

✓ 24. Gradient Descent

✓ 25. Logistic Regression Algorithm

✓ 26. Pre-Notebook: Gradient Desc...

✓ 27. Notebook: Gradient Descent

✓ 28. Perceptron vs Gradient Descent

✓ 29. Continuous Perceptrons

✓ 30. Non-linear Data

✓ 31. Non-Linear Models

✓ 32. Neural Network Architecture

✓ 33. Feedforward

✓ 34. Backpropagation

✓ 35. Pre-Notebook: Analyzing Stud...

✓ 36. Notebook: Analyzing Student ...

✓ 37. Outro

Knowledge

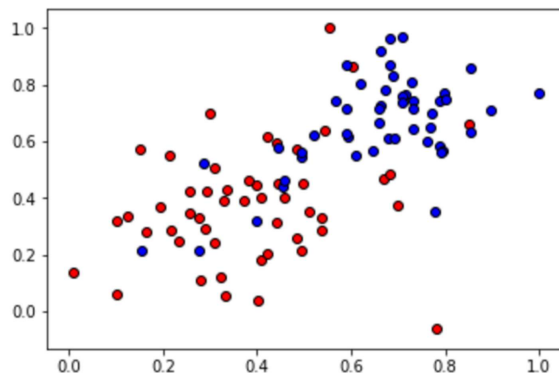
Search project Q&A

Student Hub

Chat with peers and mentors

Implementing Gradient Descent

In the following notebook, you'll be able to implement the gradient descent algo sample dataset with two classes.



Red and blue data points with some overlap.

Workspace

To open this notebook, you have two options:

- Go to the next page in the classroom (recommended)
- Clone the repo from [Github](#) and open the notebook **GradientDescent.ipynb** in the **networks > gradient-descent** folder. You can either download the repository line with

```
git clone https://github.com/udacity/deep-learning-v2-py
```

download it as an archive file from [this link](#).

Instructions

In this notebook, you'll be implementing the functions that build the gradient descent namely:

- `sigmoid`: The sigmoid activation function.
- `output_formula`: The formula for the prediction.
- `error_formula`: The formula for the error at a point.
- `update_weights`: The function that updates the parameters with one gradient step.

When you implement them, run the `train` function and this will graph the seven data points. The points are drawn in successive gradient descent steps. It will also graph the error function decreasing as the number of epochs grows.

This is a self-assessed lab. If you need any help or want to check your answers, find the solutions notebook in the same folder, or by clicking [here](#).