

DLIP - Assignment 1: Training

Bradley Spronk - s2504758

February 12, 2026

1 Python data science refresher.

[Link of github repository to past python project](#) demonstrating I don't need the refresher.

2 Tensorflow playground.

Duarte Homework 1 Problem 4A: First, select a linear model with no hidden layers. For the features, select the two independent variables x_1 and x_2 . Can you fit the data with this linear model? Why or why not?

It is not possible to fit the data with a linear model with only the features x_1 and x_2 , because the data is not linearly separable. Or in other words, we cannot draw a straight line that separates the two classes of points in the feature space defined by x_1 and x_2 .

What happens if you add the feature x_1x_2 ?

However, if we add the feature x_1x_2 , we can fit the data perfectly, because it allows us to capture the interaction between x_1 and x_2 , which is necessary to separate the classes.

We can't change the output neurons activation. For instance when there is no hidden layer and we choose the ReLU model ($\max(0, x)$) the output still shows negative numbers.

Now, return the features to just (x_1, x_2) and start adding hidden layers. What's the smallest neural network (least number of layers and least number of neurons per layer) you can create that fits the training data "perfectly" (i.e. a training loss 0.001)?

I suppose we have to use a non-linear activation function in the hidden layer, otherwise we would just be fitting a linear model again. With ReLU, a neural network with 1 hidden layer with 4 neurons is the smallest neural network that fits the training data perfectly (with a training loss of ≤ 0.001). With tanh, a neural network with 1 hidden layer with 5 neurons is the smallest neural network that fits the training data perfectly.

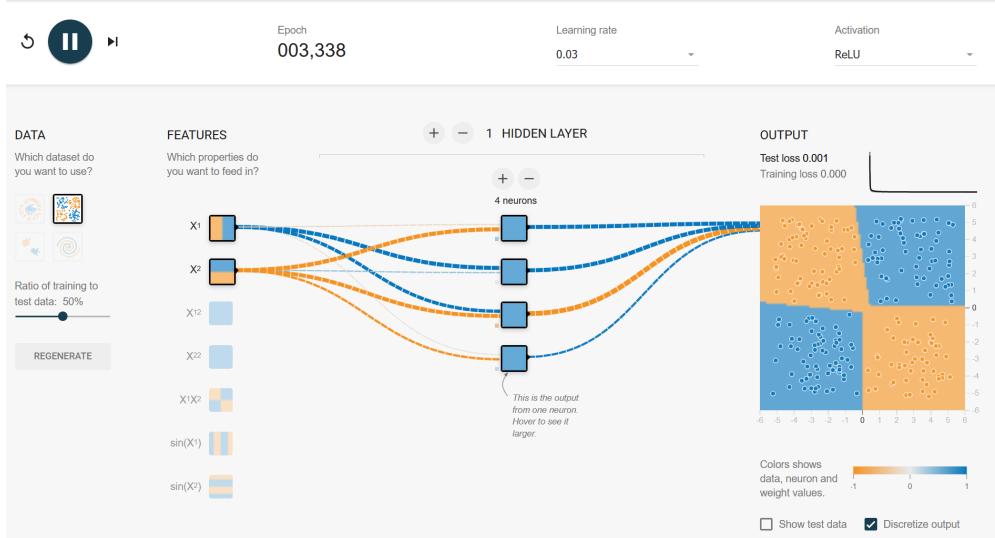


Figure 1: Screenshot of the smallest neural network that fits the training data perfectly.

What is the corresponding test loss?

Detail your hyperparameter choices by providing a screenshot and the URL to your solution (the URL contains all your settings choices).

Duarte Homework 1: Problem 4B

Navigate your web browser to the TensorFlow Playground <https://playground.tensorflow.org/#dataset=spiral&discretize=true>. Select the spiral pattern for the data.

Using all the features available, find a solution that fits the training data. What training and test error do you achieve? Is this a low bias/high variance or high bias/low variance model? How do you know?

Using only (x_1, x_2) , find a solution that fits the training data. What training and test error do you achieve? Is this a low bias/high variance or high bias/low variance model? How do you know?

For both solutions, detail your hyperparameter choices by providing a screenshot and the URL to your solution (the URL contains all your settings choices).

3 Training curves for polynomial regression.

Question on Brightspace: A sustained increase in the training error with epochs is unusual and indicates some problem (e.g. too high learning rate.). However, a sustained increase in training error as you increase the amount of training data is quite normal if the test error is high. (Why?)

This is because when the training data is small the model can sort of memorize the test data (test error is high in this case). When you increase the training data the training error will increase because now it has to learn to actually model the data well.

Duarte Homework 1 Problem 2C

The dataset `bv_data.csv` is provided and has a header denoting which columns correspond to which values. Using this dataset, plot learning curves for 1st-, 2nd-, 6th-, and 12th-degree polynomial regression (4 separate plots) by following these steps for each degree $d \in \{1, 2, 6, 12\}$:

1. For each $N \in \{20, 25, 30, 35, \dots, 100\}$:
 - (a) Perform 5-fold Cross-validation on the first N points in the dataset (setting aside the other points), computing the both the training and validation error for each fold.
 - Use the mean squared error loss as the error function.
 - Use NumPy's `polyfit` method to perform the degree- d polynomial regression and NumPy's `polyval` method to help compute the errors. (See the example code and [NumPy documentation](#) for details.)
 - When partitioning your data into folds, although in practice you should randomize your partitions, for the purposes of this set, simply divide the data into K contiguous blocks.
 - (b) Compute the average of the training and validation errors from the 5 folds.
2. Create a learning curve by plotting both the average training and validation error as functions of N .

Duarte Homework 1 Problem 2D

Based on the learning curves, which polynomial regression model (i.e. which degree polynomial) has the highest bias? How can you tell?

Duarte Homework 1 Problem 2E

Which model has the highest variance? How can you tell?

3.1 Duarte Homework 1 Problem 2F

What does the learning curve of the quadratic model tell you about how much the model will improve if we had additional training points?

3.2 Duarte Homework 1 Problem 2G

Why is training error generally lower than validation error?

3.3 Duarte Homework 1 Problem 2H

Based on the learning curves, which model would you expect to perform best on some unseen data drawn from the same distribution as the training data, and why?

4 Stochastic gradient descent for linear regression.

Duarte Homework Problem 2 Problem 1A-I

[Open .ipynb](#)