

Task1:

Play with the web app at this [link](#).

Explain what happens when we do matrix multiplication.

Yes, I know I basically already asked you to do this in the notes. Don't copy and paste your answer from before; honestly do the exercise again.

Matrix multiplication combines two sets of information to show how they interact. You take a row from the first matrix and a column from the second matrix. Then, you multiply each pair of numbers in matching positions (first with first, second with second, etc.) and add them up. That total goes into the new matrix. You do this for every row and column combo. The result shows how each row of the first matrix affects each column of the second one.

Task2:

Look at code for any neural network, such as my notes, or any other source.

1. Point out where in the code the neural network is multiplying matrices.
2. Conceptually, what part of a neural network is that matrix multiplication that you pointed out? In other words, what does matrix multiplication do for a neural network?

In neural network code, matrix multiplication usually happens during the forward pass. For example, at `y_pred = linear(X_train)`.

This step is where the neural network "learns" patterns by multiplying inputs by weights to compute the signals passed to the next layer. It's how the network combines data to make decisions.

Task3:

Go to the [TensorFlow Playground website](#).

Explain what part of the neural network visualization is matrix multiplication.

At the curvy lines/nodes is where the matrix multiplication actually happens. We can even see it visually with the lines converging.

Task4:

Go to the [TensorFlow Playground website](#).

Set up the scenario with the following characteristics:

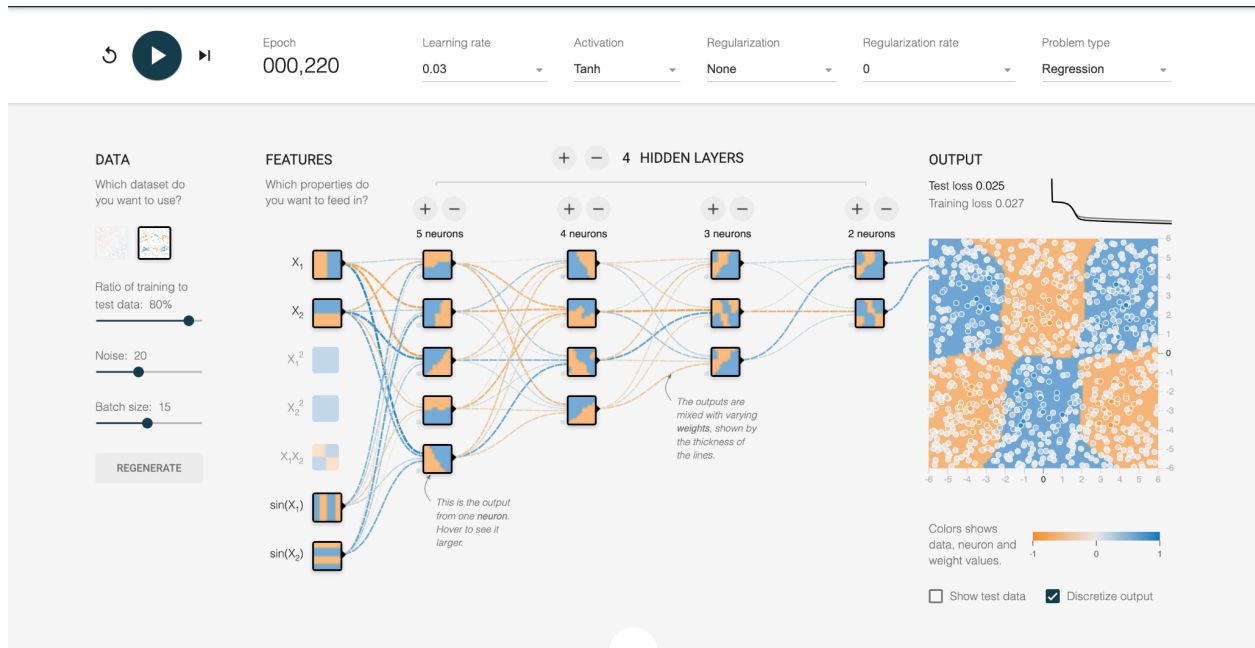
- problem type: regression
- Data: multi gaussian
- Ratio: 80% training
- Noise: 20
- Batch size: 15
- Discretize output: start with it checked on

You may turn discretize output on or off once you get an idea of what to do, so you can see what each node is doing.

Goal: get your test loss to under **0.03**.

You may change any of the other parameters that I didn't explicitly tell you to set up. Remember to use the back/restart button in between model adjustments.

Your answer will be a link to the model you created. How do you get that link? Once you've achieved the appropriate test loss, copy and paste the url.



[Task 4 Link](#)

Task5:

Go to the [TensorFlow Playground website](#).

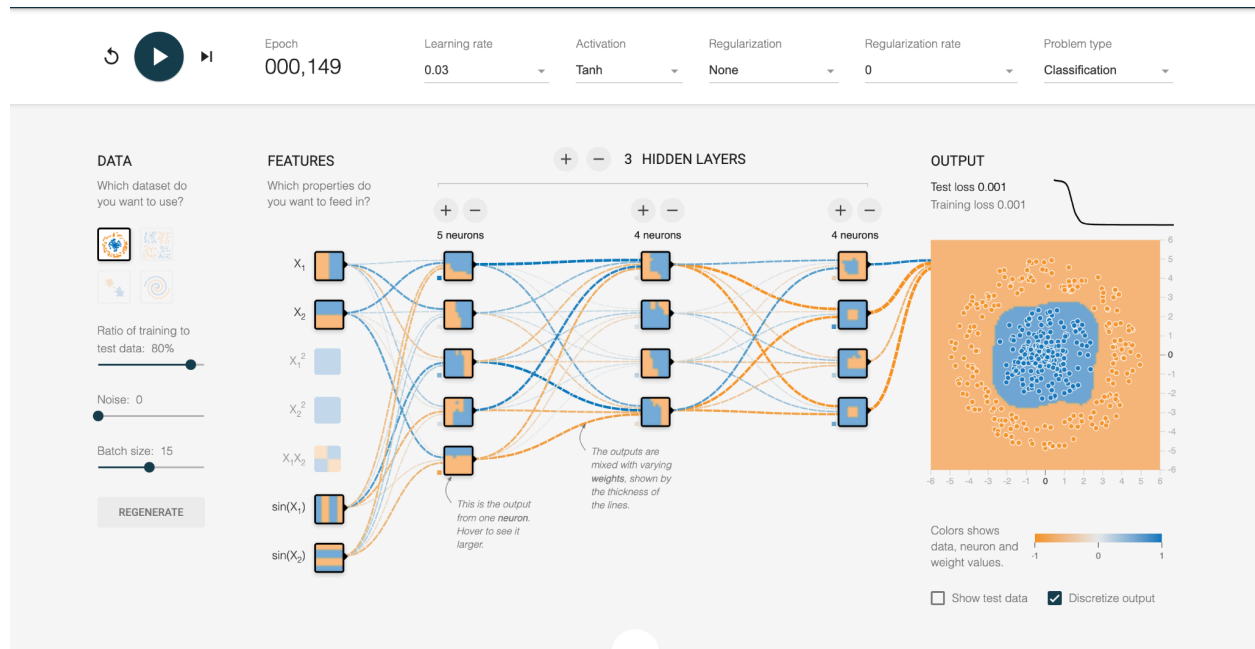
Set up the scenario with the following characteristics:

- problem type: classification
- Data: circle
- Ratio: 80% training
- Noise: 0
- Batch size: 15

Goal: get your test loss to under **0.005**.

You may change any of the other parameters that I didn't explicitly tell you to set up. Remember to use the back/restart button in between model adjustments.

Your answer will be a link to the model you created. How do you get that link? Once you've achieved the appropriate test loss, copy and paste the url.



[Task 5 Link](#)

Task6:

Go to the [TensorFlow Playground website](#).

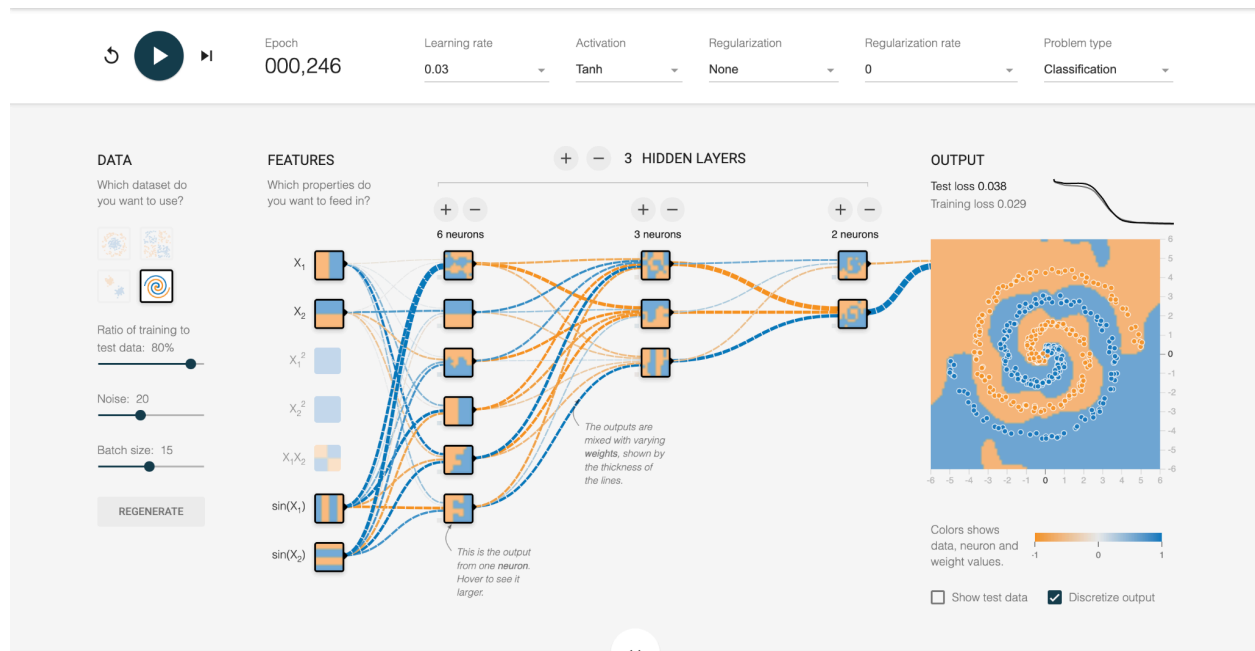
Set up the scenario with the following characteristics:

- problem type: classification
- Data: spiral
- Ratio: 80% training
- Noise: 20
- Batch size: 15

Goal: get your test loss to under **0.05**.

You may change any of the other parameters that I didn't explicitly tell you to set up. Remember to use the back/restart button in between model adjustments.

Your answer will be a link to the model you created. How do you get that link? Once you've achieved the appropriate test loss, copy and paste the url.



[Task 6 Link](#)