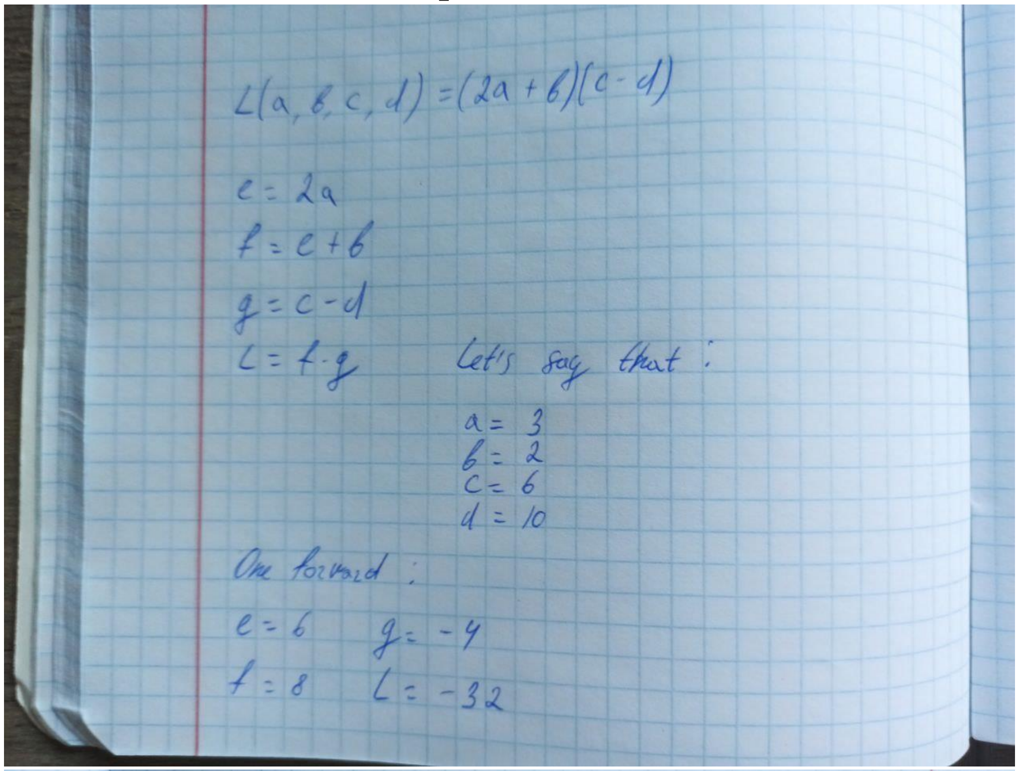
$$f'(x) = (1+e^{-x})^{-1} = -1 \cdot (1+e^{-x})^{-2} \cdot (-e^{-x}) =$$

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

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}+1-1}{1+e^{-x}} = \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) =$$

$$= f(x) \cdot (1 - f(x))$$

3. Forward and backward steps for L function.



$L(a, b, c, d) = (2a + b)(c - d)$

$e = 2a$

$f = e + b$

$g = c - d$

$L = f \cdot g$

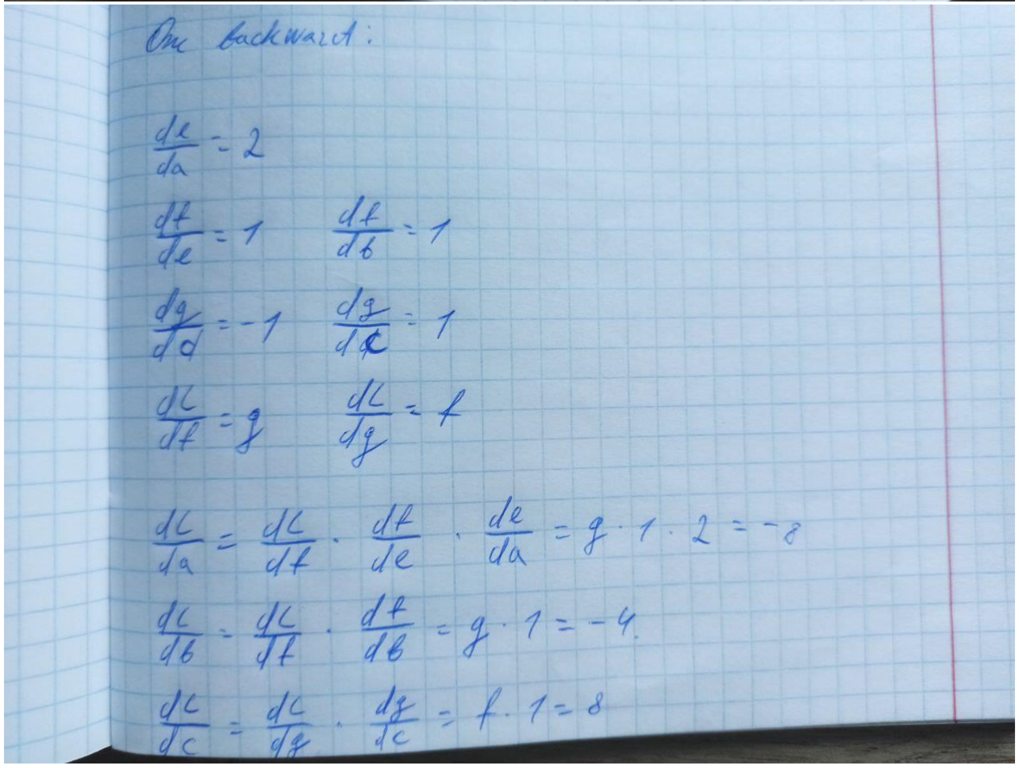
Let's say that:

$a = 3$
$b = 2$
$c = 6$
$d = 10$

On forward:

$e = 6$ $\quad g = -4$

$f = 8$ $\quad L = -32$



On backward:

$\frac{de}{da} = 2$

$\frac{df}{de} = 1 \quad \frac{df}{db} = 1$

$\frac{dg}{dd} = -1 \quad \frac{dg}{dc} = 1$

$\frac{dL}{df} = g \quad \frac{dL}{dg} = f$

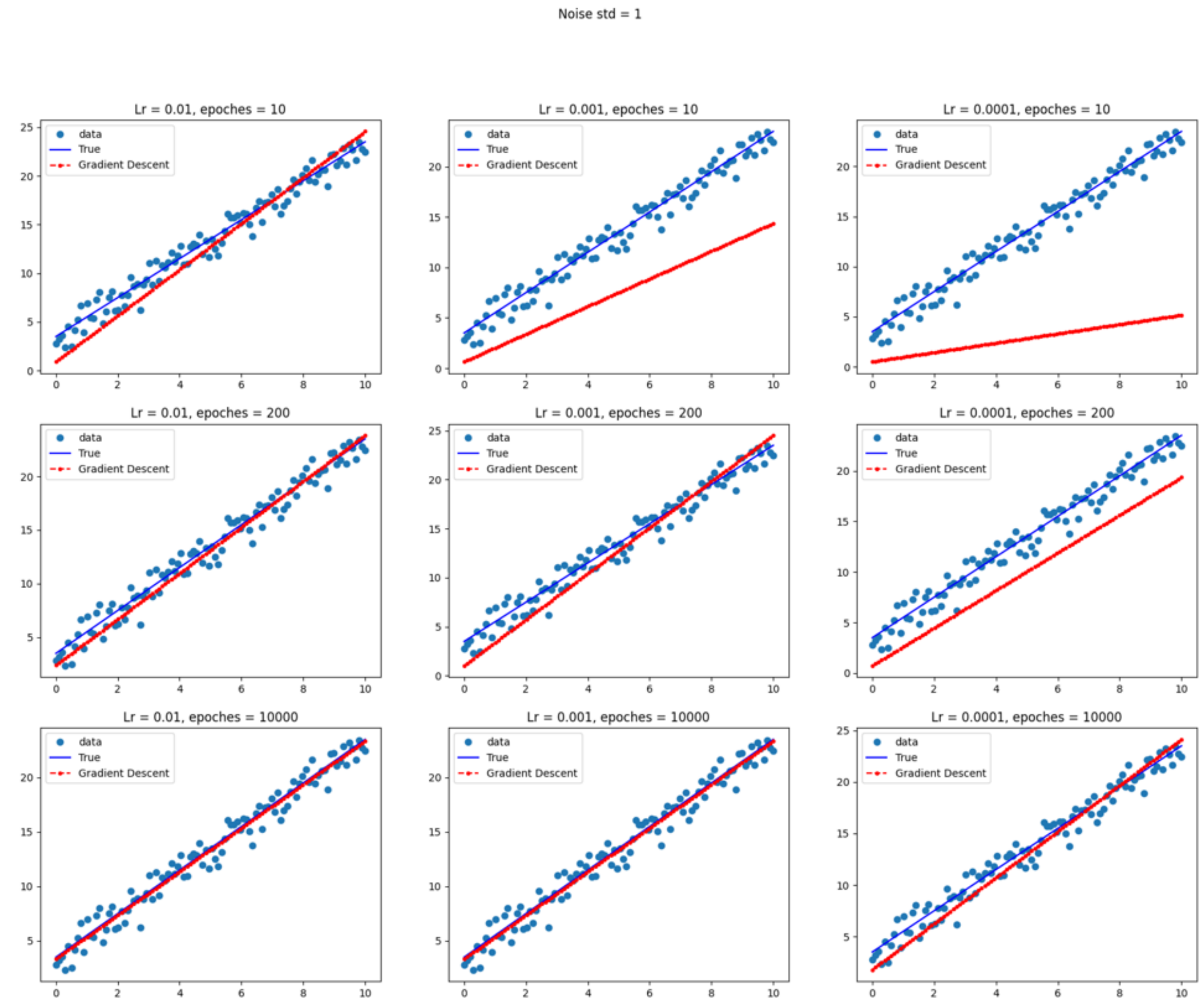
$\frac{dL}{da} = \frac{dL}{df} \cdot \frac{df}{de} \cdot \frac{de}{da} = g \cdot 1 \cdot 2 = -8$

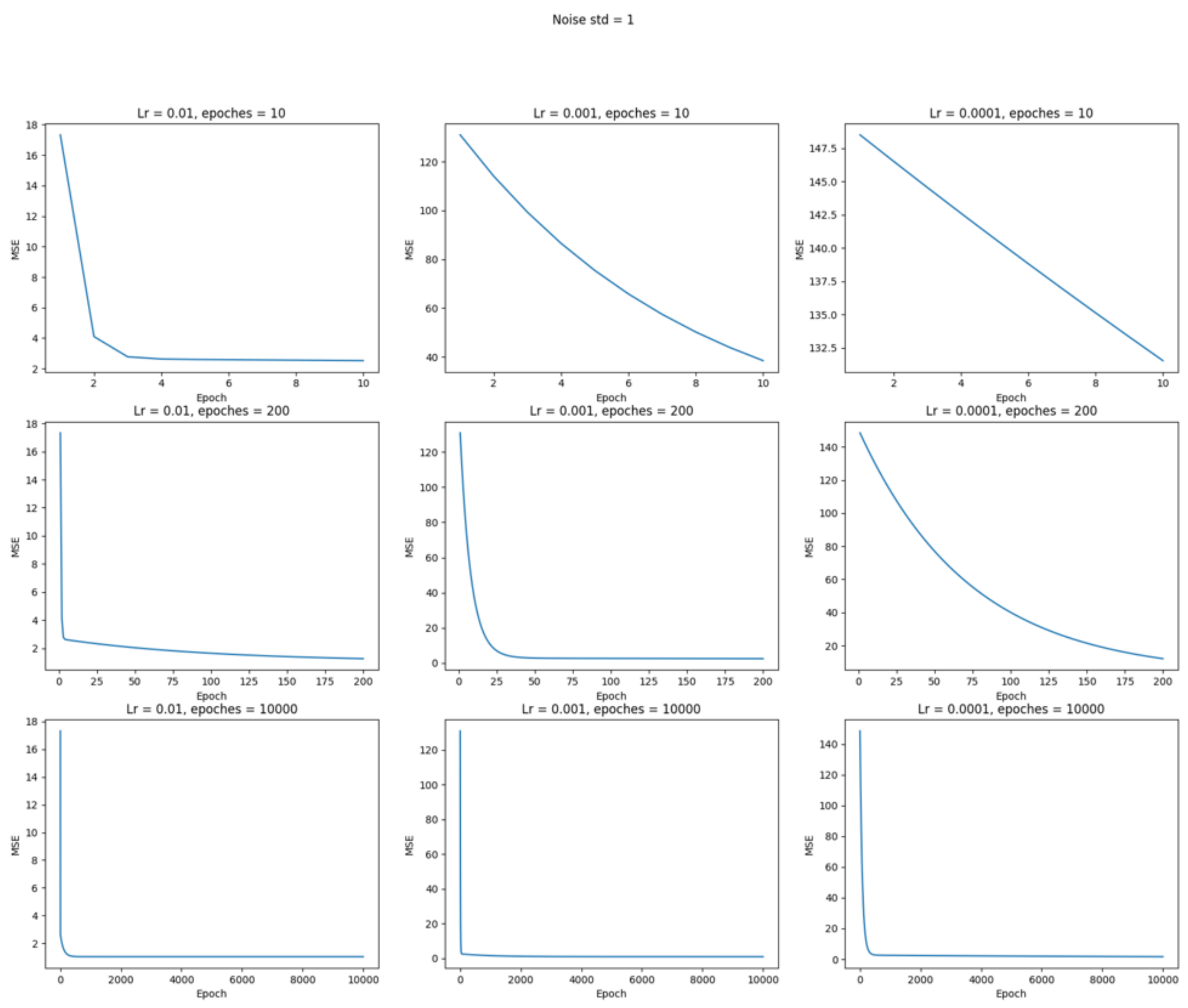
$\frac{dL}{db} = \frac{dL}{df} \cdot \frac{df}{db} = g \cdot 1 = -4.$

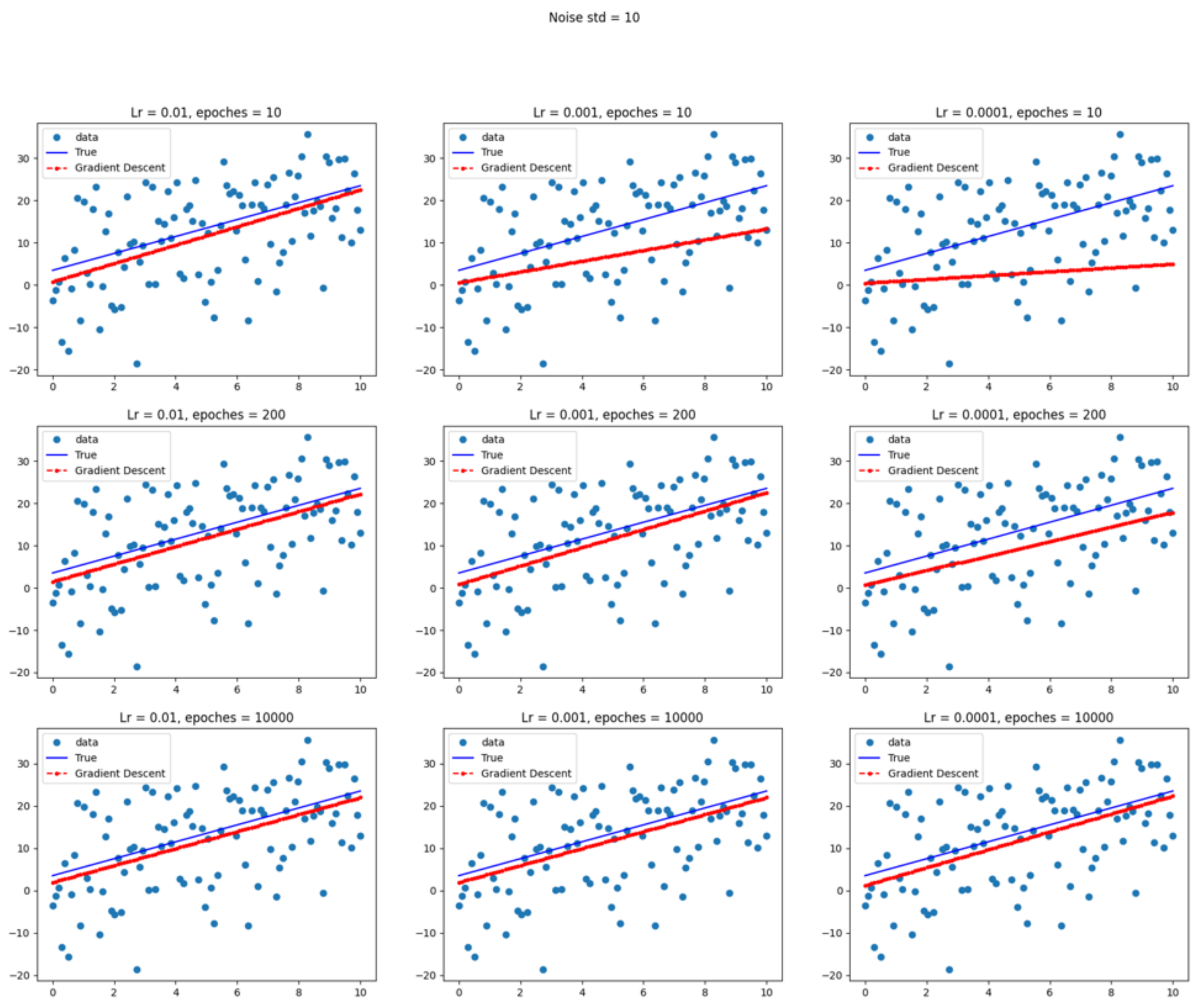
$\frac{dL}{dc} = \frac{dL}{dg} \cdot \frac{dg}{dc} = f \cdot 1 = 8$

$$\frac{dc}{dd} = \frac{dc}{dq} \cdot \frac{dq}{dd} = f \cdot (-1) = -8.$$

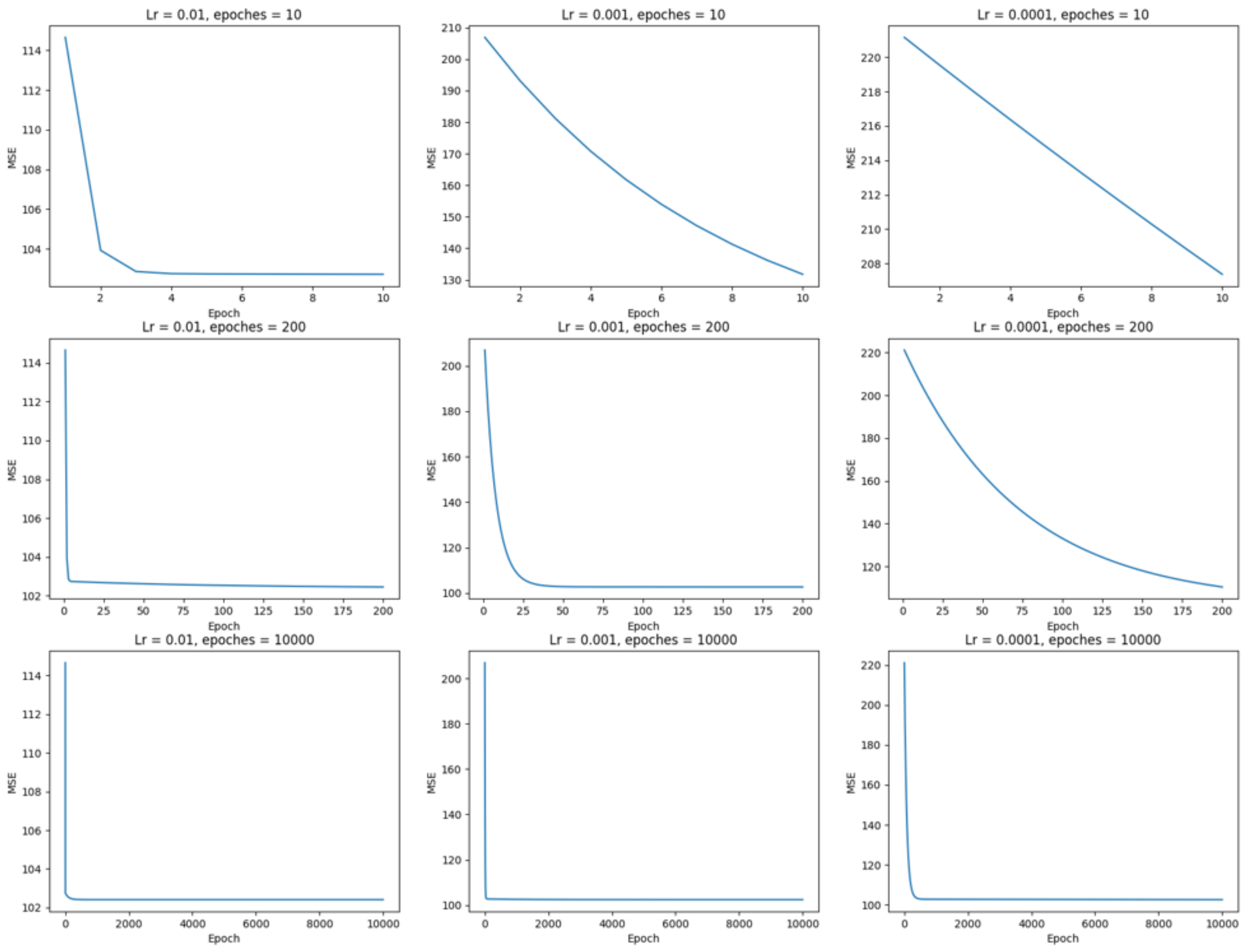
4. Plotting the graphs of model performance and MSE to epochs.




Noise std = 1
Lr = 0.01, epoches = 10
Lr = 0.001, epoches = 10
Lr = 0.0001, epoches = 10
Lr = 0.01, epoches = 200
Lr = 0.001, epoches = 200
Lr = 0.0001, epoches = 200
Lr = 0.01, epoches = 10000
Lr = 0.001, epoches = 10000
Lr = 0.0001, epoches = 10000
MSE
Epoch

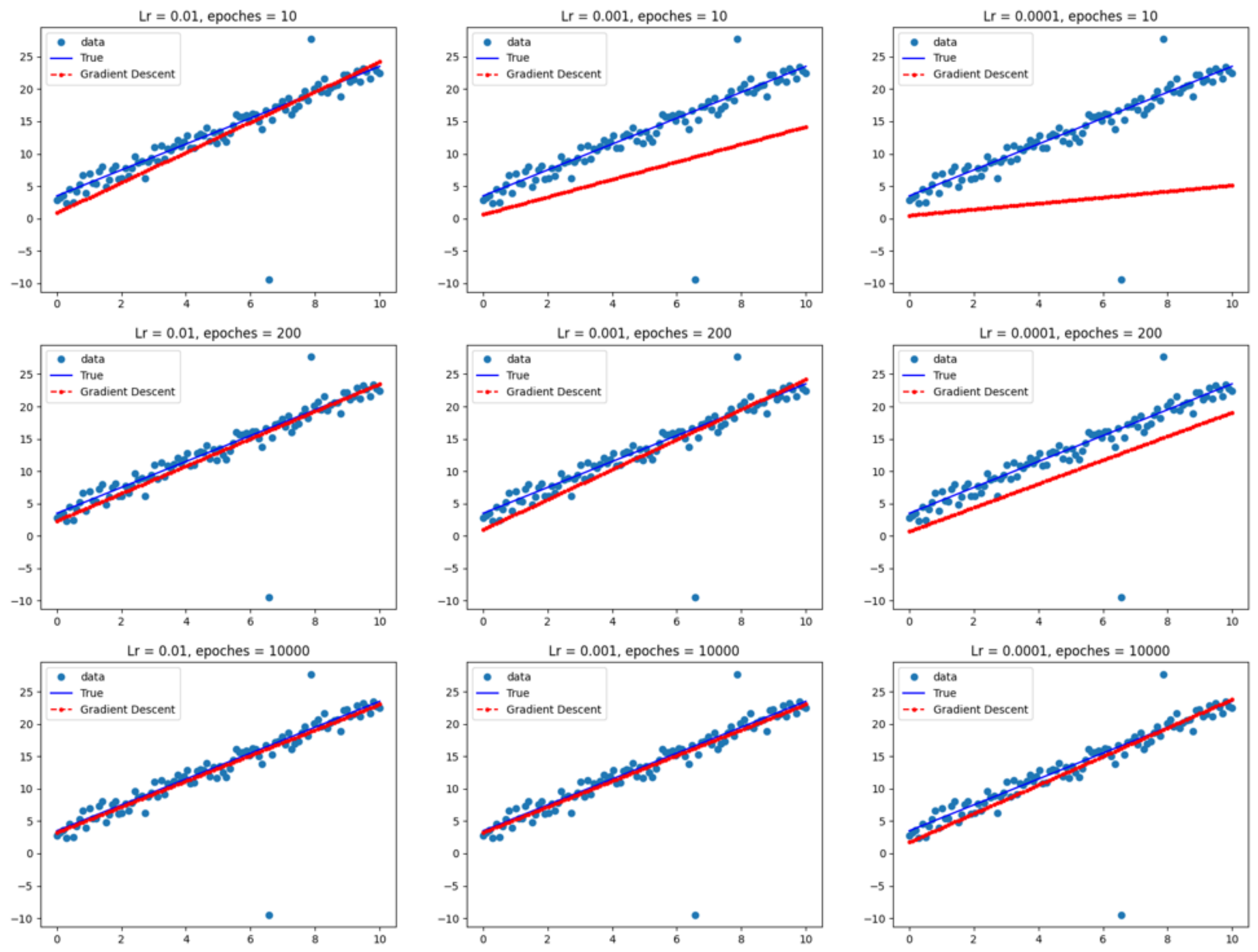

Noise std = 10
Lr = 0.01, epoches = 10
Lr = 0.001, epoches = 10
Lr = 0.0001, epoches = 10
Lr = 0.01, epoches = 200
Lr = 0.001, epoches = 200
Lr = 0.0001, epoches = 200
Lr = 0.01, epoches = 10000
Lr = 0.001, epoches = 10000
Lr = 0.0001, epoches = 10000
data
True
Gradient Descent
30
20
10
0
−10
−20
0
2
4
6
8
10

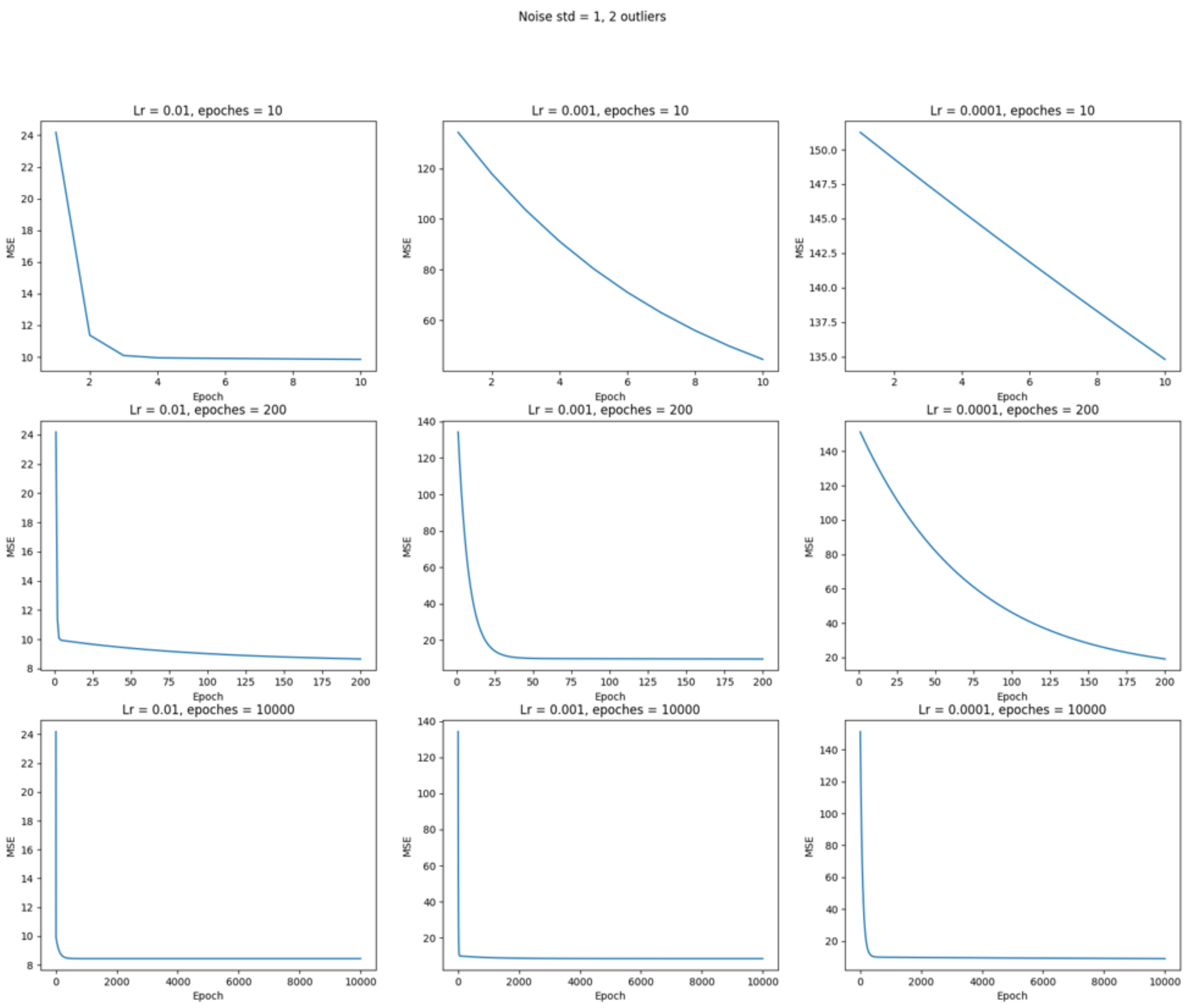
Noise std = 10



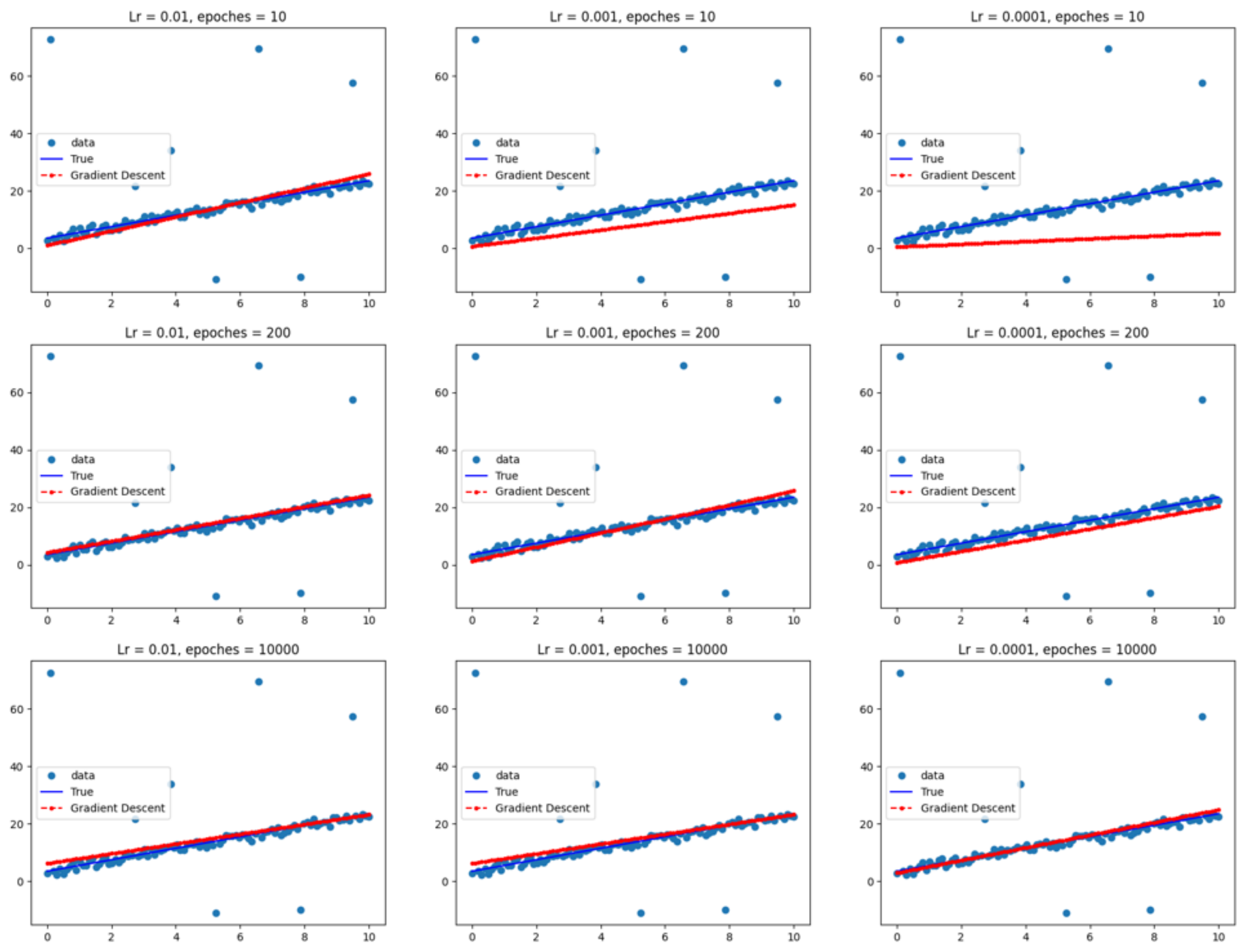
Lr = 0.01, epoches = 10
Lr = 0.001, epoches = 10
Lr = 0.0001, epoches = 10
Lr = 0.01, epoches = 200
Lr = 0.001, epoches = 200
Lr = 0.0001, epoches = 200
Lr = 0.01, epoches = 10000
Lr = 0.001, epoches = 10000
Lr = 0.0001, epoches = 10000
MSE
Epoch

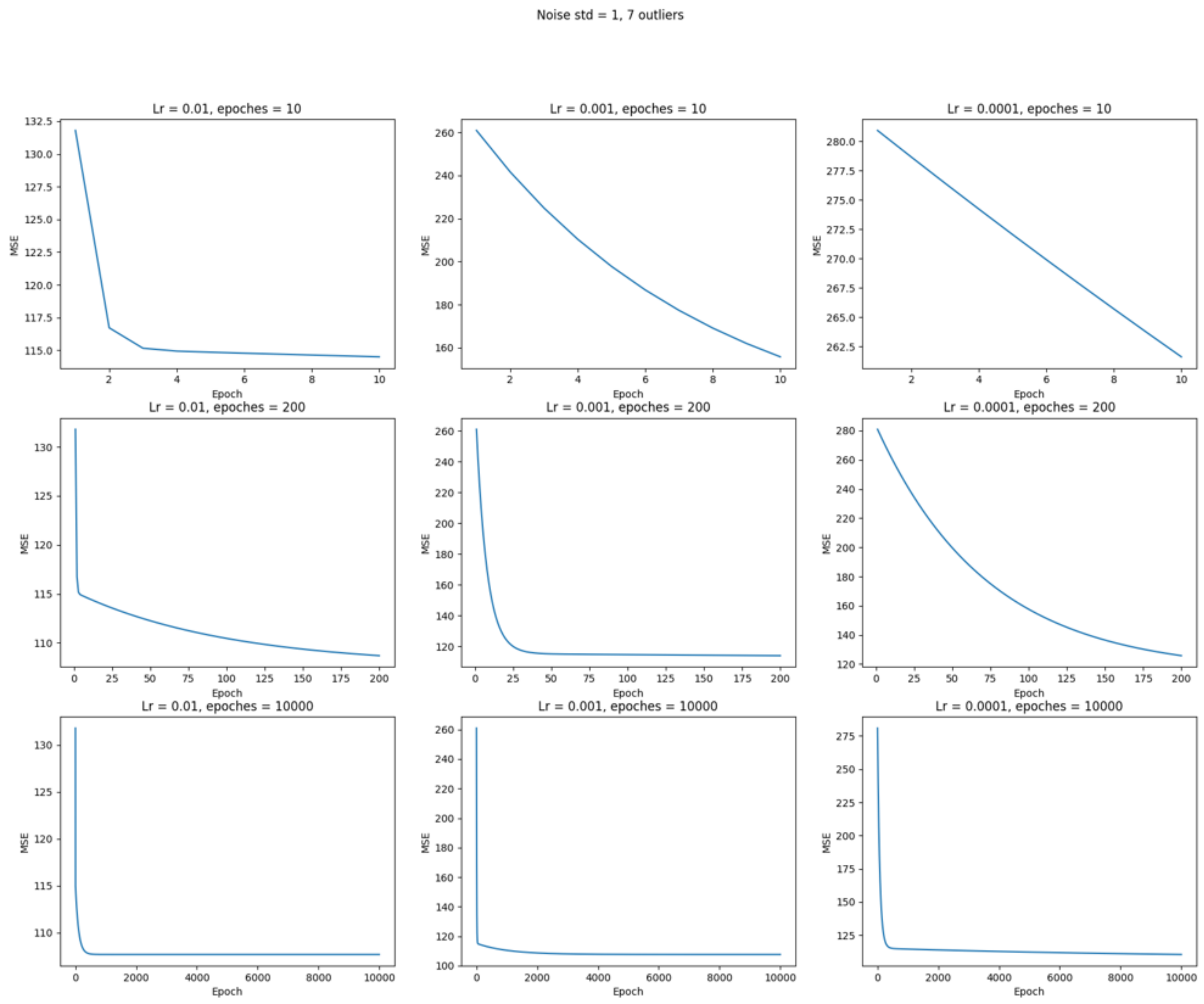
Noise std = 1, 2 outliers




Noise std = 1, 2 outliers
Lr = 0.01, epoches = 10
Lr = 0.001, epoches = 10
Lr = 0.0001, epoches = 10
Lr = 0.01, epoches = 200
Lr = 0.001, epoches = 200
Lr = 0.0001, epoches = 200
Lr = 0.01, epoches = 10000
Lr = 0.001, epoches = 10000
Lr = 0.0001, epoches = 10000
MSE
Epoch

Noise std = 1, 7 outliers





5. Conclusion.

- From the first experiment we can conclude that taking the beautiful data without big noise, learning rates of 0.01, 0.001 give the best coefficients for the large number of epochs. The smallest learning rate of 0.0001 finds the good coefficients but not the best in comparison with others.
- The big noise in the initial data makes the gradient descent find pretty good and identical parameters for all of three learning rates, but this happens only with a big amount of epochs. The fewer epochs we have, and the smaller the learning rate is, the worse the gradient's performance is.
- Also worth adding that the learning rate of 0.01 finds the best coefficients faster than others, regardless of outliers, noise and epochs.
- Only one conclusion can be drawn about outliers: the more epochs you have – the less outliers affect your performance.