



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ _____ **«Информатика и системы управления»**

КАФЕДРА _____ **«Теоретическая информатика и компьютерные технологии»**

Лабораторная работа № 1
по курсу «Теория искусственных нейронных сетей»
«Реализация однослойного перцептрона»

Студент группы ИУ9-72Б Шевченко К.

Преподаватель Каганов Ю. Т.

Москва, 2023

Цель работы

1. Реализовать на языке высокого уровня однослойный персептрон и проверить его работоспособность на примере искусственных данных типа цифр от 0 до 9 и букв русского алфавита. Размер поля 5×4 .
2. Исследовать работу персептрона на основе использования различных функций активации (линейной, сигмоиды, гиперболического тангенса, ReLu).

Постановка задачи

1. Изучить основные принципы работы однослойного персептрона.
2. Разработать и реализовать на выбранном языке программирования структуру однослойного персептрона, включающую инициализацию весов и функций активации.
3. Создать обучающий набор данных для проверки работоспособности персептрона, который будет содержать искусственные примеры цифр от 0 до 9. Размер поля будет составлять 5×4 единицы.
4. Обучить персептрон на обучающем наборе данных, реализовав процесс обратного распространения ошибки и обновления весов.
5. Проверить работу обученного персептрона на тестовом наборе данных и оценить его производительность и точность классификации.
6. Провести исследование работы персептрона с использованием различных функций активации, таких как линейная, сигмоидная, гиперболический тангенс и ReLu. Сравнить результаты и проанализировать их влияние на производительность персептрона.

Реализация

Рисунок 1 демонстрирует модель однослойного персептрона для решения поставленной задачи. Нейронная сеть состоит из 10 нейронов. Каждый нейрон предназначен для распознавания конкретной цифры. Обучить персептрон необходимо таким образом, чтобы при подаче на вход изображения с цифрой i нейрон с номером i возвращал 1 (или, хотя бы, число, близкое к 1), а все остальные – 0.

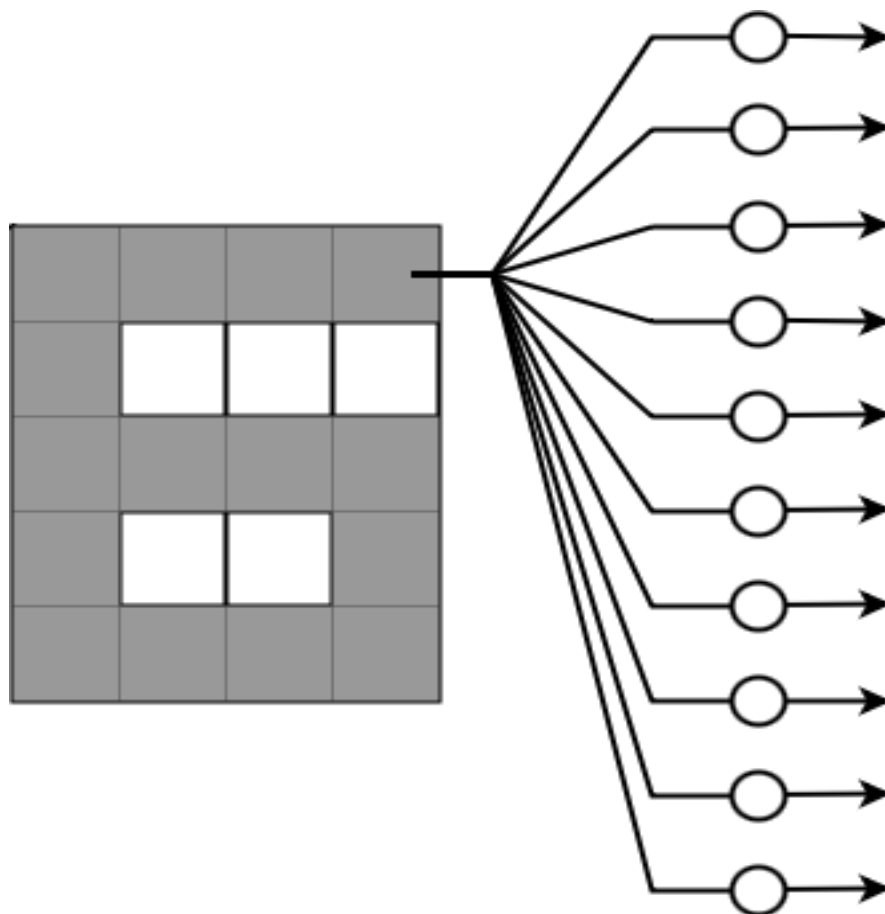


Рисунок 1 — Модель однослойного персептрона для решения задачи

На рисунке 2 представлен пример растрового изображения с цифрой 6, подаваемого на вход нейронной сети.



Рисунок 2 — Пример изображения 4x5, подаваемого на вход персептрона

Исходный код программы приведён в листингах 1–5.

Листинг 1: Импорт необходимых зависимостей

```

1 from PIL import Image
2 import matplotlib.pyplot as plt
3 import math

```

Листинг 2: Загрузка изображения, преобразование в матрицу

```
1 def load_image(file_path: str) -> list[list[int]]:
2     image = Image.open(file_path)
3     image = image.convert('L')
4     pixels = image.load()
5     width, height = image.size
6     matrix = [[0 for _ in range(width)] for _ in range(height)]
7     for j in range(height):
8         for i in range(width):
9             if pixels[i, j] < 128:
10                 matrix[j][i] = 1
11     return matrix
```

Листинг 3: Получение матриц, соответствующих изображениям

```
1 def get_images(path) -> list[list[list[int]]]:
2     images = []
3     for i in range(10):
4         file_path = f'{path + str(i)}.bmp'
5         images.append(load_image(file_path))
6     return images
```

Листинг 4: Класс для представления нейрона

```
1 class Neuron:
2     def __init__(self, weights):
3         self.weights = weights
4
5     def feedforward(self, input):
6         output = sum([input[i] * self.weights[i] for i in range(len(input))])
7         return output
8
9     @classmethod
10    def linear(self, x):
11        return x
12
13    @classmethod
14    def relu(self, x):
15        return max(0, x)
16
17    @classmethod
18    def sigmoid(self, x):
19        return 1 / (1 + math.exp(-x))
20
21    @classmethod
22    def tanh(self, x):
23        return (math.exp(x) - math.exp(-x)) / \
24            (math.exp(x) + math.exp(-x))
25
26    def activation_func_derivative(self, x):
27        delta = 0.001
```

```

28         return (self.activate(x + delta) - self.activate(x)) / delta
29
30     def activate(self, x):
31         # return self.linear(x)
32         # return self.relu(x)
33         # return self.sigmoid(x)
34         return self.tanh(x)

```

Листинг 5: Класс для представления однослойного персептрона

```

1  class OneLayerPerceptron:
2      def __init__(self, *neurons):
3          self.neurons = neurons
4
5      def feedforward(self, input):
6          outputs = []
7          for neuron in self.neurons:
8              output = neuron.activate(neuron.feedforward(input))
9              outputs.append(output)
10         return outputs
11
12     @classmethod
13     def find_mean_squared_error(self, true, outputs) -> float:
14         return 1 / len(true) * sum((true[i] - outputs[i]) ** 2 for i in
15                                     ↪ range(len(true)))
16
17     def train(self, epochs, learning_rate, data, trues):
18         epochs_x = []
19         mses = [[] for _ in range(len(self.neurons))]
20         mean_mse = []
21         for epoch in range(epochs):
22             digit_index = 0
23             for input, true in zip(data, trues):
24                 outputs = self.feedforward(input)
25                 mse = self.find_mean_squared_error(true, outputs)
26                 if epoch % 10:
27                     mses[digit_index].append(mse)
28                     for i, output in enumerate(outputs):
29                         # частные производные по весам
30                         #  $L = error$ 
31                         #  $d_L d_{wj} = d_L d_{output} * d_{output} d_{wj}$ 
32                          $d_L d_{output} = -2 * (true[i] - output)$ 
33                         #  $output = sigmoid(x1 * wi_1 + ... + x20 * wi_{20})$ 
34                         for j in range(len(self.neurons[i].weights)):
35                              $d_{output} d_{wj} = input[j] * \backslash$ 
36                                  $self.neurons[i].activation\_func\_derivative( \backslash$ 
37                                      $self.neurons[i].feedforward(input))$ 
38                              $d_L d_{wj} = d_L d_{output} * d_{output} d_{wj}$ 
39                              $self.neurons[i].weights[j] -= learning\_rate * d_L d_{wj}$ 
40             digit_index += 1
41         if epoch % 10:
42             epochs_x.append(epoch)
43             mean_mse.append(sum(mses[i][-1] for i in
44                               ↪ range(len(self.neurons))) / len(self.neurons))
45         for i in range(len(self.neurons) + 1):
46             fig, ax = plt.subplots(nrows=1, ncols=1)

```

```

45         fig.suptitle('Dependency of the MSE on learning epoch')
46         if i != len(self.neurons):
47             ax.set_title(f'Digit: {i}')
48         ax.set_xlabel('Epoch')
49         ax.set_ylabel('MSE')
50         if i != len(self.neurons):
51             ax.plot(epochs_x, mses[i], color='red')
52         else:
53             ax.plot(epochs_x, mean_mse, color='red')
54
55     def predict(self, data):
56         flatten_data = [j for elem in data for j in elem]
57         outputs = self.feedforward(flatten_data)
58         print(f'Результат обучения: {outputs}')
59         print(f'Предсказанная цифра: {str(outputs.index(max(outputs)))}')

```

Тестирование

Исходный код для тестирования разработанной программы представлен в листинге 6.

Листинг 6: Исходный код для тестирования программы

```

1  if __name__ == "__main__":
2      weights = [0 for _ in range(20)]
3
4      perceptron = OneLayerPerceptron(
5          Neuron(weights.copy()),
6          Neuron(weights.copy()),
7          Neuron(weights.copy()),
8          Neuron(weights.copy()),
9          Neuron(weights.copy()),
10         Neuron(weights.copy()),
11         Neuron(weights.copy()),
12         Neuron(weights.copy()),
13         Neuron(weights.copy()),
14         Neuron(weights.copy()),
15     )
16
17     flatten_data = []
18     images = get_images('./training_data/')
19     for image in images:
20         flatten_data.append([j for elem in image for j in elem])
21
22     trues = [[1 if i == j else 0 for j in range(10)] for i in range(10)]
23     perceptron.train(500, 0.05, flatten_data, trues)
24
25     for i in range(len(images)):
26         print(f'Матричное представление изображения:')
27         for row in images[i]:
28             print(row)
29         perceptron.predict(load_image(f"./training_data/{str(i)}.bmp"))
30
31     plt.show()

```

Тестирование проводилось путём обучения однослойного персептрона на 1000 эпохах. На рисунках 3–6 представлены результаты обучения однослойного персептрона с использованием линейной функции, ReLu, сигмоиды, гиперболического тангенса в качестве функций активации нейронов.

```
Результат обучения: [0.9992793215870867, 0.00037766138765549934, -0.0009739240864682919, 0.0002763673137320122, -0.00022318902211423963, 0.00039061239557330873, -0.0015822158182324797, -7.580335417937956e-05, 0.004492030075303077, -0.002260679592443693]
Предсказанная цифра: 0
Результат обучения: [0.00034521046866176075, 0.9998114435389962, 0.00046563873297850167, -0.00011147640775957535, 0.00010260303889961242, -0.00023532110754381774, 0.0008552014986215051, 4.430066749301631e-05, -0.002259838189201391, 0.0011272762850241813]
Предсказанная цифра: 1
Результат обучения: [-0.0010143447138166173, 0.0005298977662889295, 0.9986290250155507, 0.00039350541550550733, -0.0003150671399667626, 0.00053935228052652, -0.0022058979633758913, -0.00010496370872896399, 0.0062990831629052835, -0.003172273604132969]
Предсказанная цифра: 2
Результат обучения: [-9.418334561742125e-05, 6.844219191864953e-05, -0.0001250881089512254, 0.9999839825255187, -1.8428688418392625e-05, 0.0001712865249788309, -0.0004494314962731824, -2.9832891821623964e-05, 0.0008566827073208522, -0.0004061485223485206]
Предсказанная цифра: 3
Результат обучения: [-0.0002864026897688443, 0.00014550998974836166, -0.0003875698588247656, 0.00012232820157578872, 0.9999087285775867, 0.0001264087661043134, -0.0005706117273628908, -2.5348025714008737e-05, 0.0017205294221045442, -0.000871872345902247]
Предсказанная цифра: 4
Результат обучения: [0.00035219746497008875, -0.00023303004598096422, 0.00047039566324109244, -2.678027060498689e-06, 8.180372044835904e-05, 0.9995037921576537, 0.0013894253649244098, 8.764310430081634e-05, -0.002879929427776498, 0.0013859123965600795]
Предсказанная цифра: 5
Результат обучения: [-0.0023420065719915173, 0.001313953555101395, -0.0031550372599590815, 0.0006614134571809316, -0.0006765446876174919, 0.0018152912627041329, 0.993756469786713, -0.0003368099558815202, 0.01582205131888781, -0.007849220517858968]
Предсказанная цифра: 6
Результат обучения: [-0.0005073822652188198, 0.00028199414203677886, -0.0006838266242666993, 0.0001505742347292522, -0.00014806963280428187, 0.00037647597846434885, -0.0013187260487242114, 0.9999298155998054, 0.0033900904023490275, -0.0016850231442808589]
Предсказанная цифра: 7
Результат обучения: [0.003991888174808944, -0.0021814823049333062, 0.005384350935175364, -0.0012861092981818756, 0.0011858532708946823, -0.002728000228877814, 0.009903028908992417, 0.0005134093442957371, 0.9738527354086643, 0.013041699928971862]
Предсказанная цифра: 8
Результат обучения: [-0.0015494616305707365, 0.0008461537521661858, -0.0020900184596461258, 0.0005008306743495561, -0.00046062675232297345, 0.0010551354641230026, -0.0038363280057334137, -0.00019866044609211098, 0.010140731665518876, 0.9949412770845766]
Предсказанная цифра: 9
```

Рисунок 3 — Результаты обучения однослойного персептрона (линейная функция)

```
Результат обучения: [0.777119029274315, 0.005023980459898593, 0.03352802202773332, 0.03992705765497062, 0.043380958422586456, 0.023487542192921713, 0.12402648892163083, 0.01771060132554682, 0.22284980774037927, 0.1354995646356561]
Предсказанная цифра: 0
Результат обучения: [0.014476887839020678, 0.8887172373594225, 0.08614592806612957, 0.06544512987122998, 0.07793977178545124, 0.039777197478659244, 0.04719455953444509, 0.04806553965088684, 0.03214682207175668, 0.06117017757044381]
Предсказанная цифра: 1
Результат обучения: [0.0388680527982746, 0.04808958055494589, 0.8208706355400452, 0.10505266489855916, 0.01162092338743098, 0.0189993281868181, 0.08972930203546053, 0.04489321891875346, 0.16995915286551025, 0.05484773623822998]
Предсказанная цифра: 2
Результат обучения: [0.07630506763276437, 0.0502852048518315, 0.11623494846977694, 0.8033489741560618, 0.013526921760925662, 0.009863535607204368, 0.0167901170902278, 0.0908989400452624, 0.06157556739378849, 0.09090519975858245]
Предсказанная цифра: 3
Результат обучения: [0.10535495201966419, 0.0825510517957495, 0.029202861240590464, 0.018565369166318155, 0.9051766286225901, 0.015676540624144635, 0.1381047914807905, 0.04449025894122052, 0.11092820289973478, 0.05342343038961736]
Предсказанная цифра: 4
Результат обучения: [0.043138716128746125, 0.021275189506526425, 0.02620775836845407, 0.088110972744228, 0.011441786105709069, 0.7935869821133179, 0.22163386054181086, 0.03994511412371243, 0.12395054167355768, 0.18204468947396937]
Предсказанная цифра: 5
Результат обучения: [0.08279182897489268, 0.00775970916442221, 0.050530394414574166, 0.010083771733534887, 0.02443415537252013, 0.12343627486890912, 0.6016397966235388, 0.008667379894195935, 0.24002598373133546, 0.12021307395228824]
Предсказанная цифра: 6
Результат обучения: [0.0518713065168811, 0.049493057281179295, 0.07531230707831704, 0.11106981521516111, 0.034883547747632, 0.0690263649101967, 0.017500057323894973, 0.871104355456254, 0.07108854184458913, 0.08873435090629059]
Предсказанная цифра: 7
Результат обучения: [0.12045088257771445, 0.004313545410349046, 0.06316116245968445, 0.014801983742866792, 0.016084967822130224, 0.03673039943332959, 0.19221358006489547, 0.011994498543010977, 0.3337289117482982, 0.20561759767197976]
Предсказанная цифра: 8
Результат обучения: [0.08391789008158822, 0.008126254753742273, 0.026742084377054387, 0.03012240090166001, 0.011029022470743693, 0.08489293025719925, 0.10132647824576822, 0.018946709492371993, 0.2281335374649943, 0.609635017896715]
Предсказанная цифра: 9
```

Рисунок 4 — Результаты обучения однослойного персептрона (сигмоида)


```

Результат обучения: [0.9162660316475467, 0.00045221545490205906, -0.0067660562282855375, 0.005703477120376925, -0.0002567355290248863,
-0.003989501572846514, 0.00044393711942021253, -2.741038757045365e-05, 0.01596650266213287, -0.005049982200628487]
Предсказанная цифра: 0
Результат обучения: [-0.000719772355394245, 0.9533505732421615, 0.00319263177931891, 0.0005169553163254449, 0.0023488596019791883,
-0.0006266397164780795, 0.0026457550322653107, 5.7316065334268985e-05, -0.006585331093579999, 0.004460681121772038]
Предсказанная цифра: 1
Результат обучения: [-0.009019578824682352, 0.0033357835486156275, 0.9120380890674119, 0.008748541752151125, -0.0036590488175480425,
-0.005953621586389316, 0.0012512004497812238, 0.000959450961624087, 0.020975700954392884, -0.011365945408271376]
Предсказанная цифра: 2
Результат обучения: [0.0019605741016866245, 0.0005051742677927103, 0.004044988395175691, 0.9222705312339893, 0.00043382617412886015,
0.006258669675600967, -0.007358732232982671, 0.0008823825576462728, -0.00015016967778632836, 0.0009651751962995412]
Предсказанная цифра: 3
Результат обучения: [-0.001859930013406111, 0.0021447792869203446, -0.0047705949226859974, 0.0016560967344771082, 0.9520441527055384,
-0.0037070276332760777, 0.0026739028392775283, 0.0007222047497788414, 0.0066648871775840035, -0.0039235858456030375]
Предсказанная цифра: 4
Результат обучения: [-0.006458763384955125, -7.587179971968622e-05, -0.007288701383085678, 0.008415738423138508, -0.003625179288162417,
0.9000899018702083, 0.01898692854093166, 0.002524770877366274, -0.0002456335303722561, 0.00548237037435857]
Предсказанная цифра: 5
Результат обучения: [-0.005988446416273871, 0.0046739271838004606, -0.005704156070337872, -0.004618664902612419, 0.0003600542036174115,
0.01576899799828586, 0.0478207858373262, -0.003386080404320884, 0.03901881022421754, -0.01737541209546227]
Предсказанная цифра: 6
Результат обучения: [-0.0024694210899018887, 0.0009216459960574538, -0.001440098167482218, 0.0011254077786439173, 6.311668372223297e-05,
0.0023260707747006747, -0.007092264627410965, 0.954072262036625, 0.010168483743494954, -0.0023533780413212495]
Предсказанная цифра: 7
Результат обучения: [0.0174894579823929, -0.009631105804937538, 0.023146408498168364, -0.006322017272190623, 0.0061447660321998394,
-0.009231144839790857, 0.0325433404694615, 0.002018821525359417, 0.7777406786736193, 0.0475367566085458]
Предсказанная цифра: 8
Результат обучения: [-0.004294998540356138, 0.004116365994700526, -0.008935218614661239, 0.0031399840060439468, -0.0028628393098402644,
0.005306618815828202, -0.009558325445873027, -4.323715218762238e-05, 0.028192067210660365, 0.8624103295325897]
Предсказанная цифра: 9

```

Рисунок 5 — Результаты обучения однослойного персептрона (гиперболический тангенс)

```

Результат обучения: [0.9999999998967005, 0, 0, 0, 1.457167719820518e-16, 0, 0, 0, 0.0023412868680938725, 0]
Предсказанная цифра: 0
Результат обучения: [0, 0.9999999999999997, 8.326672684688674e-17, 0, 1.734723475976807e-16, 0, 0, 0, 2.579334201779915e-07]
Предсказанная цифра: 1
Результат обучения: [0, 1.1102230246251565e-16, 0.9999999999999993, 6.938893903907228e-16, 0, 0, 0, 0.002208287688268973, 0]
Предсказанная цифра: 2
Результат обучения: [0, 0, 0, 0.9999999999999996, 0, 1.8016144132104728e-13, 0, 6.38378239159465e-16, 0, 3.004444257548311e-07]
Предсказанная цифра: 3
Результат обучения: [0, 1.6653345369377348e-16, 0, 0, 0.9999999999999999, 0, 3.967868382304118e-06, 0, 0.0006886883307098968, 0]
Предсказанная цифра: 4
Результат обучения: [0, 0, 0, 3.0531133177191805e-16, 0, 0.99999999999995488, 1.0356617789149258e-05, 0, 0, 2.4436031443031414e-06]
Предсказанная цифра: 5
Результат обучения: [0, 0, 0, 0, 0, 0, 2.220446049250313e-13, 0.9999694724454465, 0, 0.005177318040811452, 0]
Предсказанная цифра: 6
Результат обучения: [0, 0, 0, 0, 0, 0, 0.9999999999999993, 0, 6.513933900192947e-07]
Предсказанная цифра: 7
Результат обучения: [6.420487058678148e-11, 0, 4.718447854656915e-16, 0, 0, 0, 1.5621983890434787e-05, 0, 0.991208212122199, 1.4036835166486572e-05]
Предсказанная цифра: 8
Результат обучения: [0, 0, 0, 0, 0, 0, 0, 0.0027786409954460023, 0.9999886472679471]
Предсказанная цифра: 9

```

Рисунок 6 — Результаты обучения однослойного персептрона (ReLU)

Графики зависимости функции потерь от количества эпох продемонстрированы на рисунках 7–10.

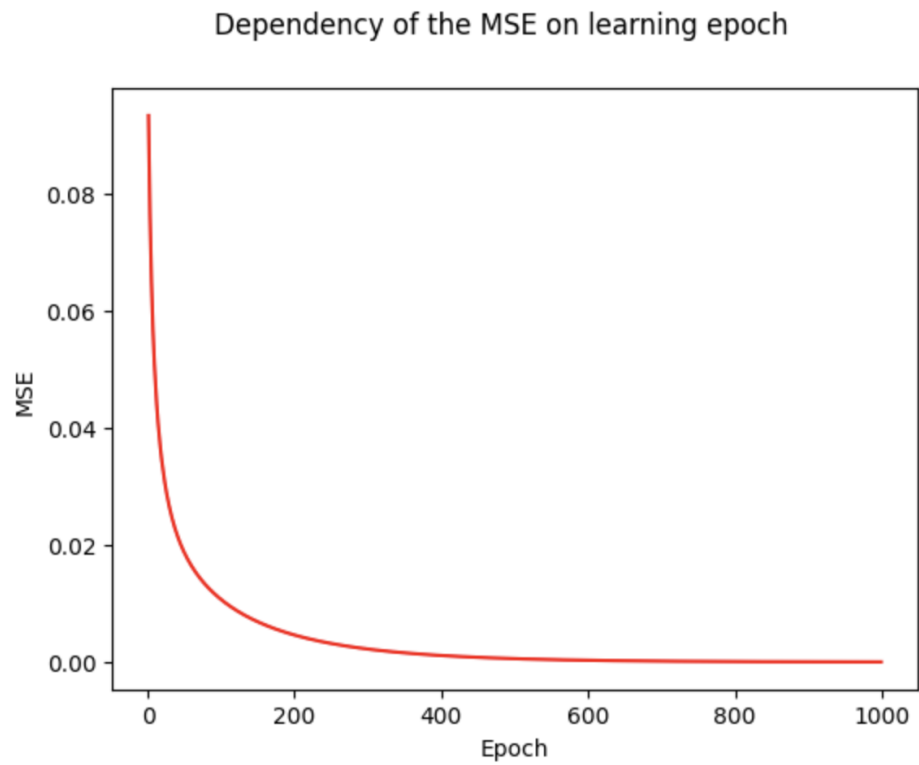


Рисунок 7 — График зависимости функции потерь от количества эпох (линейная функция)

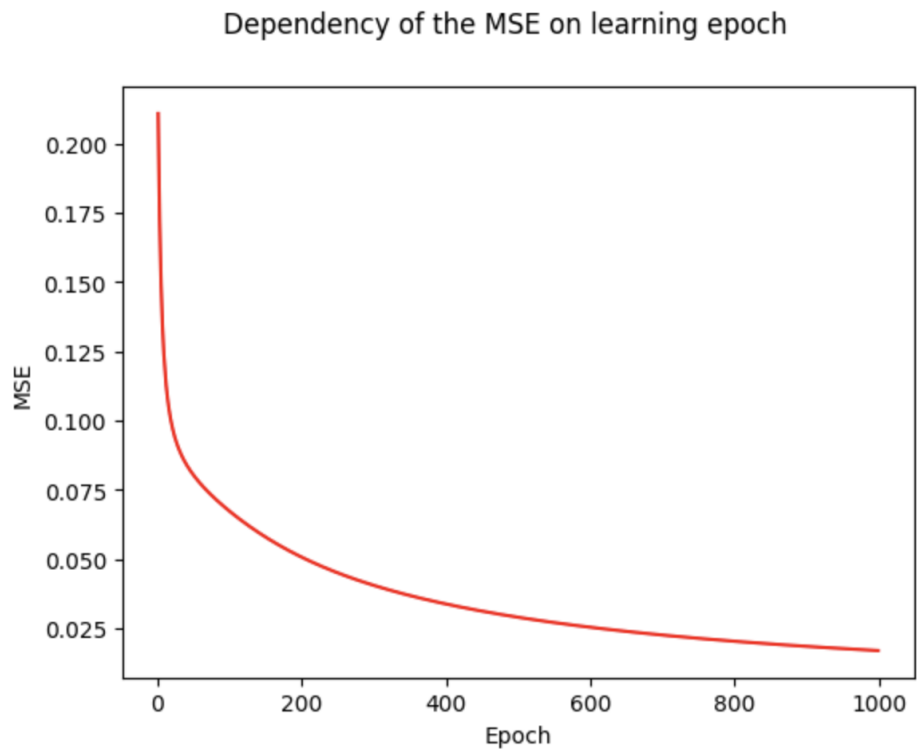


Рисунок 8 — График зависимости функции потерь от количества эпох (сигмоида)

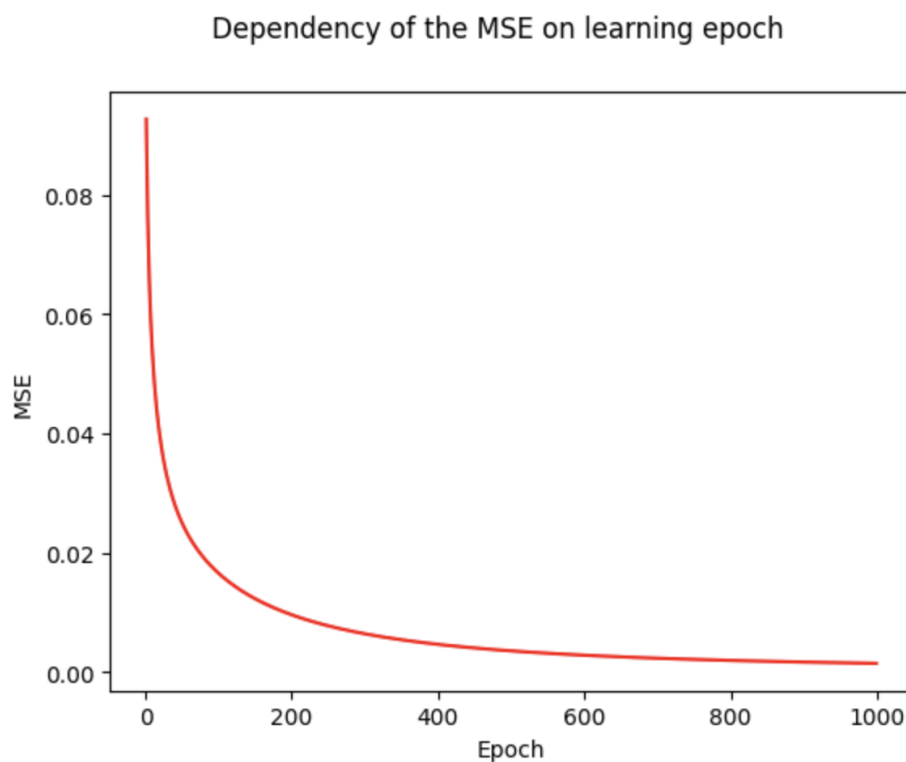


Рисунок 9 — График зависимости функции потерь от количества эпох (гиперболический тангенс)

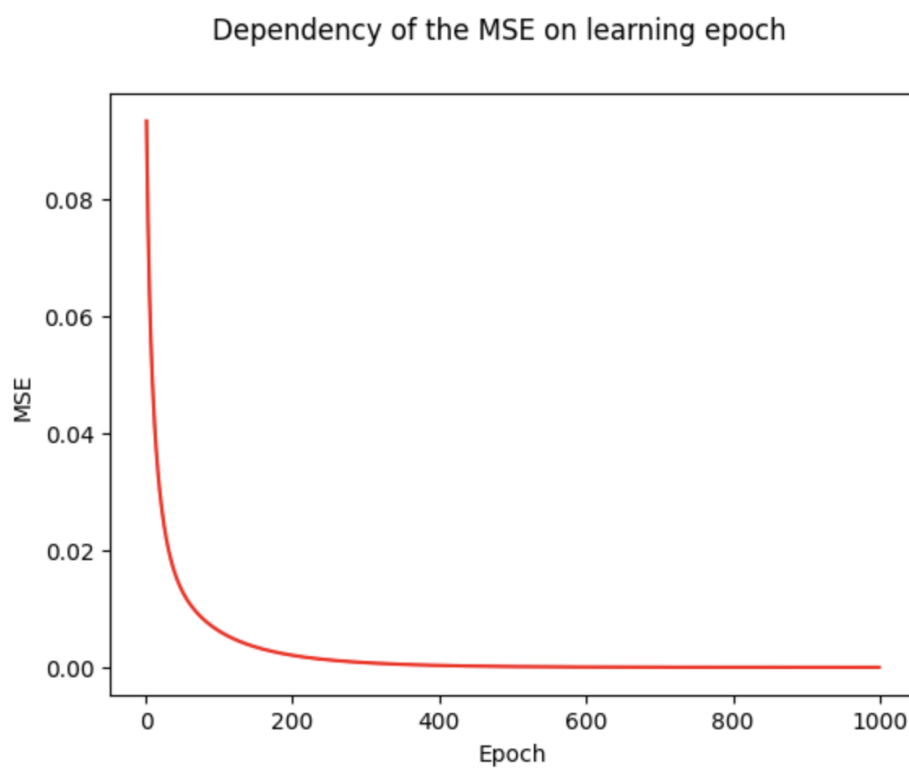


Рисунок 10 — График зависимости функции потерь от количества эпох (ReLU)

Выводы

В ходе выполнения лабораторной работы была реализована простейшая нейронная сеть в виде однослойного персептрона на языке высокого уровня Python. Персептрон был обучен и протестирован на искусственных данных, представляющих цифры от 0 до 9. Проведено исследование работы персептрона с использованием различных функций активации.

Полученные результаты позволяют сделать следующий вывод: линейная функция, функция сигмоиды и гиперболический тангенс показали хорошие результаты в качестве функций активации, однако сигмоида проявила себя хуже. Наилучший результат показала функция ReLu, позволив персептрону эффективно обучиться на входных данных и дать наиболее высокую точность классификации среди рассмотренных функций активации.

На основе проведённого исследования можно сделать вывод, что выбор функции активации играет важную роль в работе персептрона. Каждая функция имеет свои преимущества и недостатки, и выбор наиболее подходящей функции зависит от конкретной задачи.