university of groningen

faculty of science and engineering

Assignment 1: Deep Learning in Physics

Problem 2: TensorFlow Playground

Ilija Antun Gjerapić, S4437586, i.a.gjerapic@student.rug.nl

Supervisor:

February 9, 2025

Problem A

Select the checkerboard pattern for the data, which is known as the XOR dataset. The data is sampled from a 2D probability density distribution represented by (x_1, x_2). The regions $x_1, x_2 > 0$ and $x_1, x_2 < 0$ have a target value $y = +1$ and are shown as blue data points, while the regions $x_1 > 0, x_2 < 0$ and $x_1 < 0, x_2 > 0$ have a target value of $y = -1$ and are shown as orange data points.

First, select a linear model with no hidden layers. For the features, select the two independent variables x1 and x2. Can you fit the data with this linear model? Why or why not? What happens if you add the feature x_1x_2?

We can see that data will never be fitted with the linear model (Figure 1) as both the training loss and test loss are about the same. This implies that the model has a high bias always underfitting, meaning it has a high bias. More fundamentally, the data requires $y = +1$ when $x_1 \cdot x_2 > 0$ and $y = -1$ for $x_1 \cdot x_2 < 0$. This can never be acheived with only a linear model of the independent variables x_1 and x_2 as features. Therefore, a combination of the variables as a new additional feature is necessary e.g. x_1x_2. This is shown in Figure 2. We see that after 1,025 epochs, the neural network fits the training data "perfectly" with a test loss of 0.002.

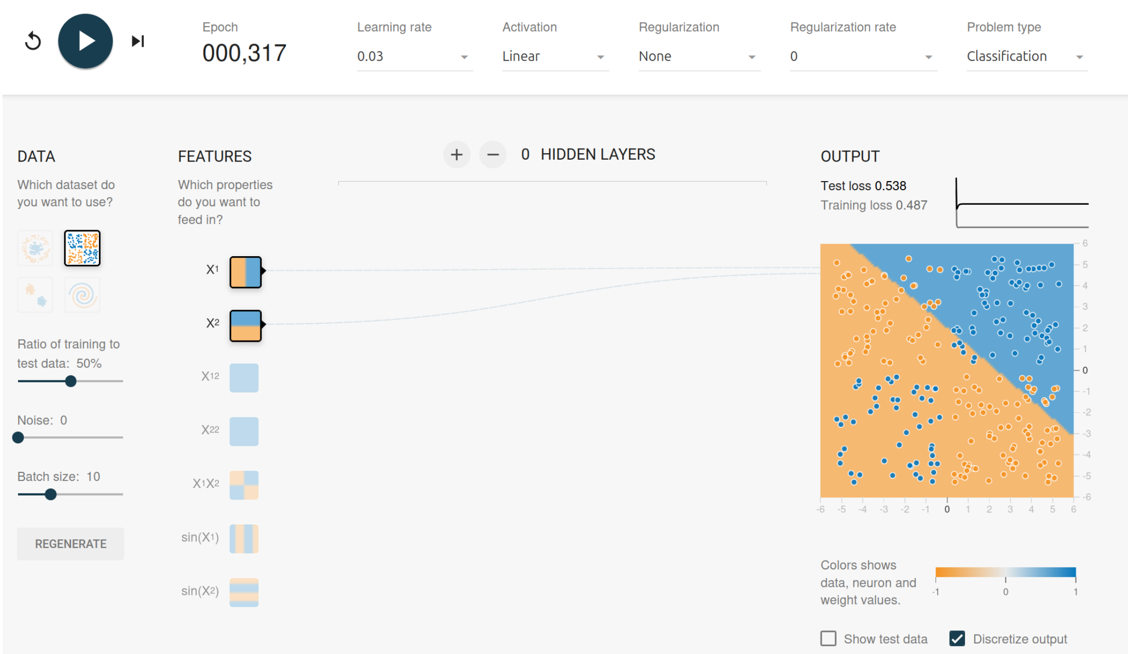

Figure 1: A linear model with no hidden layers using the two independent variables x_1 and x_2 as features.

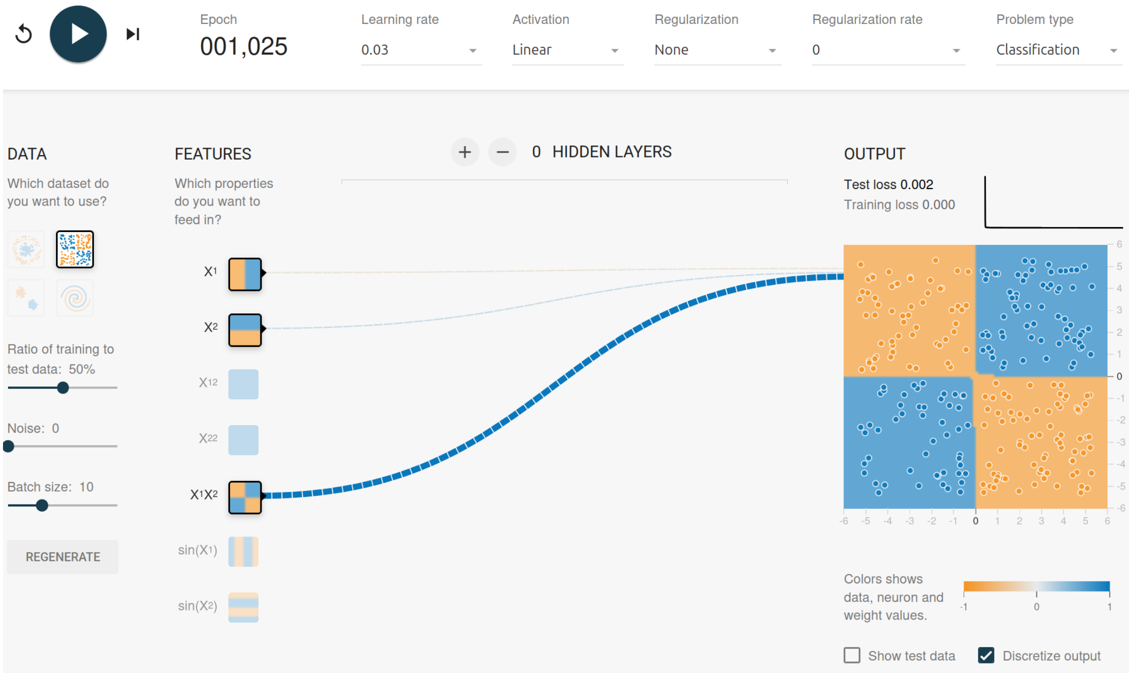

Figure 2: A linear model with no hidden layers using x_1, x_2 and x_1x_2 as features.

Now, return the features to just (x1, x2) 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)? 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)

The smallest neural network that I found able to fit the training data "perfectly" with only x_1 and x_2 as features used the ReLU activation function. It consisted of a single hidden layer with 4 neurons. The corresponding test loss is 0.002 after 1069 epochs. The result is shown in Figure 3.

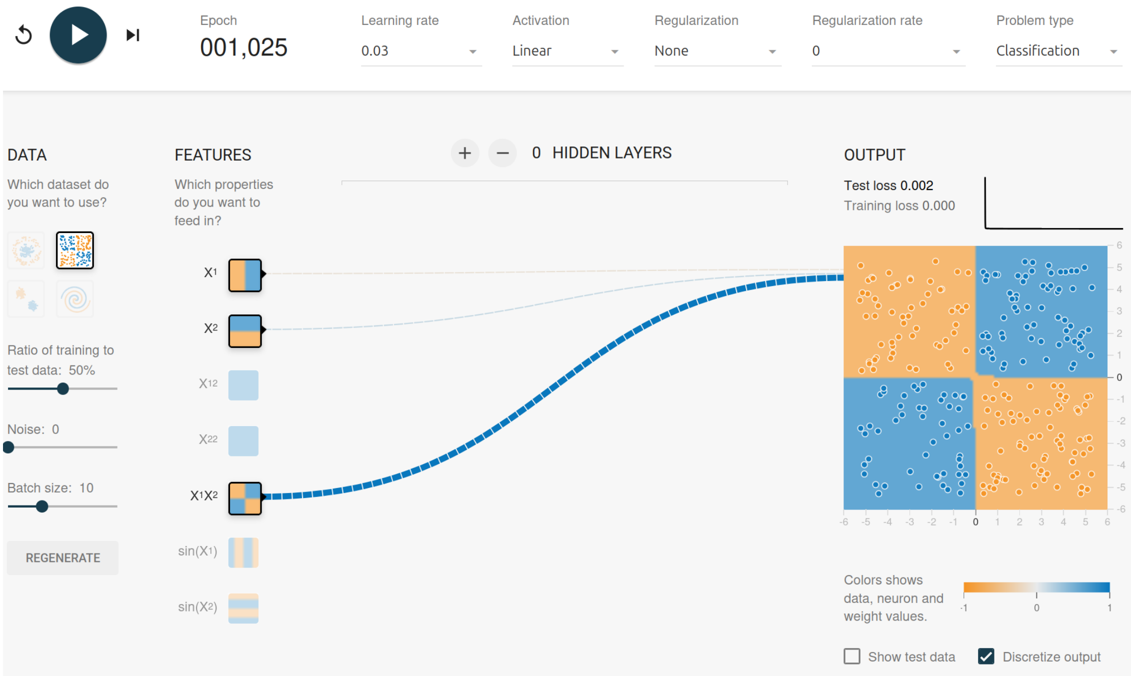

Figure 3: The smallest neural network that I found able to fit the training data "perfectly". It uses the ReLU activation funciton and a single hidden layer with 4 neurons. The corresponding test loss is 0.002 after 1069 epochs.

Problem B

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 (x1, x2), 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).

A low bias/high vairance model typically fits the traning data very well, but is not so robust with unkown test data. This is typically due to overfitting with too many parameters. On the contrary, a high bias/low variance model usually fits the training and test data equally poorly. This can be due to a model that is not complex enough to capture the trend of the data.

With complete freedom, a neural network using the ReLU activation and $x_1, x_2, x_1^2, x_2^2, x_1x_2, \sin(x_1), \sin(x_2)$ as features was used. The final model contained 2 hidden layers each with 8 neurons as shown in Figure 4. After 792 epochs, a training and test loss of 0.002 and 0.031 was obtained, respectfully. Thus, it can be said that this is a high varience/low bias model.

The neural network constrained to using only x_1, x_2 as features is shown in Figure 5. It consists of 3 hidden layers each with 8 neurons. After 822 epochs, a the training and test losses were 0.001 and 0.029, respectfully. Again, this points to the model being a low bias/high vareiance model.

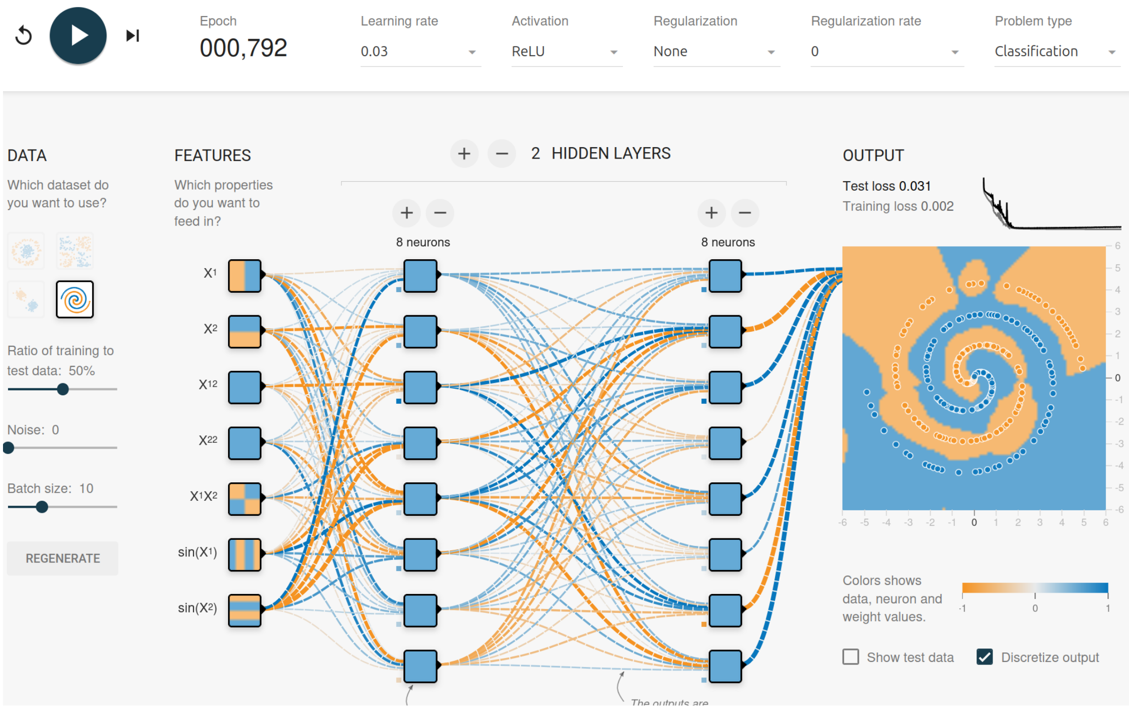

Figure 4: The neural network obtained with complete freedom.

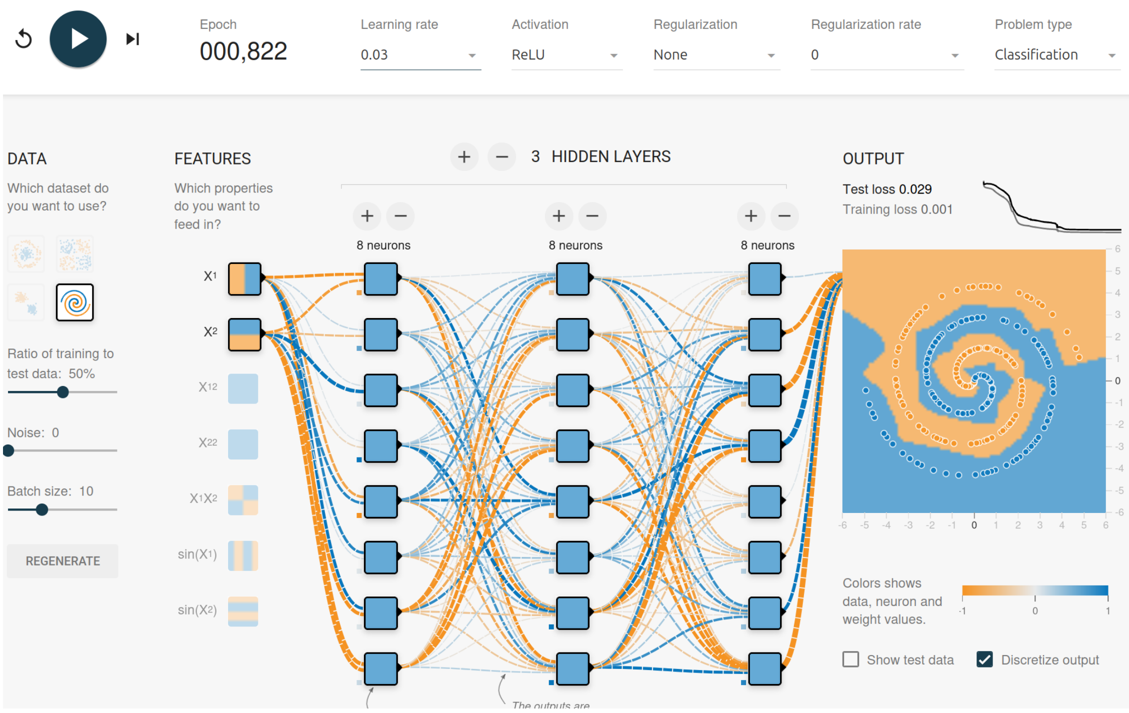

Figure 5: The neural network obtained with the features constrained to x_1 and x_2.