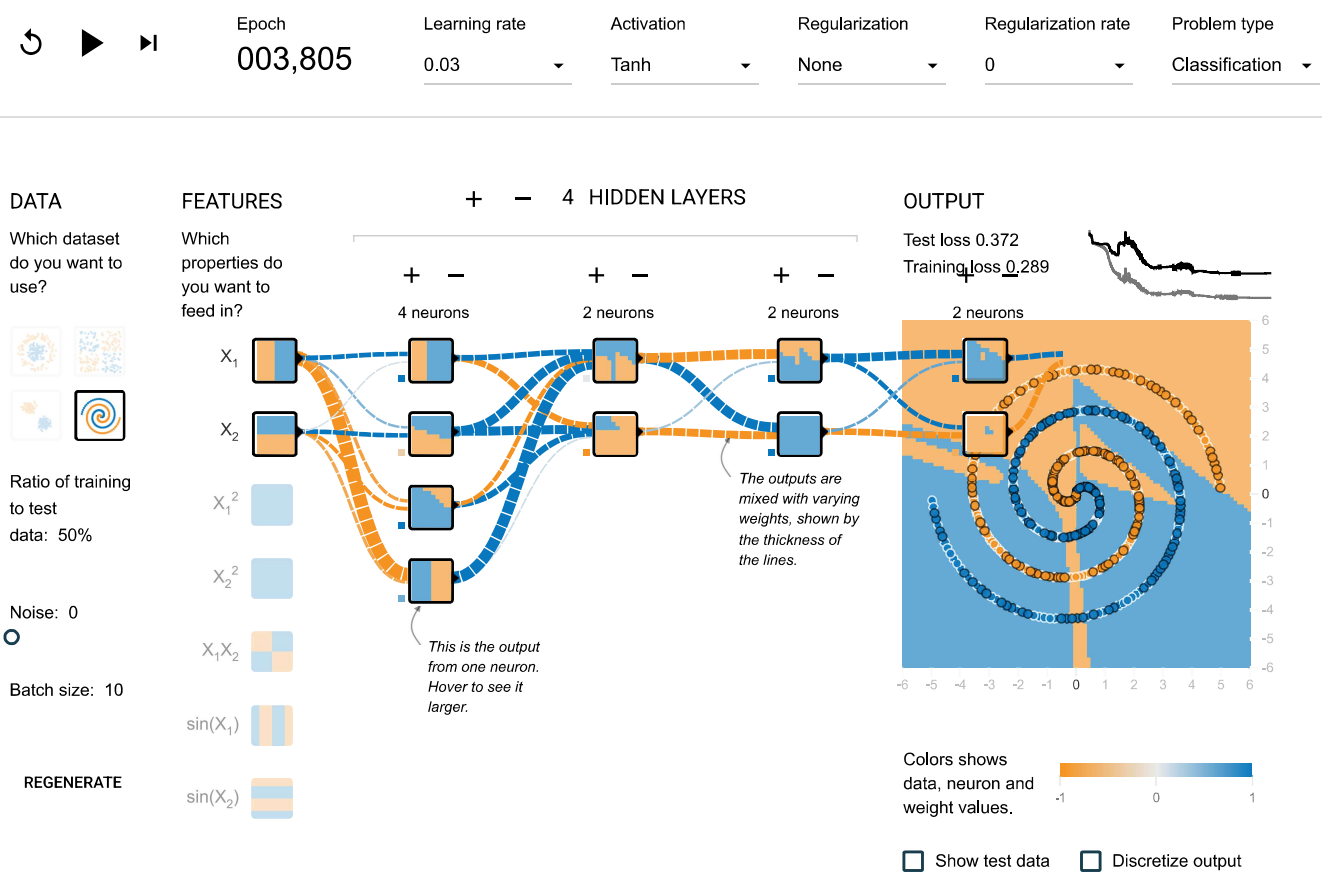
Tinker With a **Neural Network** in Your Browser. Don't Worry, You Can't Break It. We Promise.

(https://git



Um, What Is a Neural Network?

It's a technique for building a computer program that learns from data. It is based very loosely on how we think the human brain works. First, a collection of software "neurons" are created and connected together, allowing them to send messages to each other. Next, the network is asked to solve a problem, which it attempts to do over and over, each time strengthening the connections that lead to success and diminishing those that lead to failure. For a more detailed introduction to neural networks, Michael Nielsen's Neural Networks and Deep Learning (http://neuralnetworksanddeeplearning.com/index.html) is a good place to start. For a more technical overview, try Deep Learning (http://www.deeplearningbook.org/) by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

This Is Cool, Can I Repurpose It?

Please do! We've open sourced it on GitHub (https://github.com/tensorflow/playground) with the hope that it can make neural networks a little more accessible and easier to learn. You're free to use it in any way that follows our Apache License (https://github.com/tensorflow/playground/blob/master/LICENSE). And if you have any suggestions for additions or changes, please let us know (https://github.com/tensorflow/playground/issues).

We've also provided some controls below to enable you tailor the playground to a specific topic or lesson. Just choose which features you'd like to be visible below then save this link (https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.34769&showTestData= or refresh the page.

- ☐ Show test data
- ☐ Discretize output
- ☐ Play button
- ☐ Step button
- ☐ Reset button
- ☐ Learning rate
- ☐ Activation
- ☐ Regularization
- ☐ Regularization rate
- ☐ Problem type
- ☐ Which dataset
- ☐ Ratio train data
- ☐ Noise level
- ☐ Batch size
- ☐ # of hidden layers

What Do All the Colors Mean?

Orange and blue are used throughout the visualization in slightly different ways, but in general orange shows negative values while blue shows positive values.

The data points (represented by small circles) are initially colored orange or blue, which correspond to positive one and negative one.

In the hidden layers, the lines are colored by the weights of the connections between neurons. Blue shows a positive weight, which means the network is using that output of the neuron as given. An orange line shows that the network is assiging a negative weight.

In the output layer, the dots are colored orange or blue depending on their original values. The background color shows what the network is predicting for a particular area. The intensity of the color shows how confident that prediction is.

What Library Are You Using?

We wrote a tiny neural network library (https://github.com/tensorflow/playground/blob/master/src/nn.ts) that meets the demands of this educational visualization. For real-world applications, consider the TensorFlow (https://www.tensorflow.org/) library.

Credits

This was created by Daniel Smilkov and Shan Carter. This is a continuation of many people's previous work — most notably Andrej Karpathy's convnet.js demo (http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html) and Chris Olah's articles (http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/) about neural networks. Many thanks also to D. Sculley for help with the original idea and to Fernanda Viégas and Martin Wattenberg and the rest of the Big Picture (https://research.google.com/bigpicture/) and Google Brain (https://research.google.com/teams/brain/) teams for feedback and guidance.
