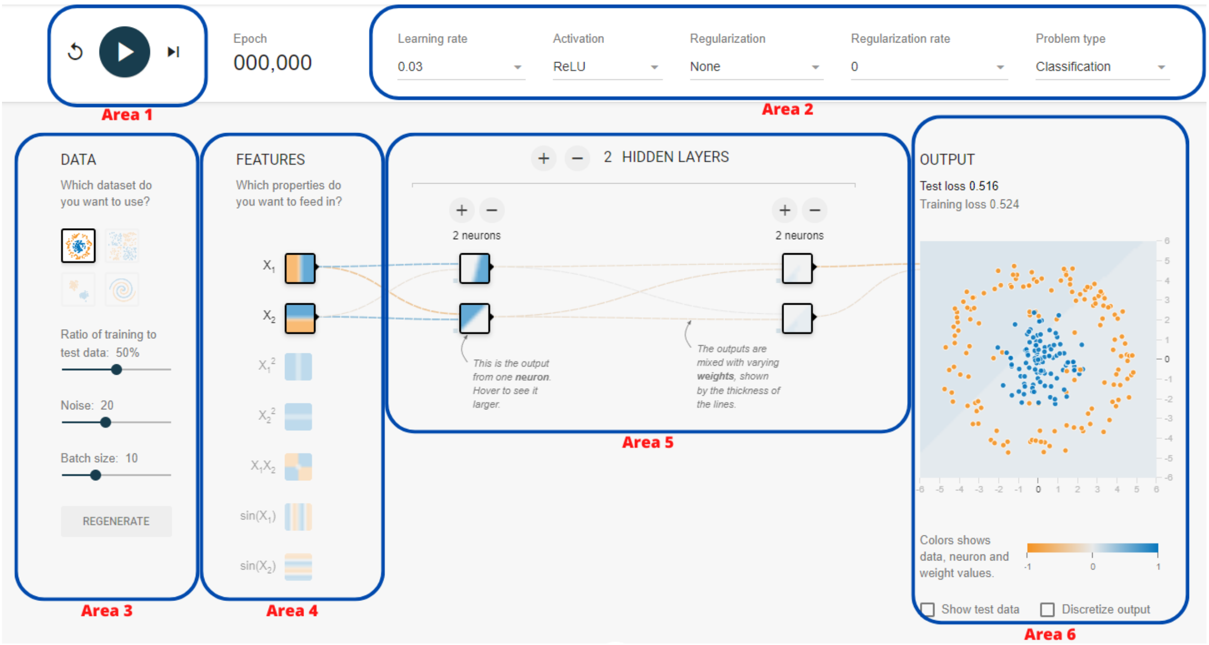
Tensorflow Playground

Exercises

TensorFlow Playground is an interactive web application that allows you to visualize and experiment with various neural network models. It is designed to help beginners understand the basics of neural networks and machine learning.

TensorFlow Playground website : https://playground.tensorflow.org/



Epoch 000,000

Learning rate 0.03

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 20

Batch size: 10

FEATURES

Which properties do you want to feed in?

2 HIDDEN LAYERS

2 neurons

2 neurons

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

Test loss 0.516

Training loss 0.524

Colors shows data, neuron and weight values.

- **Area 1 -** This area allows you to run the neural network. Don't forget to stop the training when it does not improve anymore.
- **Area 2 -** This area allows you to control some of the parameters of the neural network such as the learning rate, the activation functions of the hidden neurons, the task (regression or classification) and the regularization.
- **Area 3 -** This area allows you to control the training data used in the neural network. You can use 4 different datasets and modify the level of noise for each of them. Additionally, you can control the batch size.
- **Area 4 -** This area allows you to control the features. The top features are the most simplest and the bottom ones are the most "complex".
- **Area 5 -** This area allows you to control the number of hidden layers and their associated number of neurons.
- **Area 6 -** This area allows you to look at the result of the training. You have access to the loss of the training and the testing sets as well as an image representation of the output.

Exercise I - *Neurons & Layers*

Part I

1. Use the **circle dataset** and the default parameters.
2. Configure a neural network with **1 hidden layer** and **2 neurons**.
3. Run the training for a few epochs.
4. What happened ?
5. Re-do the experiment with a 3rd neuron.
6. What happened ? Pay attention to the neuron representation.

Part II

1. Use the **circle dataset** and the default parameters.
2. Configure a neural network with **2 hidden layers** and **2 neurons**.
3. Run the training for a few epochs.
4. What happened ? Is it better than 1 hidden layer with 2 neurons ?
5. Re-do the experiment with a 3rd neuron on both hidden layers.
6. What happened ? Are the 1st hidden layer neuron representations different from the ones of the 2nd hidden layer ? If so, why ?

Part III

1. Use the **exclusive or dataset** and the default parameters.
2. Try to get a test loss smaller than **0.01** !
   a. Find the smallest architecture possible ! (*i.e. less layers and less neurons)*
   b. You can only modify the elements in area 5.
3. When you have found it, think about this neural network configuration. Does it seem logical ?

Exercise II - *Activation Function*

1. Use the **circle dataset** and the default parameters.
2. Configure a neural network with **2 hidden layers** and **3 neurons**.
3. Select the Sigmoid activation function and run the training. Wait until the loss is stable and the instances are correctly classified.
4. Re-do the experiment for the tanH activation function.
5. Finally, re-do the experiment for the ReLU activation function.
6. What happened ? Did you find some differences ?

Exercise III - *Learning Rate*

1. Use the **exclusive or dataset** and the default parameters.
2. Re-use the best smallest configuration found at Exercise I part III and run the training with a learning rate of:
   a. 0.003
   b. 0.03
   c. 0.1
   d. 1
3. What happened ?

Exercise IV - *Features*

Part I

1. Use the **spiral dataset** with
   a. a Noise parameter (Area 3) set at 25.
   b. a ratio of training parameter (Area 3) set at 50.
   c. a batch size parameter (Area 3) set at 10.
2. Try to get a test loss smaller than **0.025** !
   a. The smaller the architecture, the better ! (*i.e. less layers and less neurons)*
   b. You can't change anything in area 4 (the features) and area 3 (the data).
3. What is your best final configuration ? Why do you think this one worked better than the others ?

Part II

1. Re-do the experiment of Part I with the same rules but now you can choose up to 6 features in area 4.
2. Try to get a test loss smaller than **0.02** !
   a. The smaller the architecture, the better ! (*i.e. less layers and less neurons)*
   b. You can't change anything in area 3 (the data).

Exercise V - *Overfitting*

1. Use the **circle dataset** with
   a. a Noise parameter (Area 3) set at 50.
   b. a ratio of training parameter (Area 3) set at 50.
   c. a batch size parameter (Area 3) set at 10.
2. Create a neural network with 6 hidden layers containing 8 neurons each.
3. Run the training for a few hundreds of epochs (500-1000).
4. Pay attention to:
   a. The losses (test and training) and especially the curves
   b. The neuron representations
   c. The overall classification
5. What happened ?