**фМИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по практической работе**

по дисциплине «Введение в информационные технологии» на тему:

«Исследование метода обратного распространения

ошибки для обучения нейронной сети»

Выполнила: студентка группы БПИ2401 Рябова Екатерина

Москва

2025

1. **Тема работы:**

Исследование метода обратного распространения ошибки для обучения нейронной сети

1. **Цель работы:**

В этом задании нужно будет обучить простую нейронную сеть (состоящую из одного нейрона), которая будет решать с некоторой точностью задачу линейной регрессии.

1. **Ход работы:**

|  |  |
| --- | --- |
| **°C** | **°F** |
| **0** | **32** |
| **1** | **33,8** |
| **10** | **50** |
| **15** | **59** |
| **22** | **71,6** |

*В левом столбце указано значение температуры в градусах Цельсия, а в правом столбце — в градусах Фаренгейта.*

*Известно, что все указанные в таблице точки лежат на одной прямой.*

*Нужно найти правило, по которому можно будет переводить градусы Цельсия в градусы Фаренгейта (с некоторой точностью).*

Ожидаемый ответ несложно получить, представив °С как k \* °F + b. Тогда решением является k = 1.8 и b = 32.

Переобучение в данном случае невозможно, поэтому за обучающие данные возьмем две строки таблицы (Так как регрессия линейная и прямая строится по двум точкам), а за тестовые – остальные три.

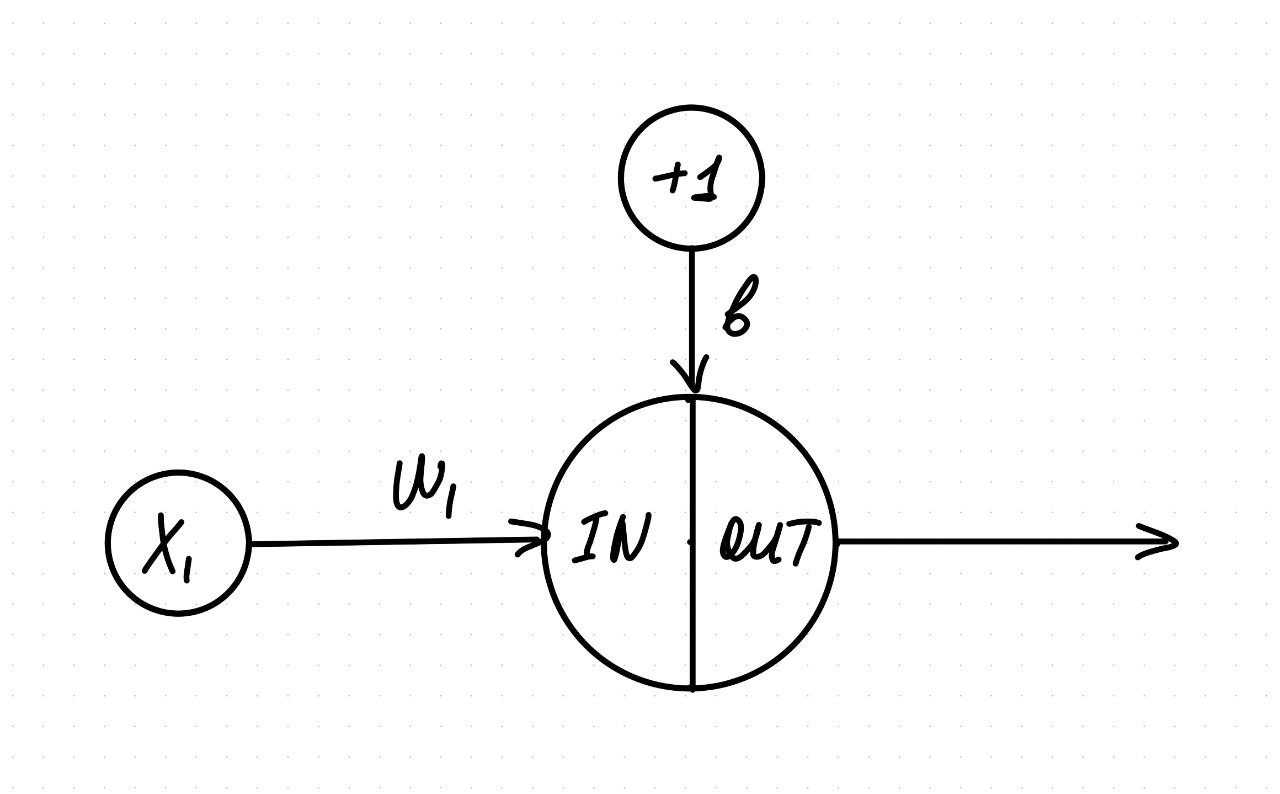
|  |  |  |
| --- | --- | --- |
| **обучение** | |  |
| **X (°C)** | **Y (°F)** | |
| **1** | **33,8** | |
| **15** | **59** | |

|  |  |  |
| --- | --- | --- |
| **тест** | |  |
| **X (°C)** | **Y (°F)** | |
| **0** | **32** | |
| **10** | **50** | |
| **22** | **71,6** | |

В тестовых данных Y это ожидаемый результат.

Значение скорости обучения (learning rate), η возьмем за 0.01

Заметим, что для задачи линейной регрессии нам будет достаточно одного нейрона с линейной функцией активации f(s) = s, и s, взвешенной суммой, равной x1 \* w1 + b, где w1 – коэффициент k, вес, x1 – входное значение, а b – коэффициент смещения.



Одна эпоха обучения будет состоять из следующих шагов:

--- Начало 1 эпохи ---

--- Обучение на 1 обучающем примере ---

1. Прямой проход для 1 примера

2. Обратный проход для 1 примера

3. Обновление весов по результатам обратного прохода

--- Конец обучения на 1 примере ---

--- Обучение на 2 обучающем примере ---

1. Прямой проход для 2 примера

2. Обратный проход для 2 примера

3. Обновление весов по результатам обратного прохода

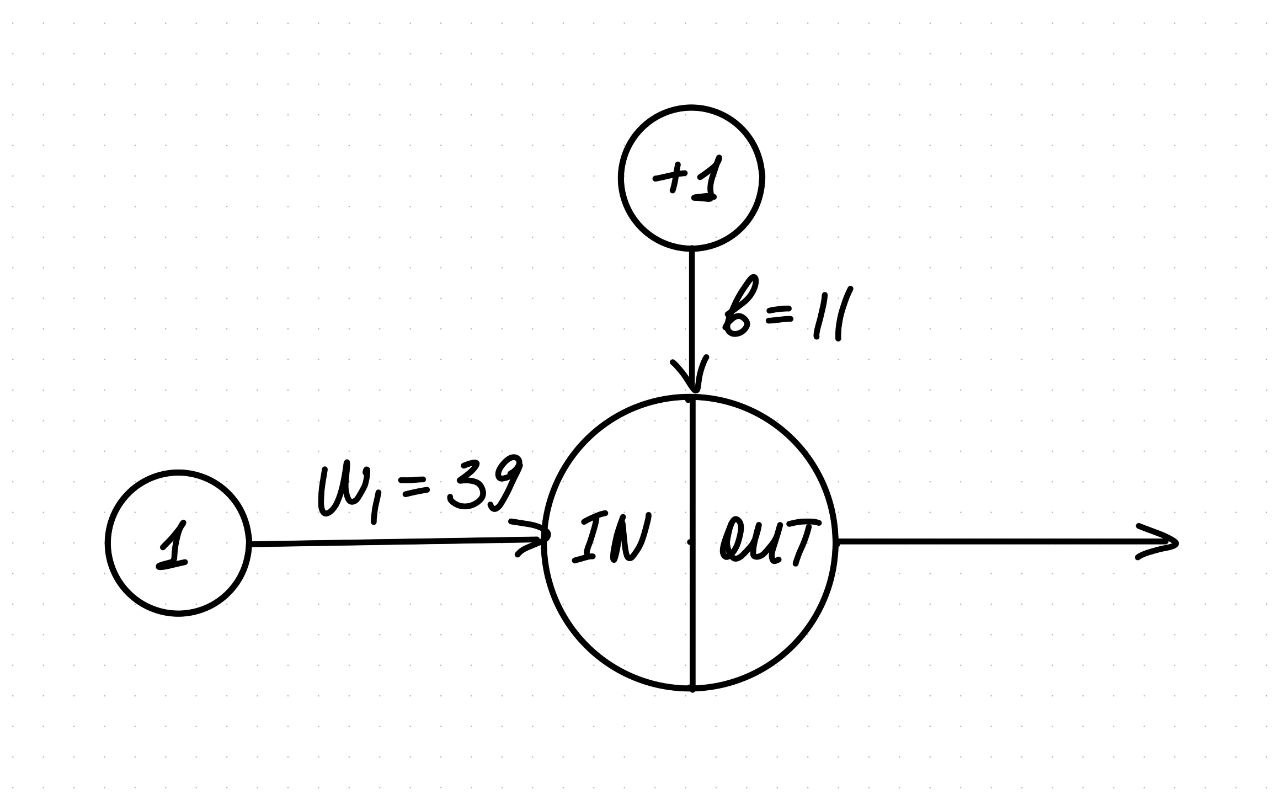
--- Конец обучения на 2 примере ---

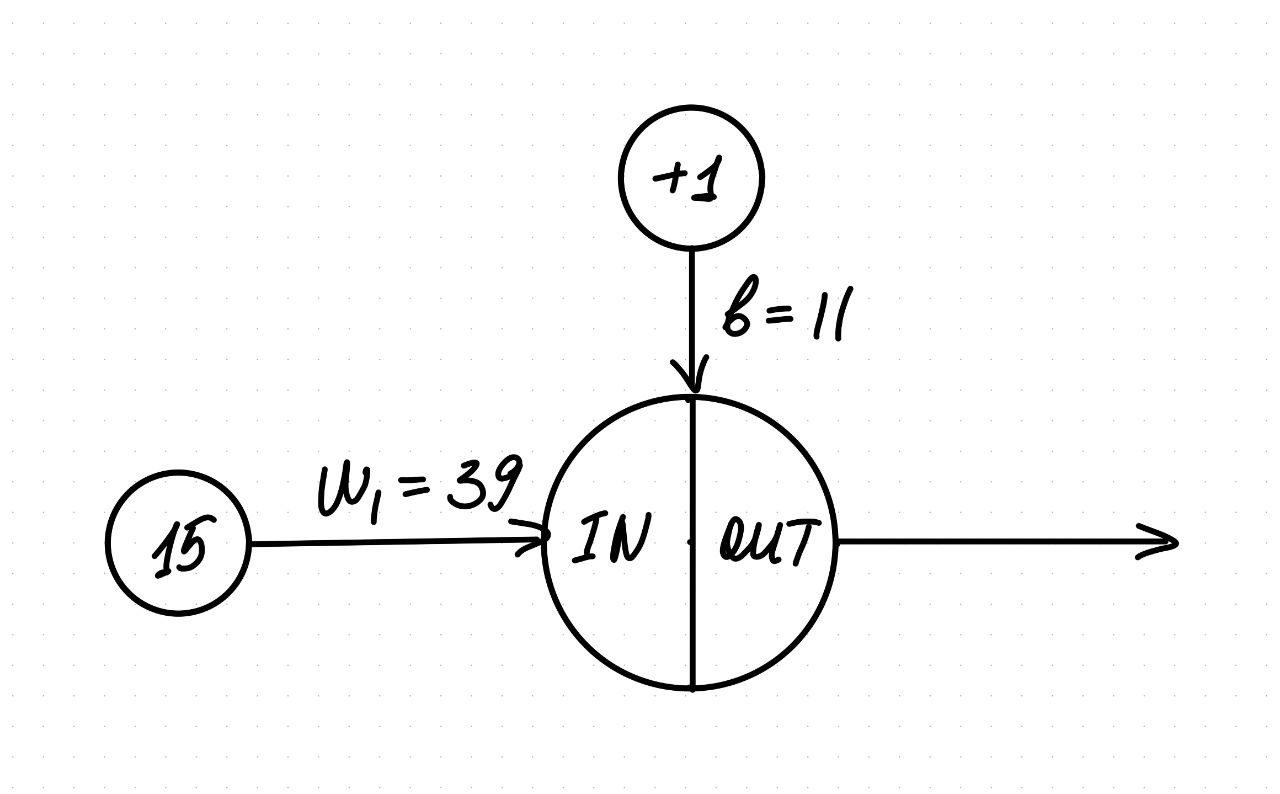
--- Конец 1 эпохи ---

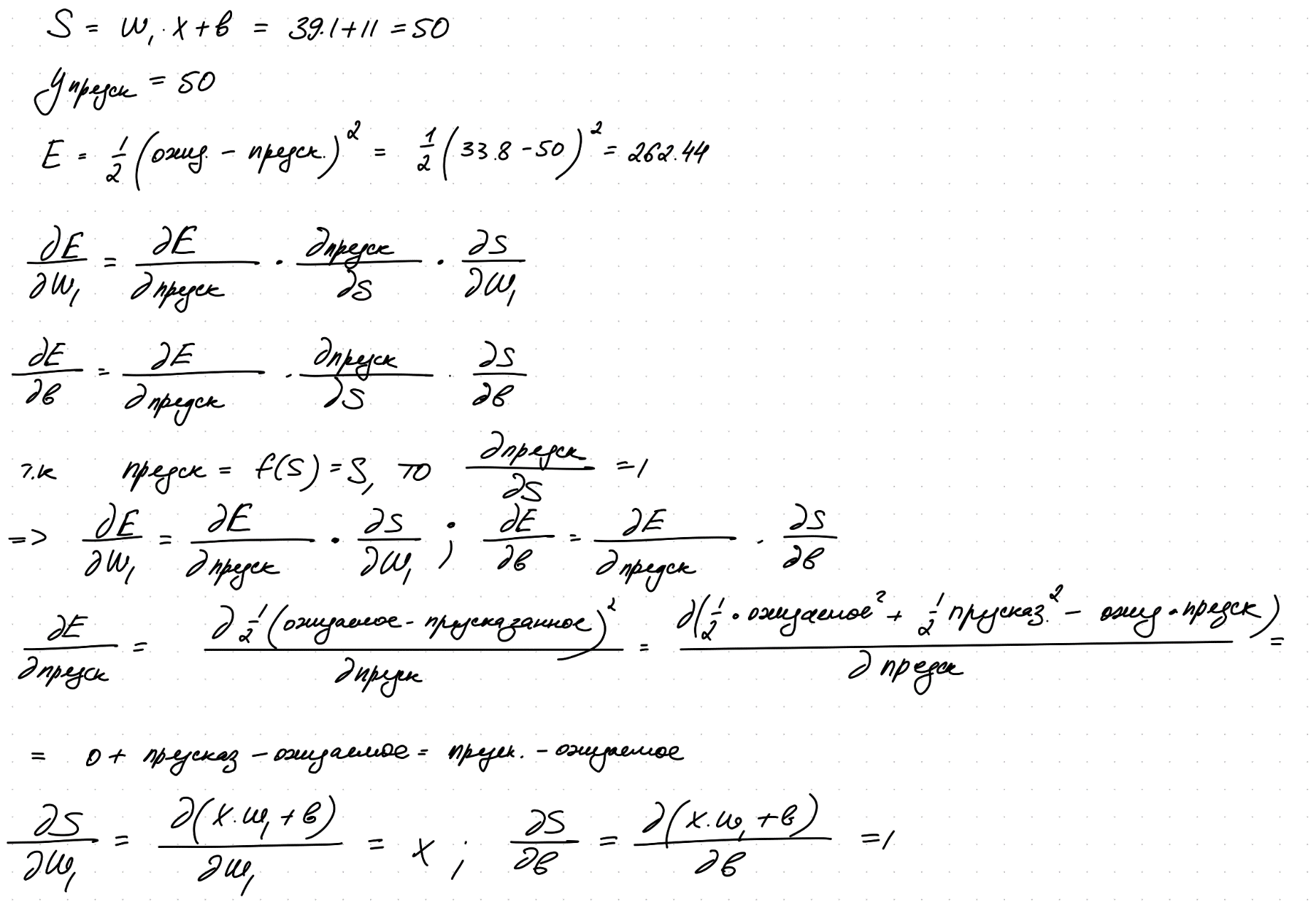
Таких эпох будет две.

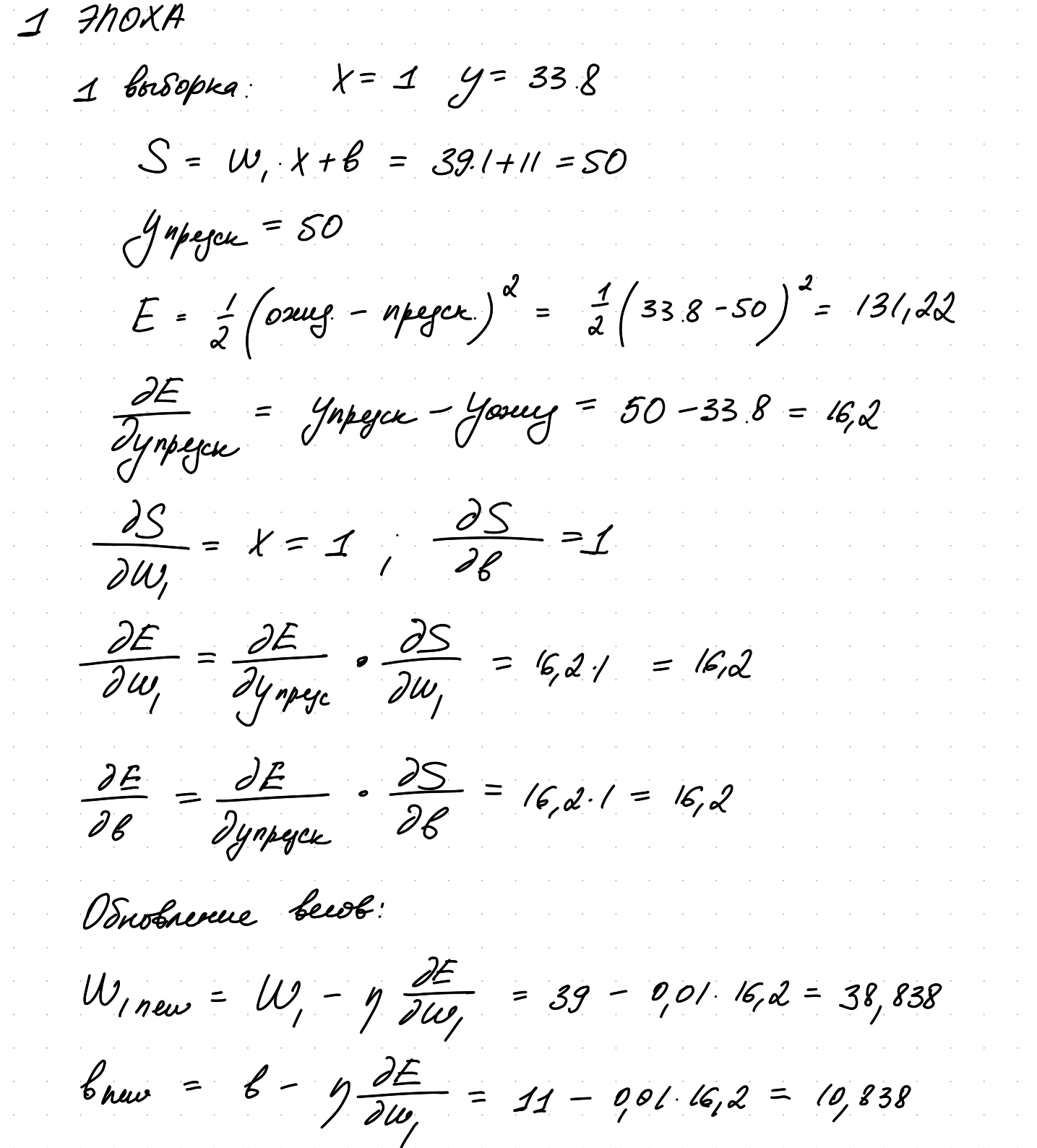
В качестве начальных значений для параметров возьмем вариант 22:

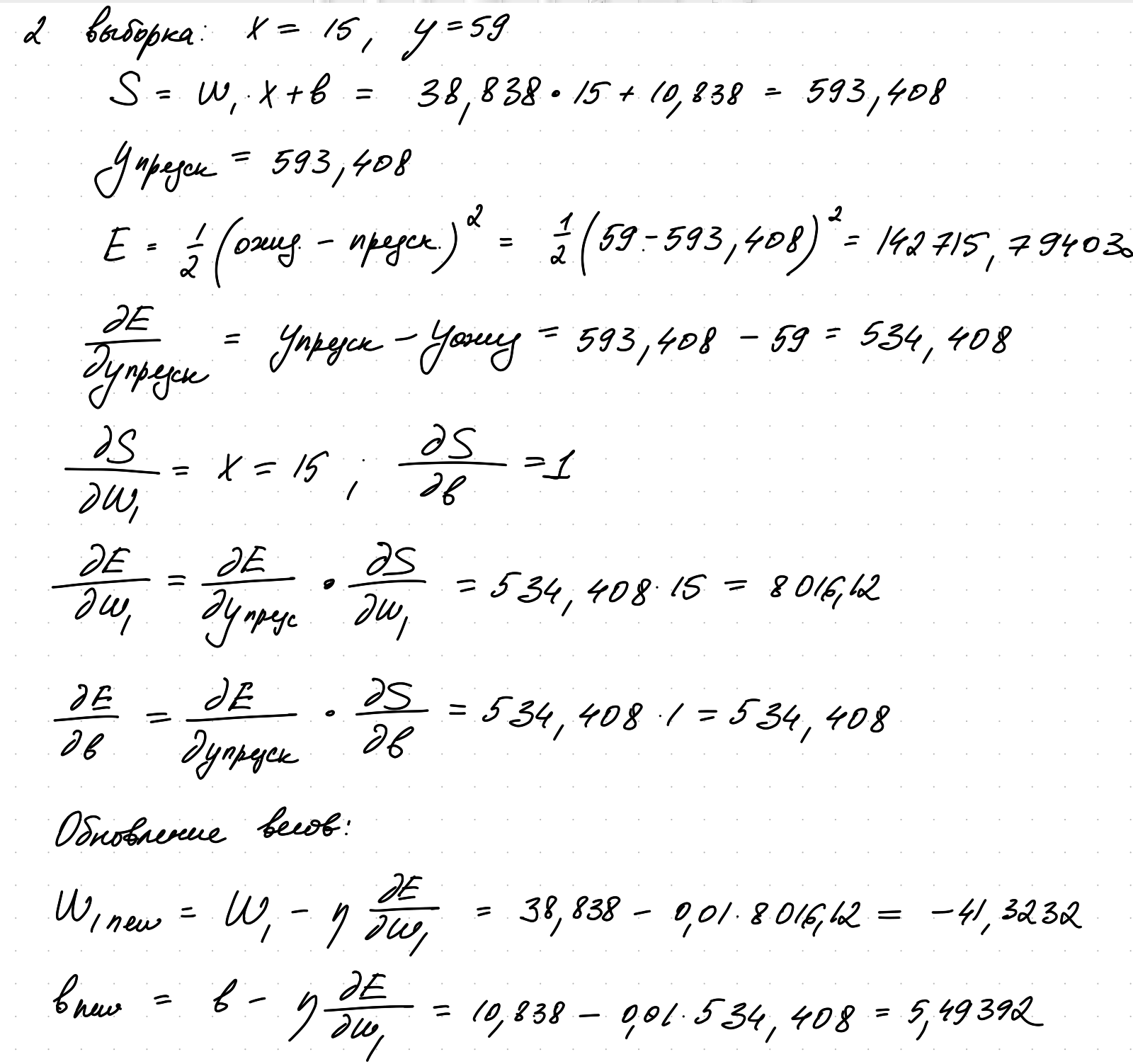
|  |  |  |
| --- | --- | --- |
| Вариант | W1 | b |
| 22 | 39 | 11 |

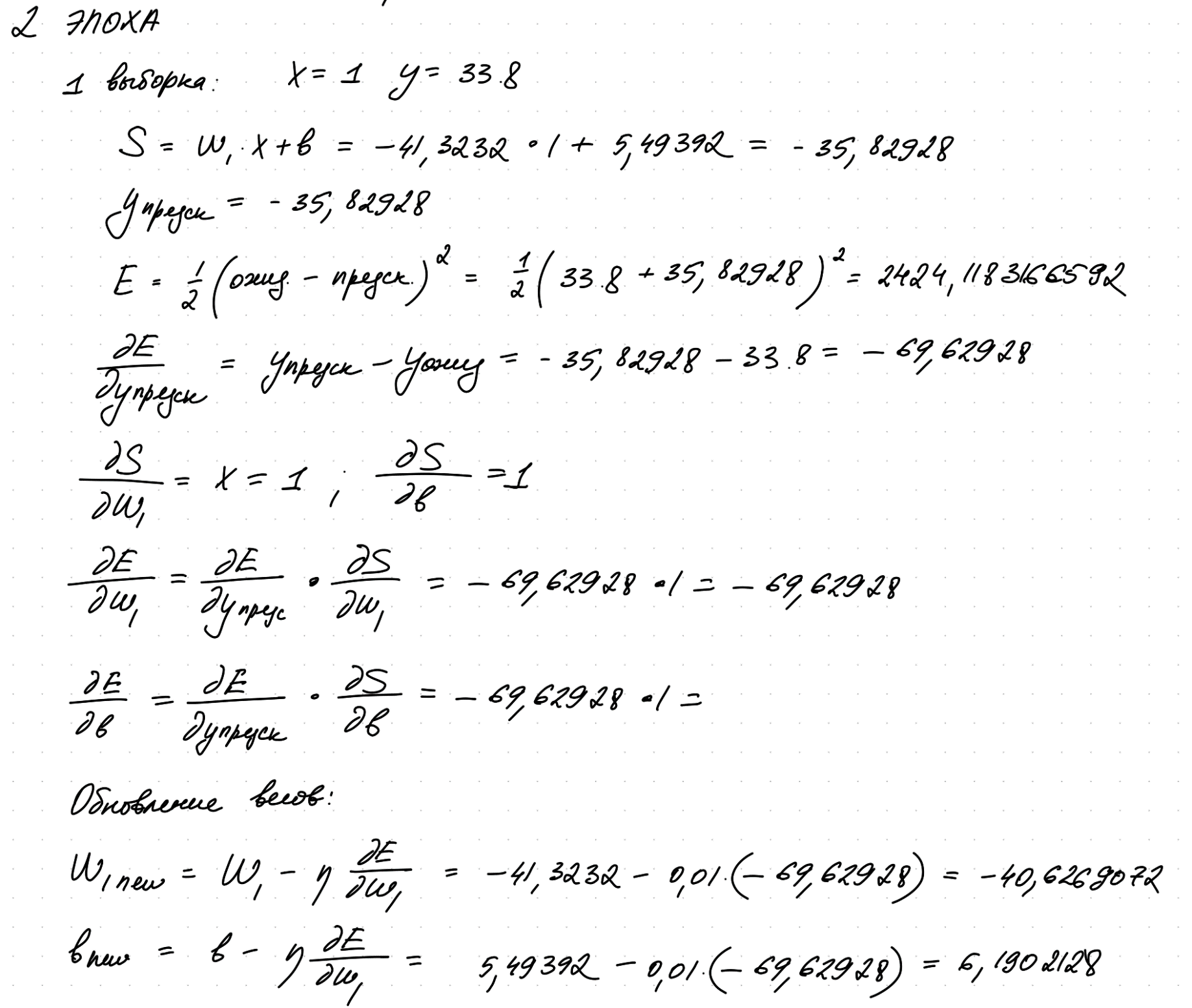
 1 набор данных

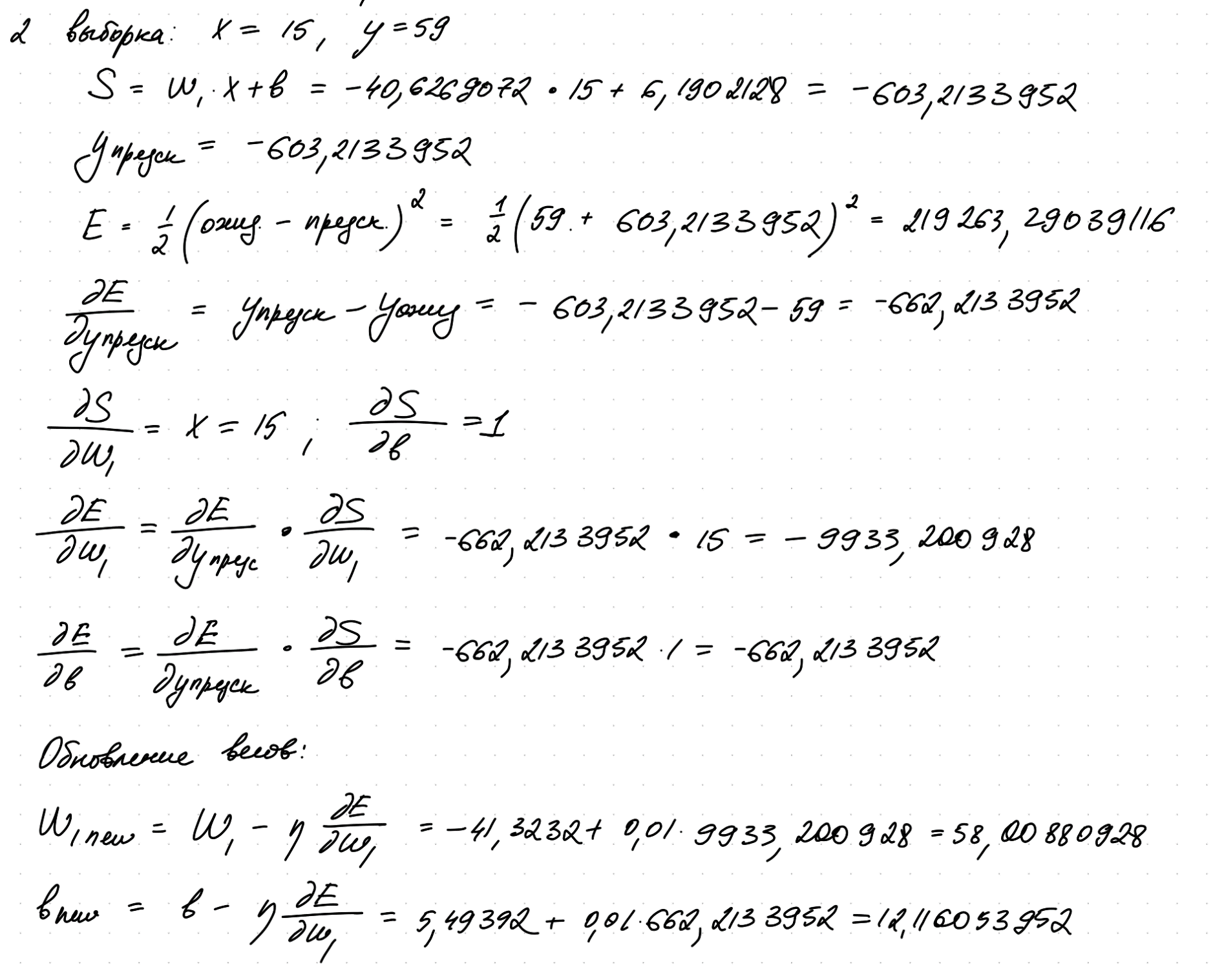
 2 набор данных











Как это работает:

Мы считаем частные производные от ошибки по весу и по смещению, и понимаем, в какую сторону и насколько нужно изменять параметры нейрона. Так, при знаке «-» производной, увеличение веса уменьшает ошибку и наоборот. Конкретно, мы должны уменьшить наше текущее значение параметра на скорость обучения, домноженную на производную.

Итог по двум эпохам:

|  |  |  |  |
| --- | --- | --- | --- |
| Итерация обучения | | Параметры нейронной сети | |
| W1 | b |
| Начальные значения (до обучения) | | 39 | 11 |
| 1 эпоха | После обучения на 1 примере | 38.838 | 10.838 |
| После обучения на 2 примере | -41.3232 | 5.49392 |
| 2 эпоха | После обучения на 1 примере | -40.6269072 | 6.1902128 |
| После обучения на 2 примере | 58,00880928 | 12,116053952 |

Далее представлена ссылка на гугл коллаб с кодом, представляющим этот алгоритм для 2 эпох и дополнительно для еще 98 эпох с более низким значением скорости обучения, а также графики с изменением ошибки и предсказаний.

Представлены также результаты проверки на тестовой выборке.

https://colab.research.google.com/drive/1V5PT\_Is\_3jR8A3WRearVPBAvUHSLuqsz?usp=sharing

1. **Вывод:**

При большей скорости обучения значения параметров скачут очень сильно, и более низкая скорость обучения дает более гладкое изменение параметров.

Итоговые значения параметров могут немного отличаться от истинных из-за округлений.

С любых значений параметров можно в действительности прийти к нужным при большом количестве итераций.