Assignment # 4: Neural Networks
Course: CAP 6610 Applied Machine Learning
Professor: Dr. Oge Marques
Student Number: Z23596812
Student Name: Renee Raven
Term: Fall 2022

Explore the datasets in the Tensorflow playground (Images below text)

The Tensorflow playground is a fun way to gain an appreciation for the ways subtle adjustments of batch size, learning rate, or proportion of training data to test data affect the performance of the model. The presence of hidden layers and the number of nodes also influence the effectiveness, but that doesn't mean more layers/nodes are always improve the model. In fact, in a couple situations (spiral and multi-gaussian) adding more layers degraded the model's performance.

The distribution of the data guides the choice of feature input (linear, multiple, conic, sine, and various combinations). The website offers four choices of classification models (circle, XOR, gaussian, spiral) and two choices for regression models (plane and multi-gaussian). Working with the circle model demonstrated the gracefulness of the conic function for modelling the data. As expected, increasing batch size decreased the performance of the model. The model may miss the small changes of the dataset when the batch is too large.

The XOR model only needed the right feature input to perform well without hidden layers. As expected, when the noise of the model increased, the performance decreased. The gaussian model, with a more complex distribution, required a hidden layer and a Tanh activation function to reach a test loss of 0.014. However, when the activation function was changed to Sigmoid, the model lost performance. I even added 2 hidden layers, but that did not improve the loss functions (test loss of 0.029 after 3 times the number of epochs).

The spiral model is a true challenge. My first somewhat haphazard attempt with the ReLU activation function produced a chaotic mess with a horrible test loss of 0.029. Lowering the learning rate, changing the activation function to Tanh, pruning to 2 hidden layers from 4, and increasing the proportion of training data use improved the model remarkably (test loss of 0.014).

Just for fun, I applied the spiral settings to a plane regression data distribution and the multi-gaussian regression data distribution. Proving that simpler is sometimes better, once I removed the hidden layers and selected the appropriate feature input, the plane model was perfect after only 60 epochs. It took some tinkering to improve the multi-gaussian model (increased training proportion, change activation function to ReLU, and increasing the learning rate), but the best performance I could wrangle was a test loss of 0.030

I decided to peek at the Q-learning example from the recommended tools because I am currently researching Q-learning and its applications to medicine for another course. This was a great demo of how the purple agent with 9 sensors (referred to as eyes) navigates their environment and eventually "learns." After about 10 minutes, the agent is able to avoid the green things (poison) and walls while gobbling up the red things (which reward the agent).

This was a fun, informative assignment. Tensorflow playground could certainly be a useful tool to use if a function falls into one of the basic types (circle XOR, gaussian, or spiral). One could visualize the effect of various settings and adjustments to help guide adjustments to an algorithm. The other recommended tools were also interesting to explore.

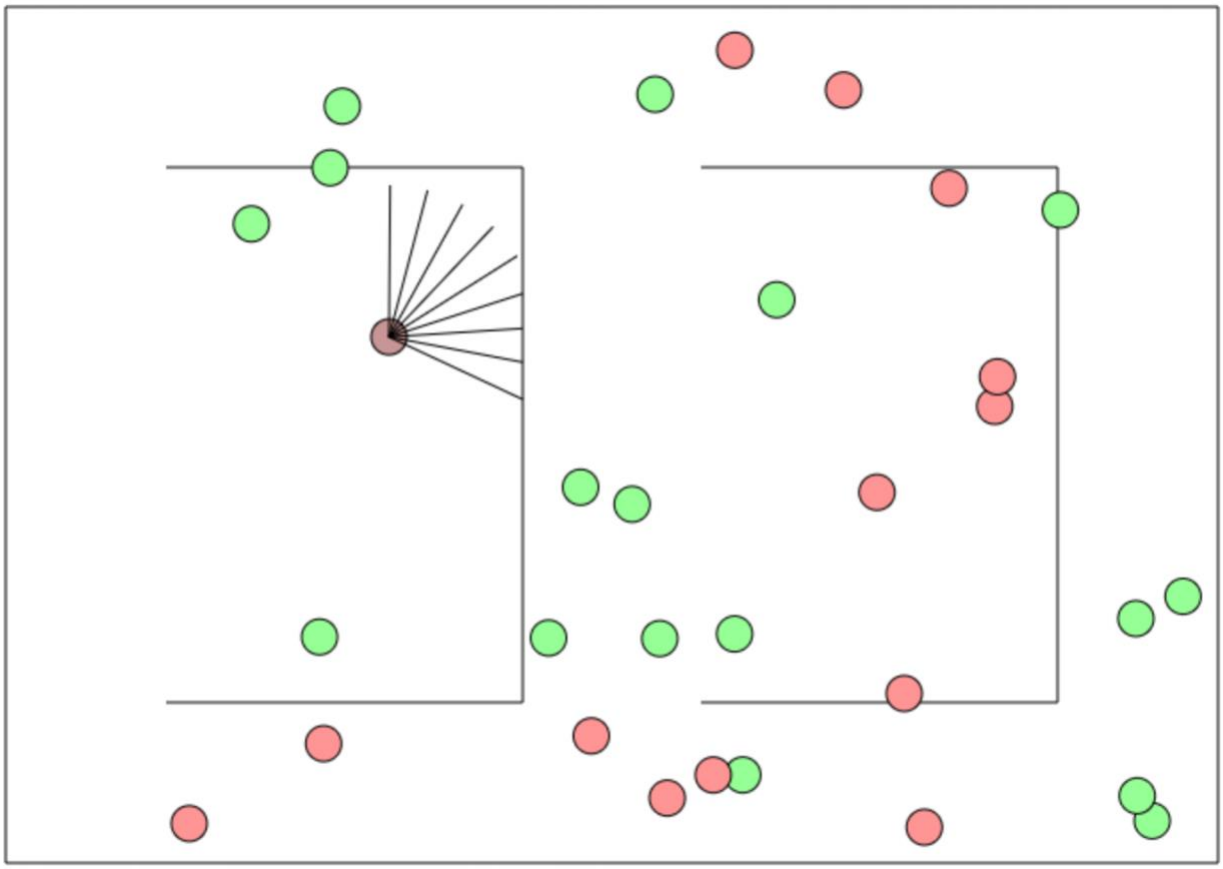

Figure 1. Demonstration of the 9 eyed agent interacting with their environment using the Q Learning algorithm

Circle (classification)

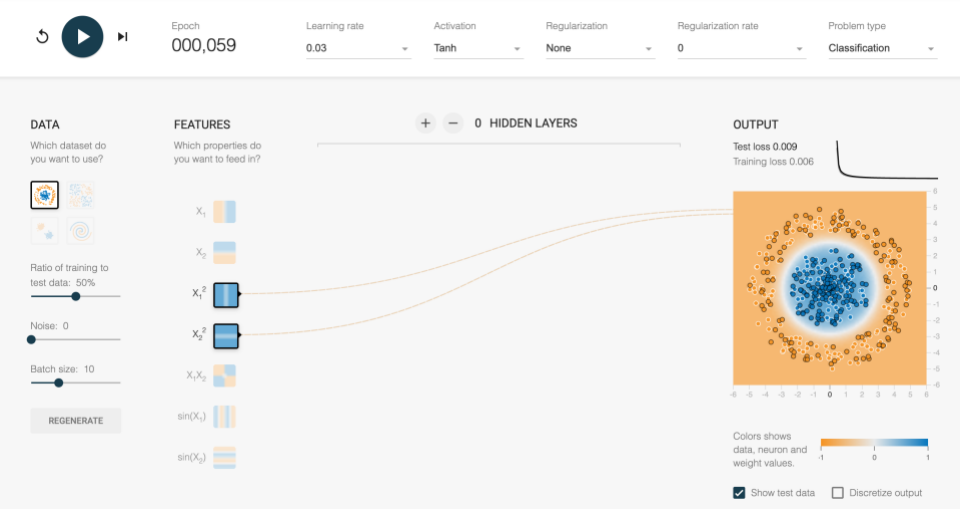

Figure 2. Using the conic circle equation to generate a great model in relatively few epochs.

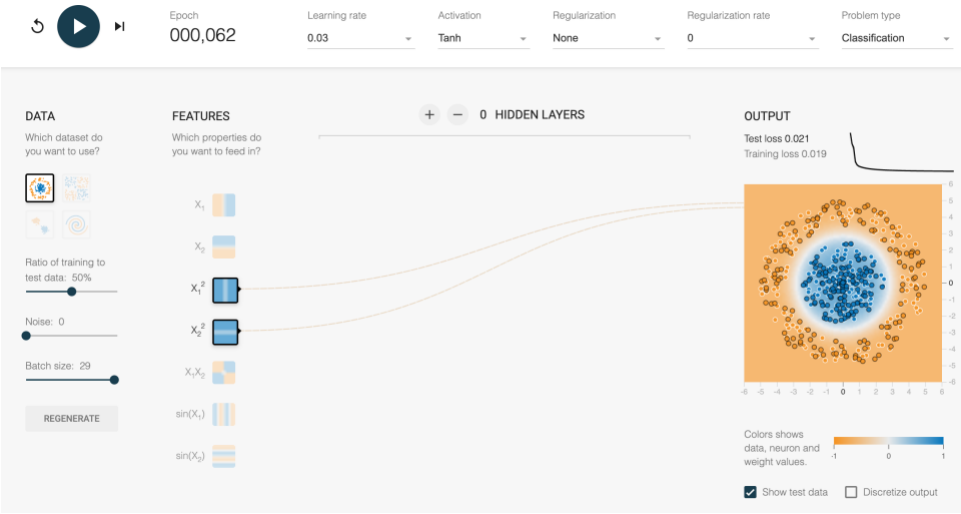

Figure 3. Using conic circle and increasing the batch size

XOR (classification)

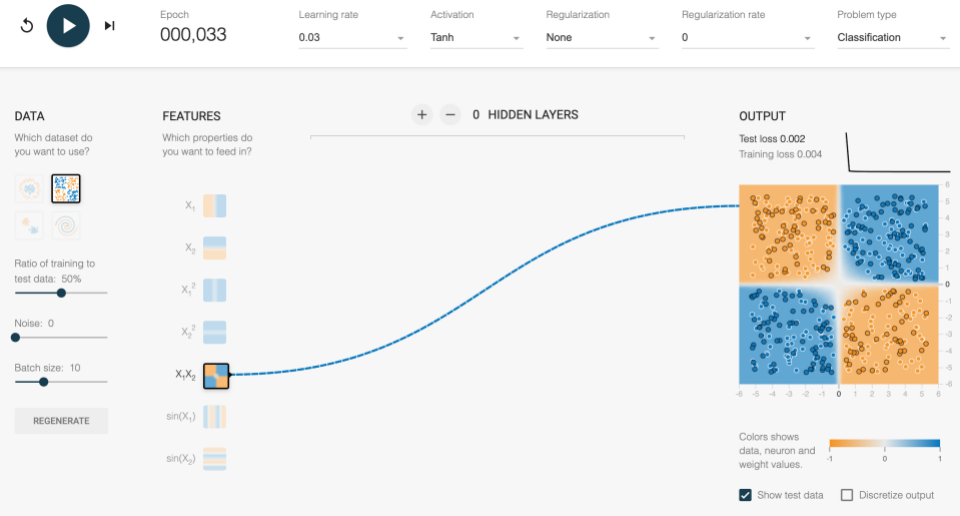

Figure 4. Simple solution to the XOR problem using multiplication an no hidden layers

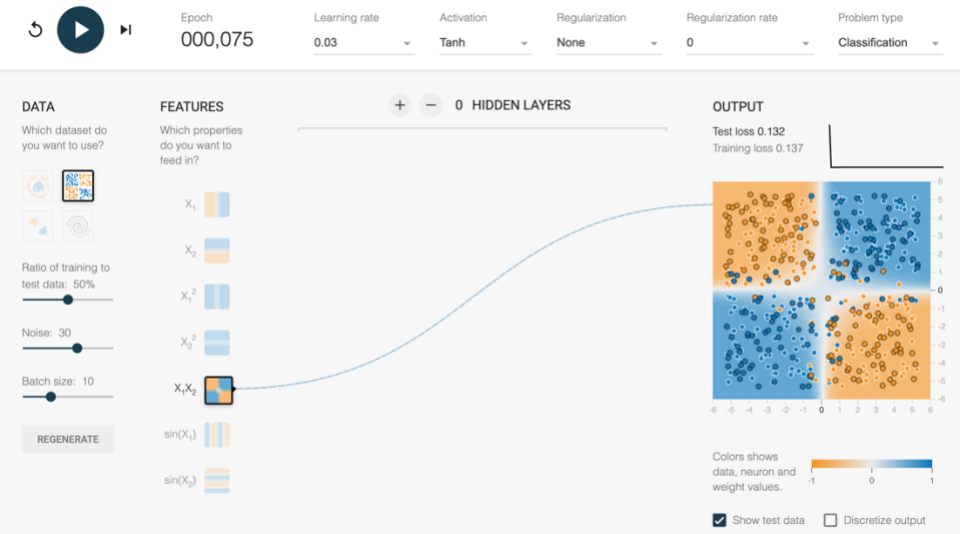

Figure 5. XOR with increased noise demonstrates poorer performance

Gaussian (classification)

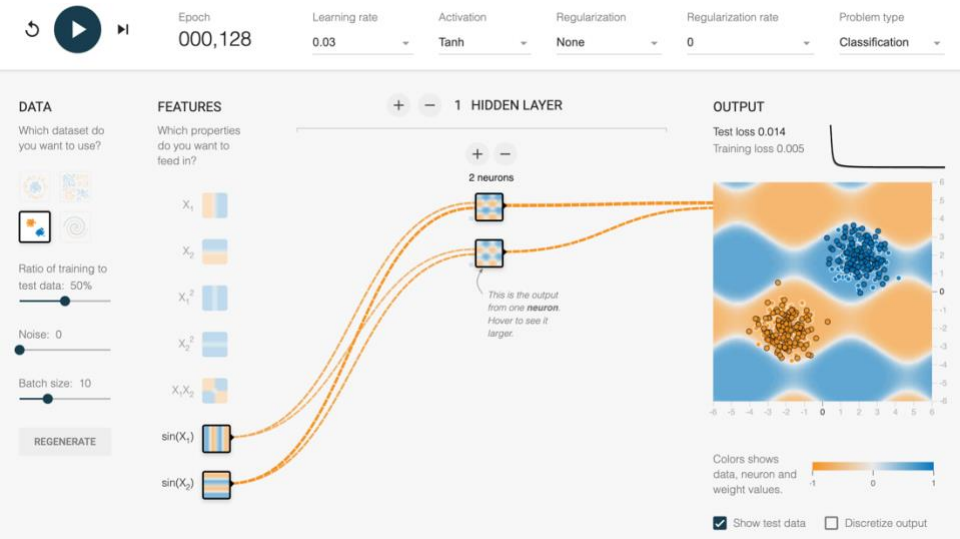

Figure 6. Gaussian solution using sine functions input and the Tanh activation function

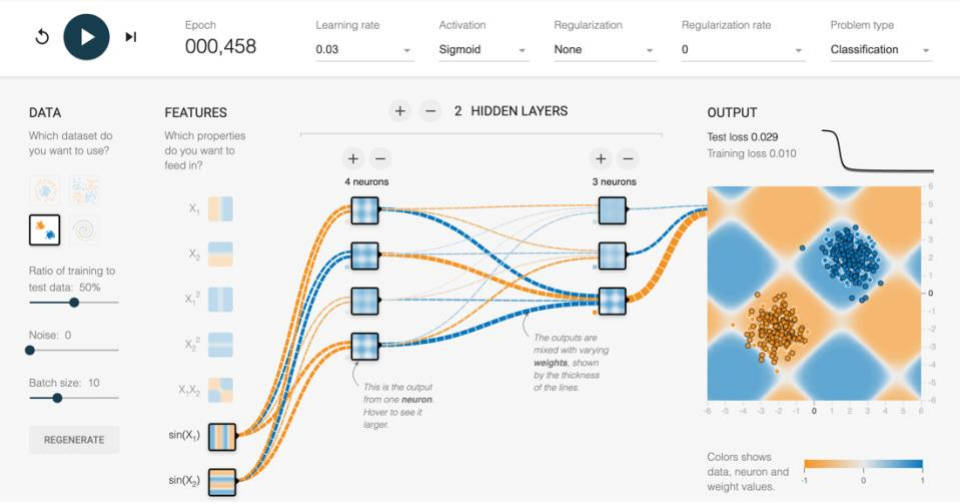

Figure 7. Gaussian model using Sigmoid activation and 2 hidden layers

Spiral (classification)

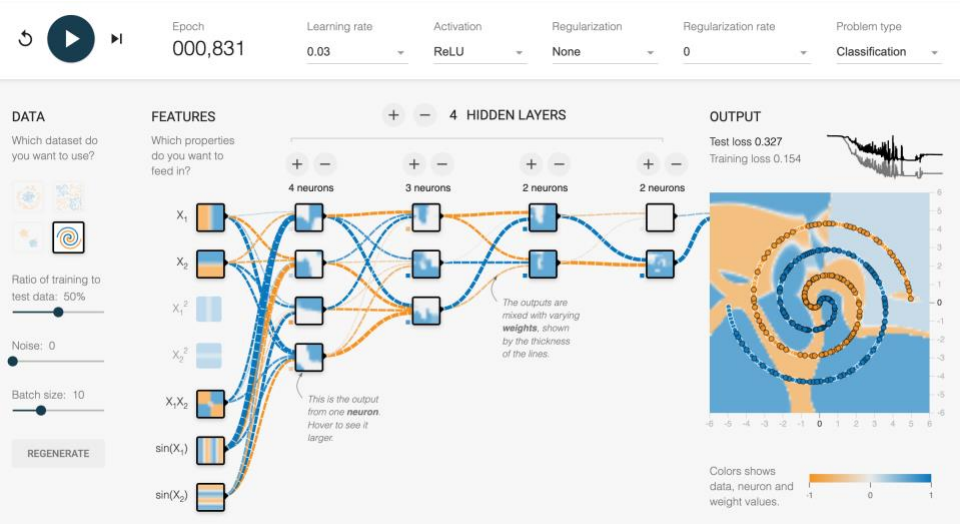

Figure 8. Poor modelling of a spiral, even with 3 hidden layers

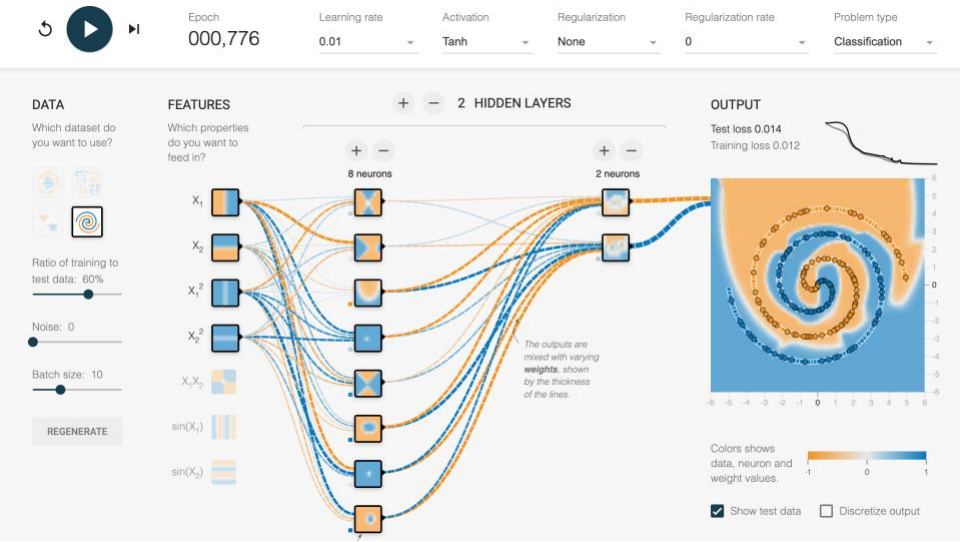

Figure 9. Spiral model with 2 hidden layers, smaller learning rate, and altered data ratio

Plane (regression)

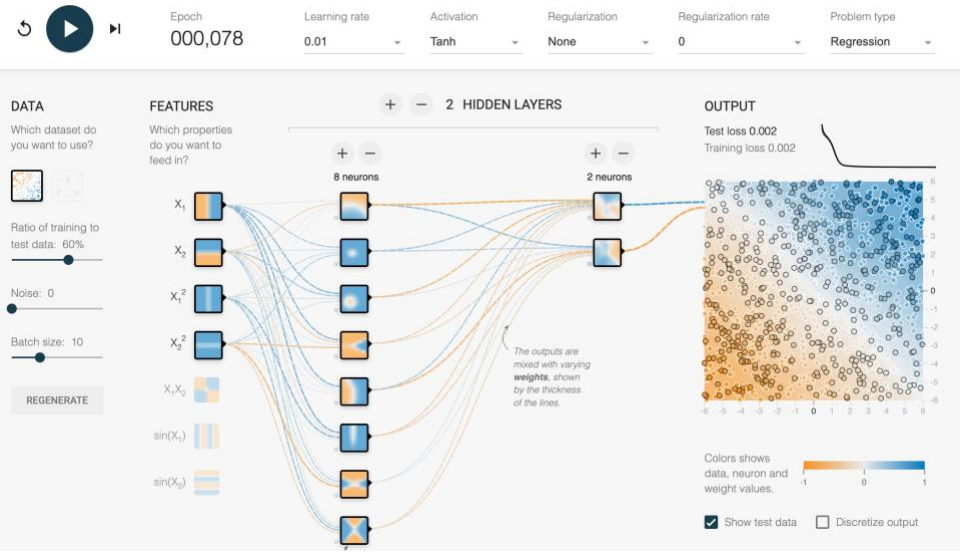

Figure 10. Spiral settings applied to plane regression

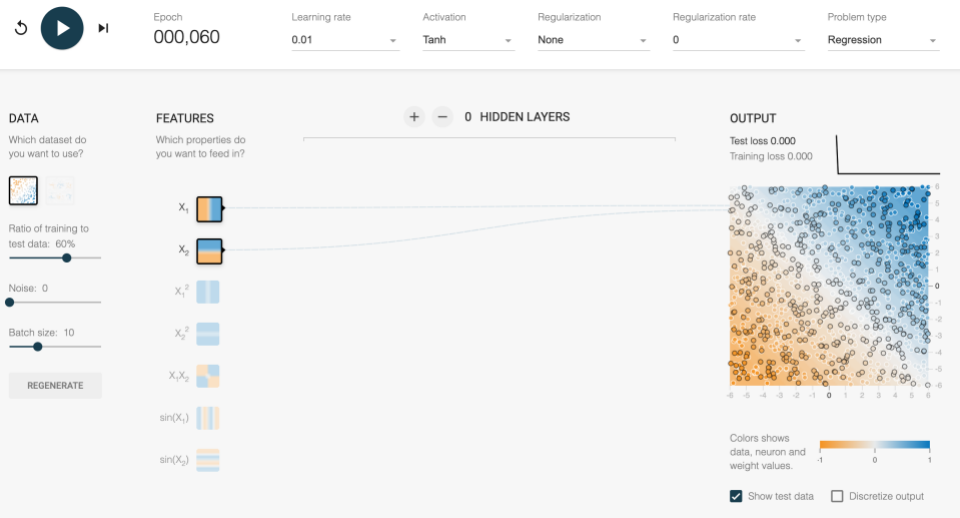

Figure 11. A much more efficient model for a plane regression

Multi-gaussian (regression)

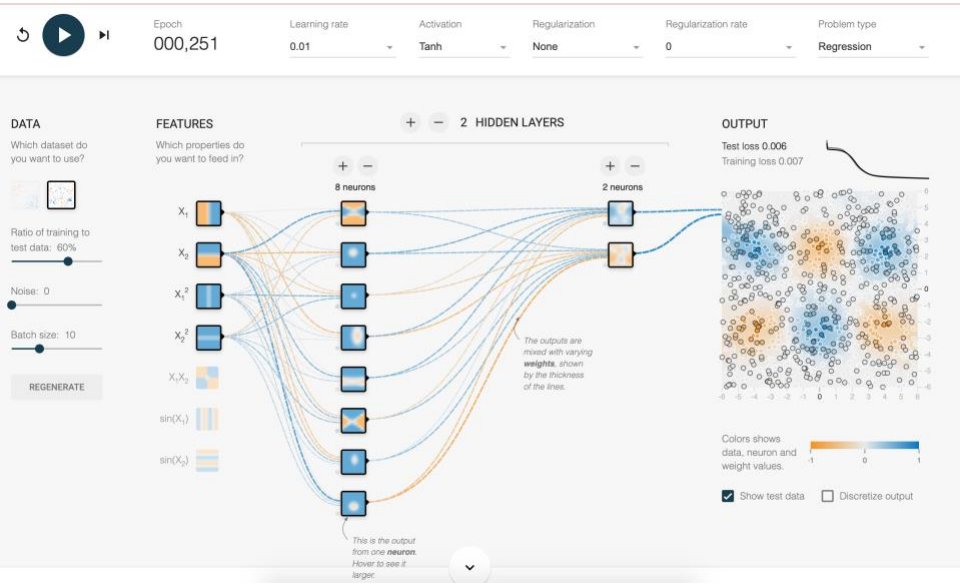

Figure 12. Spiral settings applied to multi-gaussian regression model

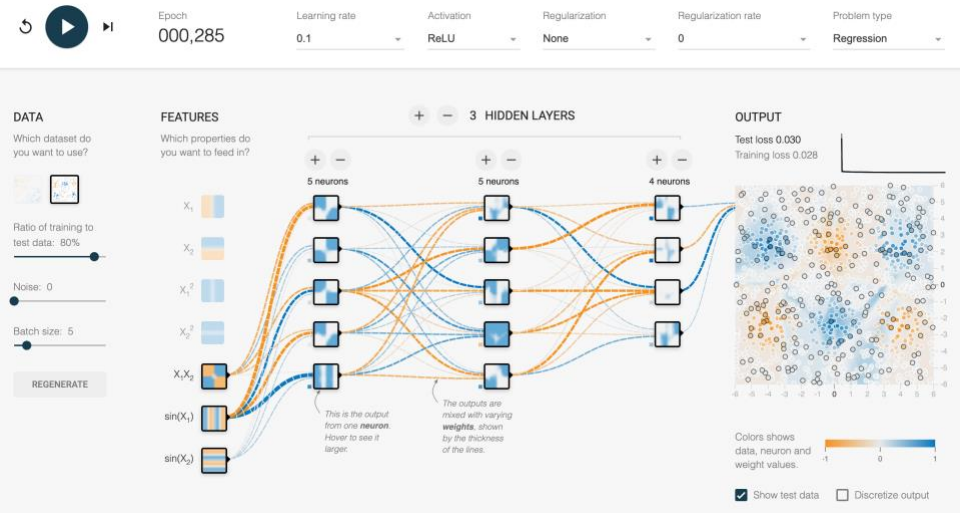

Figure 13. Multi-gaussian modelling with adjustments to batch size and learning rate