Machine Learning Techniques

A demo of ML Component Framework with Tensorflow Playground

Dr. Ashish Tendulkar

IIT Madras

A demo of Component Framework

- A short demonstration of the component framework.
- TensorFlow playground
  Link: https://playground.tensorflow.org/
- It is built in Javascript.

A demo of Component Framework

- A short demonstration of the component framework.
- TensorFlow playground
  Link: https://playground.tensorflow.org/
- It is built in Javascript.

A demo of Component Framework

- A short demonstration of the component framework.
- TensorFlow playground
  Link: https://playground.tensorflow.org/
- It is built in Javascript.

TensorFlow Playground

- It provides a way to visualize ML training.
- It allows changing different components like training data, model, loss function (to some extent) and parameters of the optimization algorithm.
- Evaluation protocol is fixed.

TensorFlow Playground

- It provides a way to visualize ML training.
- It allows changing different components like training data, model, loss function (to some extent) and parameters of the optimization algorithm.
- Evaluation protocol is fixed.

TensorFlow Playground

- It provides a way to visualize ML training.
- It allows changing different components like training data, model, loss function (to some extent) and parameters of the optimization algorithm.
- Evaluation protocol is fixed.

A tour of TensorFlow Playground

- The playground is mainly meant for experimenting with neural networks, but we will repurpose it for understanding various important concepts in ML.
- It will be used to highlight components of the ML framework and visually demonstrate how ML works.
- Play with it freely without fear of breaking anything down!

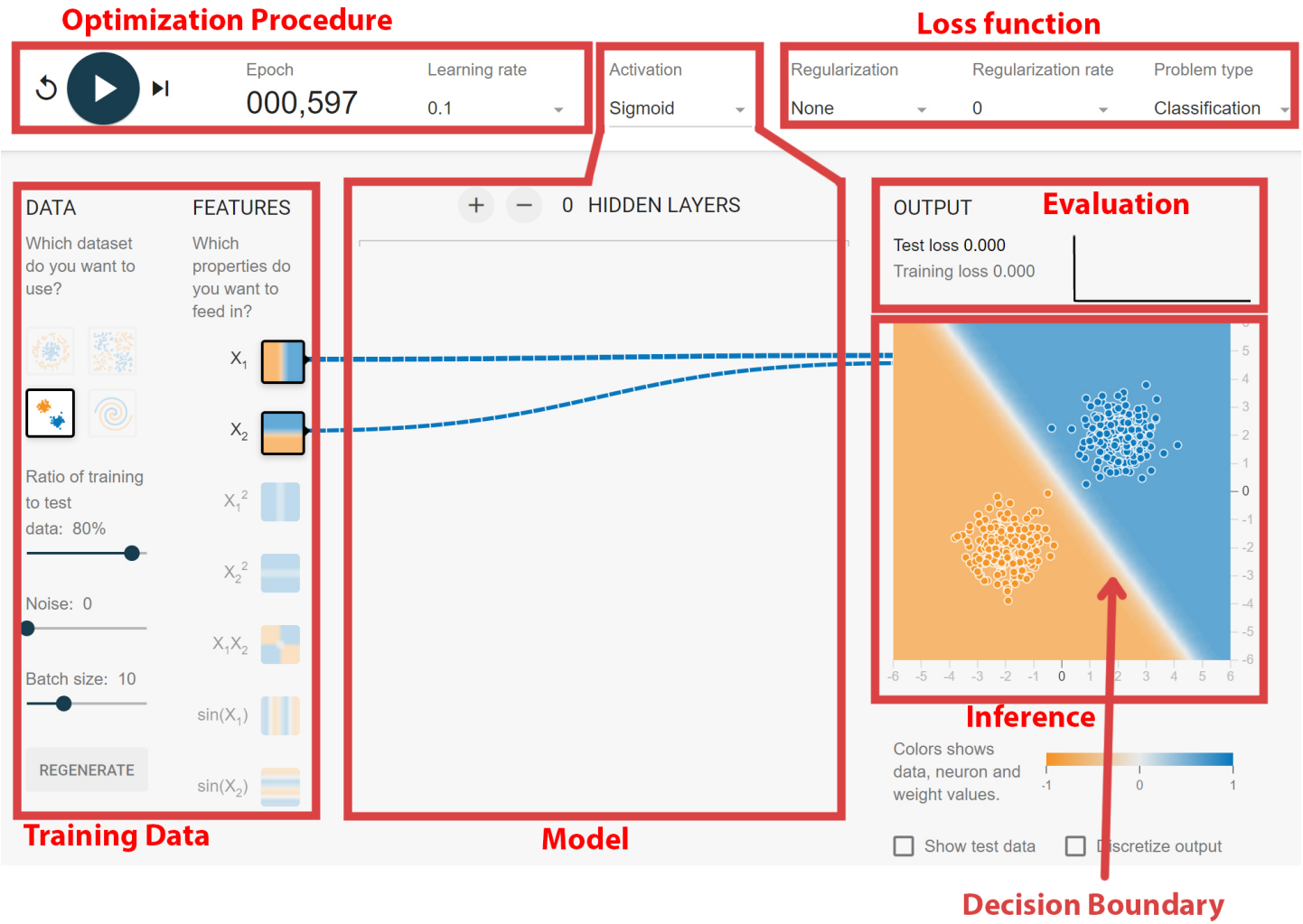
A tour of TensorFlow Playground

- The playground is mainly meant for experimenting with neural networks, but we will repurpose it for understanding various important concepts in ML.
- It will be used to highlight components of the ML framework and visually demonstrate how ML works.
- Play with it freely without fear of breaking anything down!

A tour of TensorFlow Playground

- The playground is mainly meant for experimenting with neural networks, but we will repurpose it for understanding various important concepts in ML.
- It will be used to highlight components of the ML framework and visually demonstrate how ML works.
- Play with it freely without fear of breaking anything down!

Overview: Tensorflow Playground



Training Data

- The most important ML component: data.
- No data, no ML.
- In TensorFlow playground, we have data with two classes: positive and negative classes.
- The examples from the positive class are represented with blue color and the ones from the negative class are represented with orange color.

Training Data

- The most important ML component: data.
- No data, no ML.
- In TensorFlow playground, we have data with two classes: positive and negative classes.
- The examples from the positive class are represented with blue color and the ones from the negative class are represented with orange color.

Training Data

- The most important ML component: data.
- No data, no ML.
- In TensorFlow playground, we have data with two classes: positive and negative classes.
- The examples from the positive class are represented with blue color and the ones from the negative class are represented with orange color.

Training Data

- The most important ML component: data.
- No data, no ML.
- In TensorFlow playground, we have data with two classes: positive and negative classes.
- The examples from the positive class are represented with blue color and the ones from the negative class are represented with orange color.

Training Data

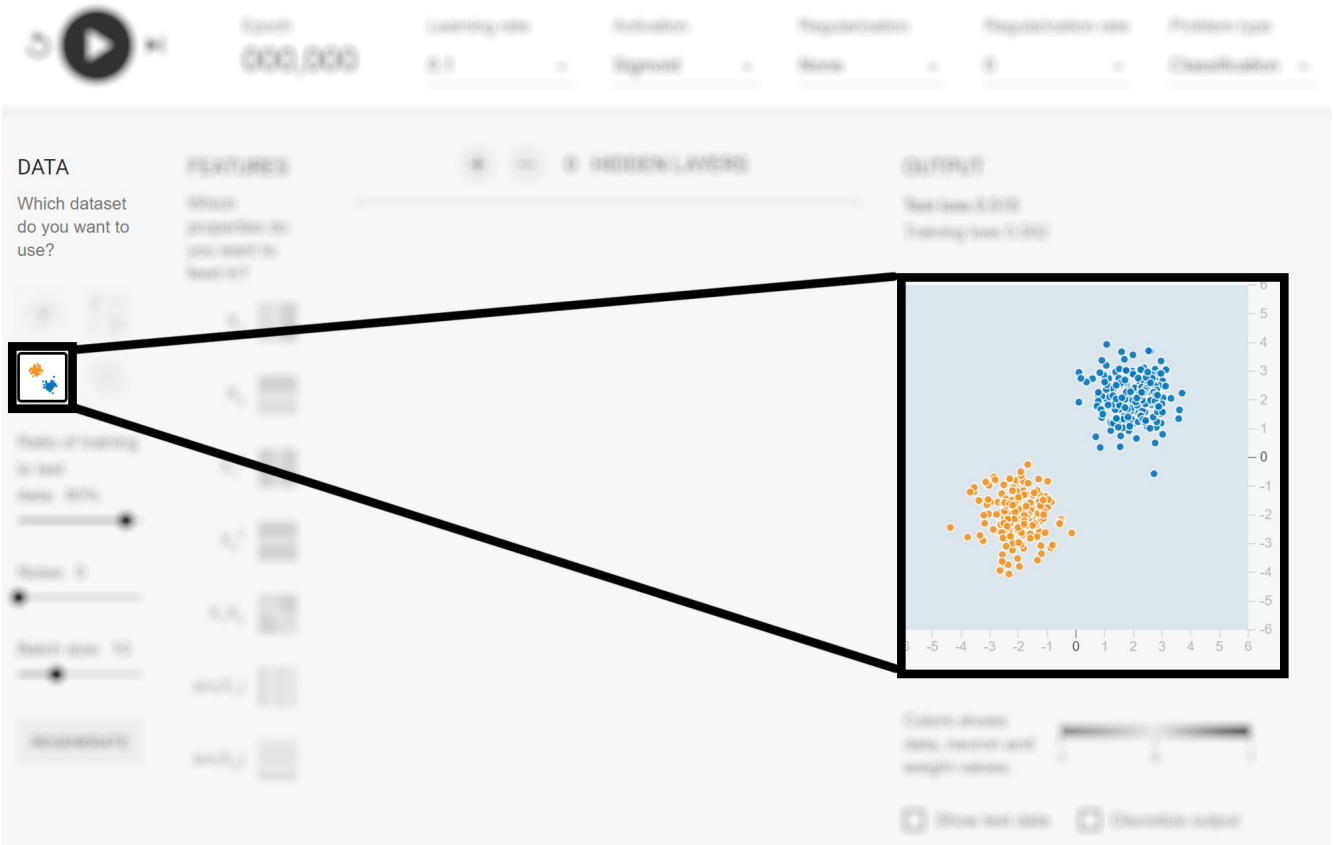
- The most important ML component: data.
- No data, no ML.
- In TensorFlow playground, we have data with two classes: positive and negative classes.
- The examples from the positive class are represented with blue color and the ones from the negative class are represented with orange color.

Data

The playground provides different permutations of training data.

Data

Linearly separable classes.



Data

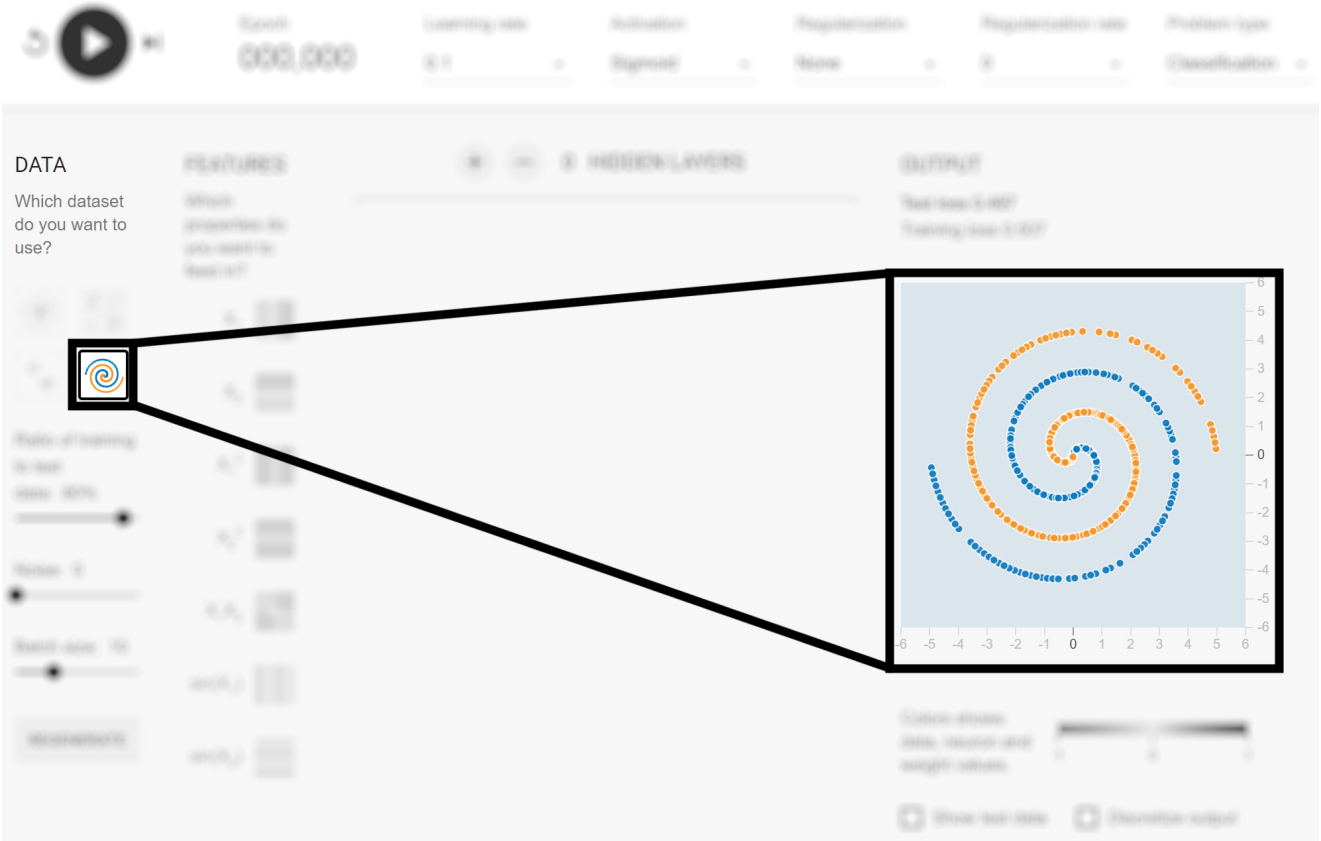
Classes separable by a circle.

Data

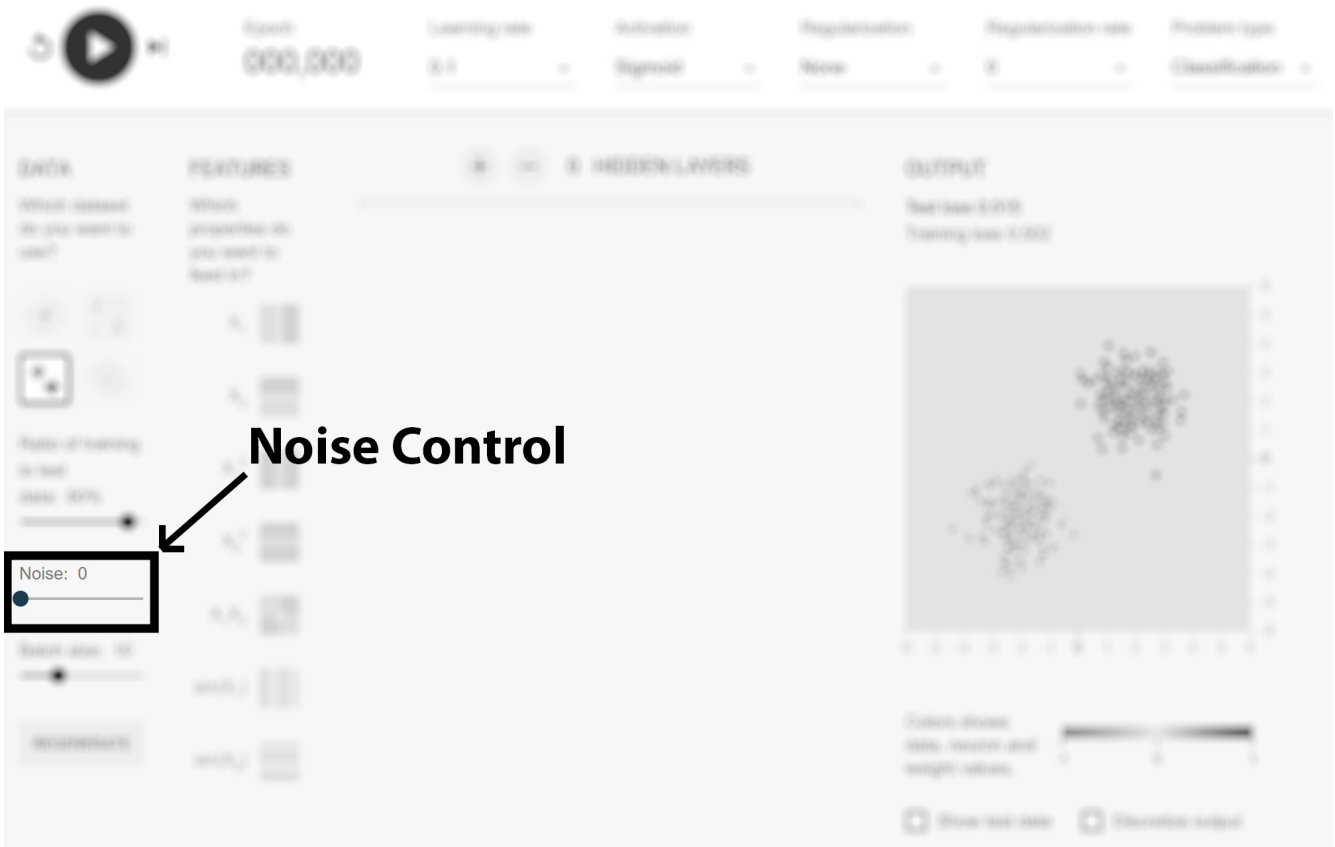
Classes arranged in an oblique manner - like x-or.

Data

Classes separable along a curvilinear path



Noise controller



Data

- Each data point has two features (x_1, x_2)
- Training set: 80%
- Test set: 20%

Model

- Logistic regression
- Train model to map features to classes.
- Learn weights of the features.

Model

- Logistic regression
- Train model to map features to classes.
- Learn weights of the features.

Model

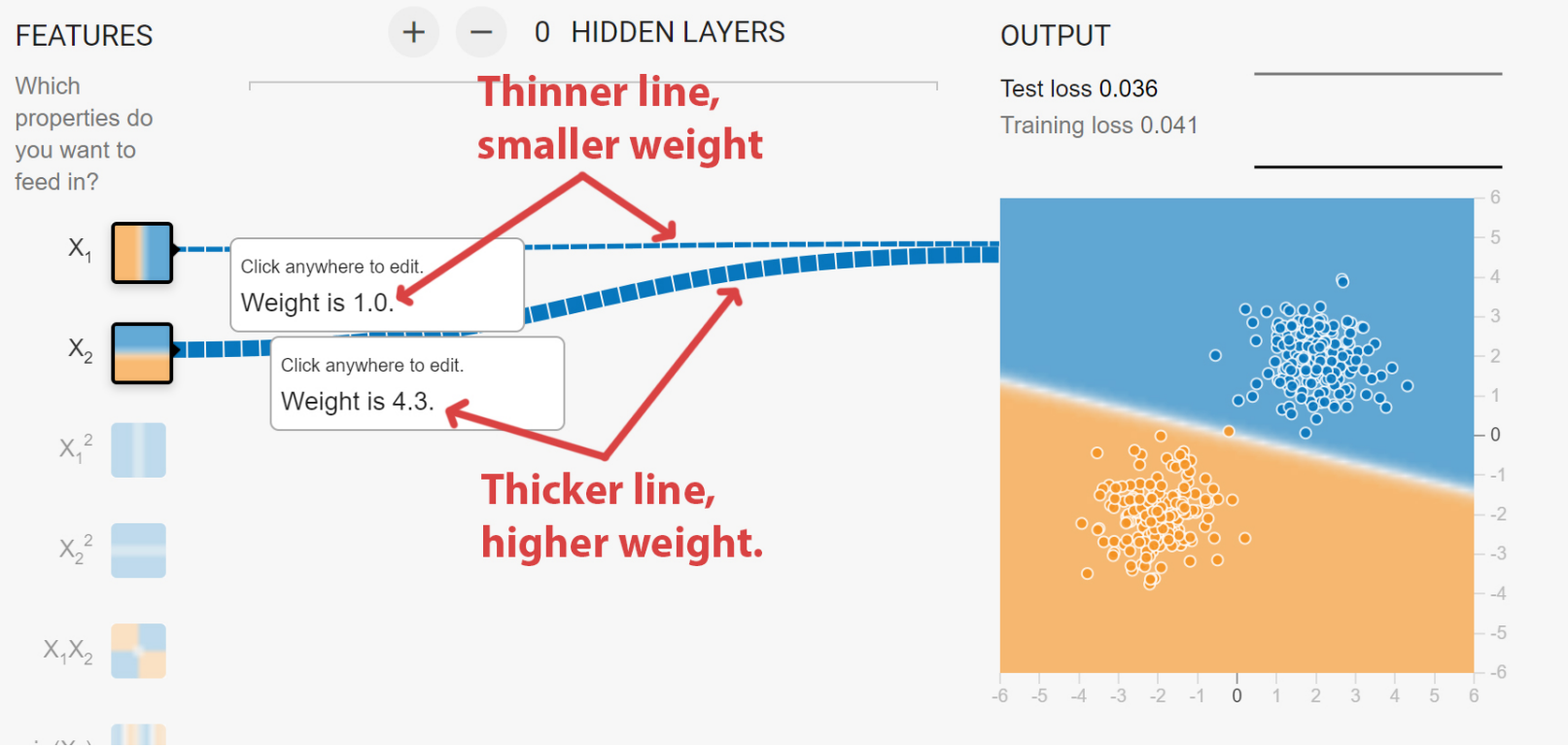
- Logistic regression
- Train model to map features to classes.
- Learn weights of the features.

Model

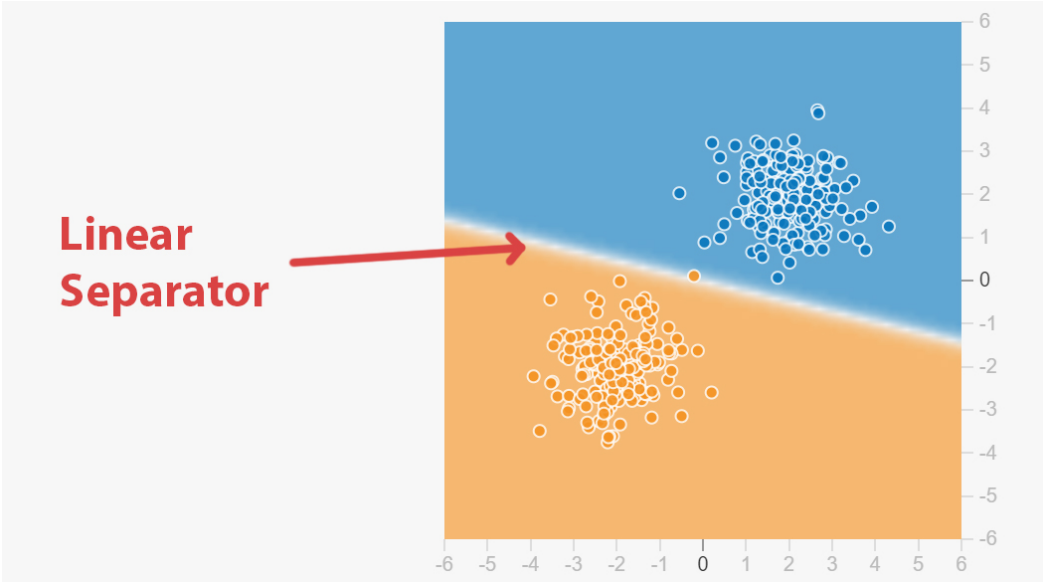
- Logistic regression
- Train model to map features to classes.
- Learn weights of the features.

Model

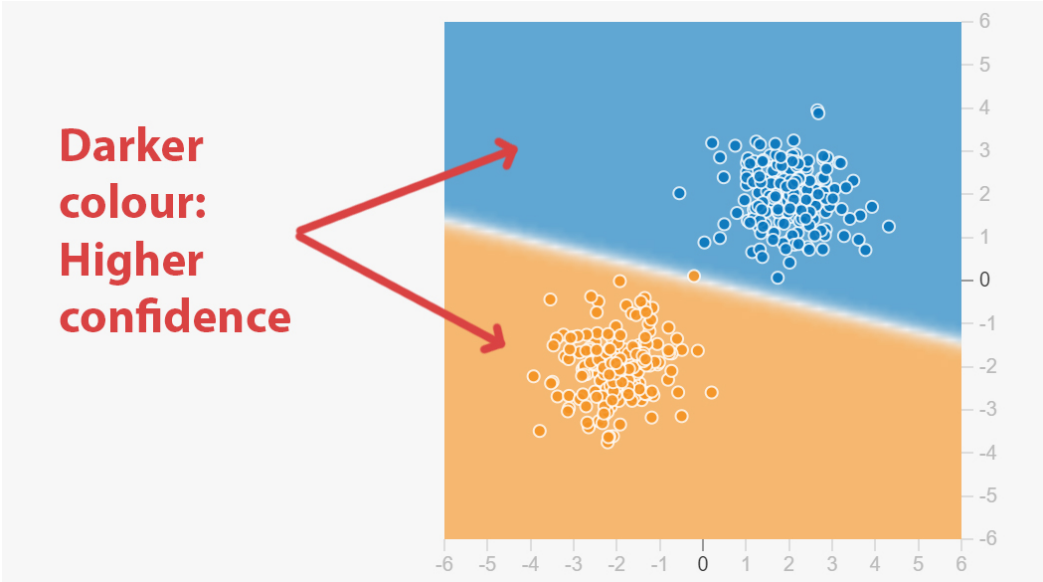
Hover over the line to see weights.



Model



Model


Darker
colour:
Higher
confidence
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6

Loss function

- Can't explicitly specify loss function in Tensorflow Playground.
- It changes with the type of problem.
- For classification 'binary cross entropy' loss function is used.
- Regularization: its type and rate can be adjusted.

Loss function

- Can't explicitly specify loss function in Tensorflow Playground.
- It changes with the type of problem.
- For classification 'binary cross entropy' loss function is used.
- Regularization: its type and rate can be adjusted.

Loss function

- Can't explicitly specify loss function in Tensorflow Playground.
- It changes with the type of problem.
- For classification 'binary cross entropy' loss function is used.
- Regularization: its type and rate can be adjusted.

Loss function

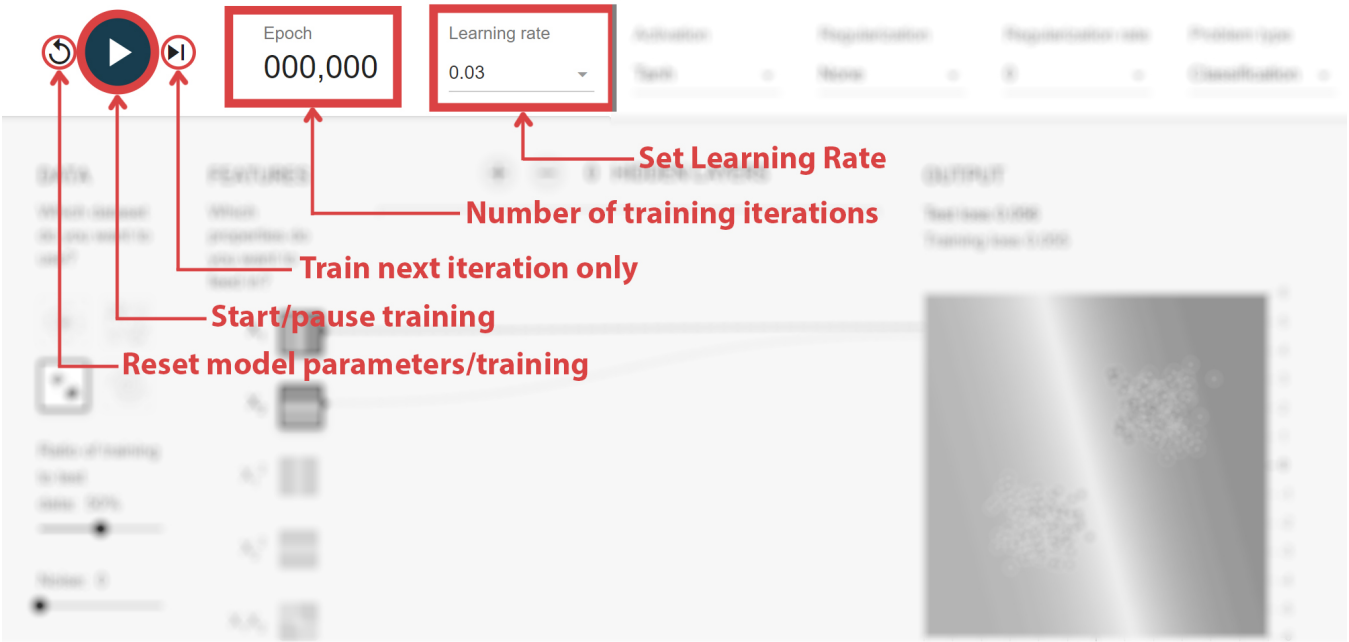
- Can't explicitly specify loss function in Tensorflow Playground.
- It changes with the type of problem.
- For classification 'binary cross entropy' loss function is used.
- Regularization: its type and rate can be adjusted.

Loss function

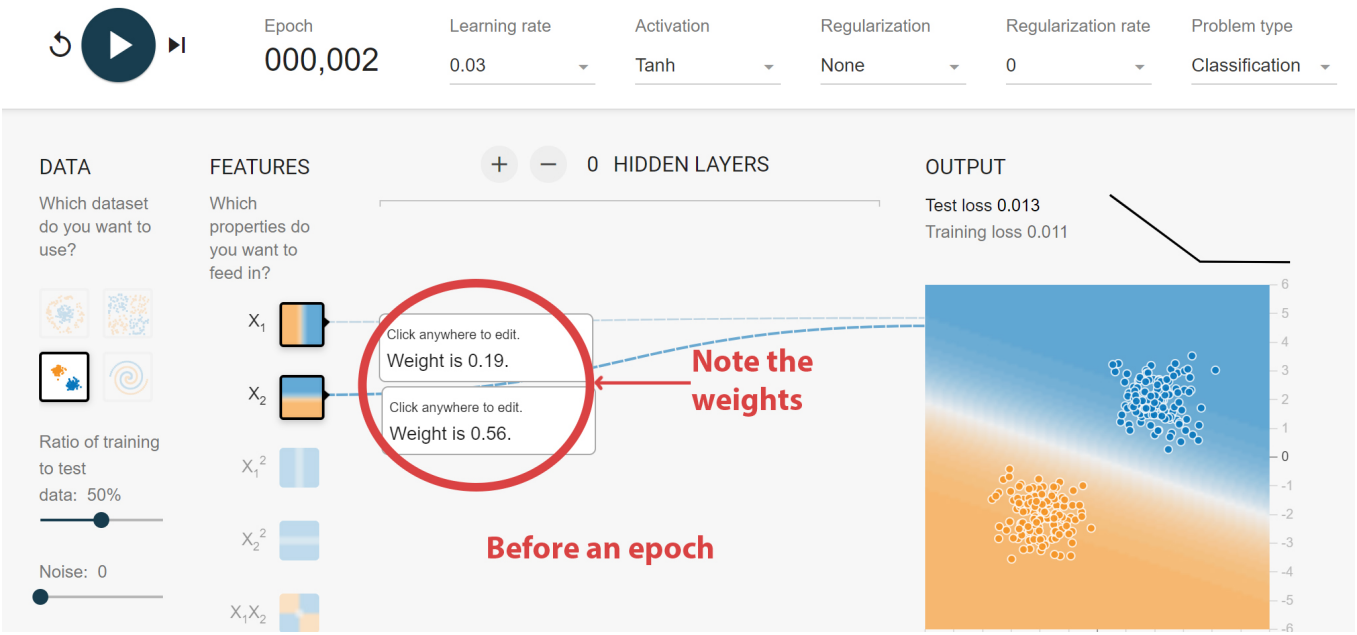
- Can't explicitly specify loss function in Tensorflow Playground.
- It changes with the type of problem.
- For classification 'binary cross entropy' loss function is used.
- Regularization: its type and rate can be adjusted.

Optimization Procedure

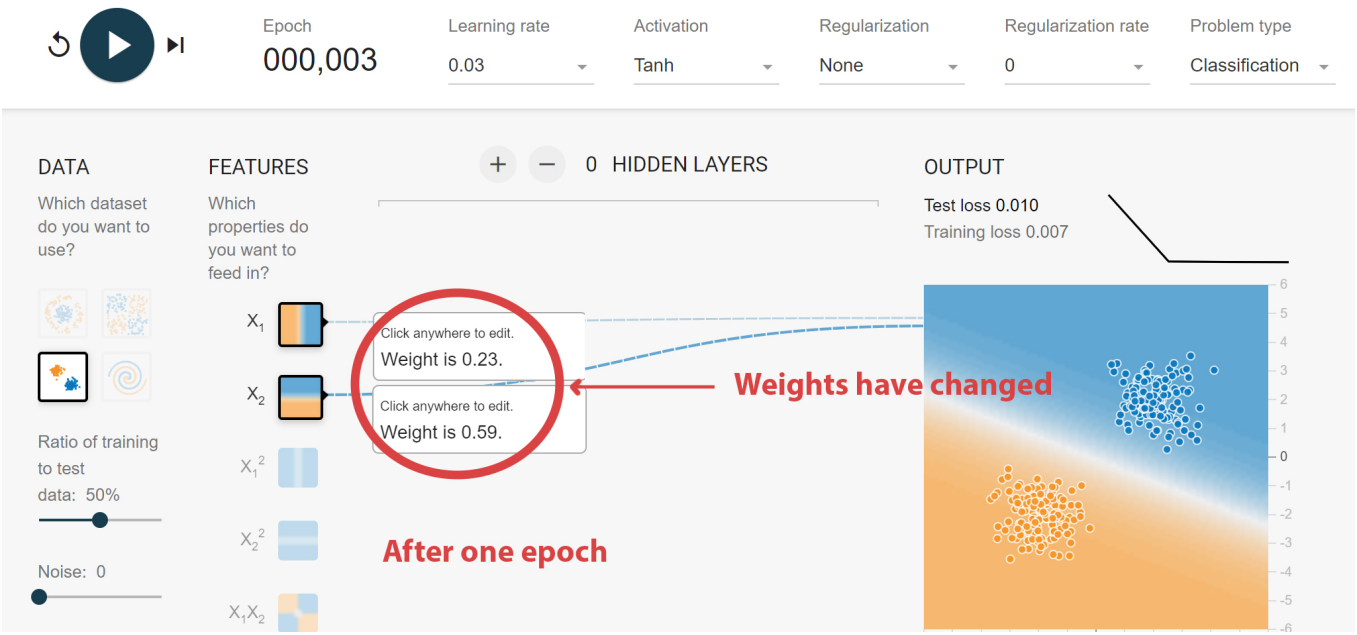
- Batch-gradient descent for optimization.



New Model Instance per epoch



New Model Instance per epoch



Evaluation Procedure

- Evaluate the goodness of the model with an evaluation protocol.
- The playground uses only training and test loss for evaluation. There is no scope for any customization or selection.
- As the training proceeds, we get to see changes in the test and training loss.

Evaluation Procedure

- Evaluate the goodness of the model with an evaluation protocol.
- The playground uses only training and test loss for evaluation. There is no scope for any customization or selection.
- As the training proceeds, we get to see changes in the test and training loss.

Evaluation Procedure

- Evaluate the goodness of the model with an evaluation protocol.
- The playground uses only training and test loss for evaluation. There is no scope for any customization or selection.
- As the training proceeds, we get to see changes in the test and training loss.

Evaluation Procedure

- Evaluate the goodness of the model with an evaluation protocol.
- The playground uses only training and test loss for evaluation. There is no scope for any customization or selection.
- As the training proceeds, we get to see changes in the test and training loss.

Evaluation Procedure

When the model is getting trained:

- Both the losses should gradually decrease.
- The rate of decrease is proportional to the learning rate.

Evaluation Procedure

When the model is getting trained:

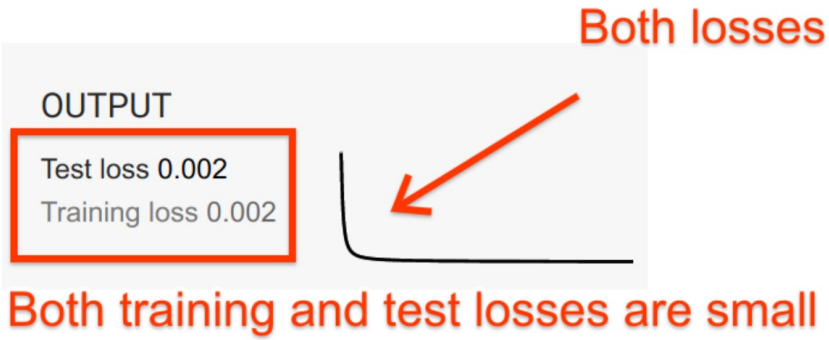
- Both the losses should gradually decrease.

- The rate of decrease is proportional to the learning rate.

Evaluation Procedure

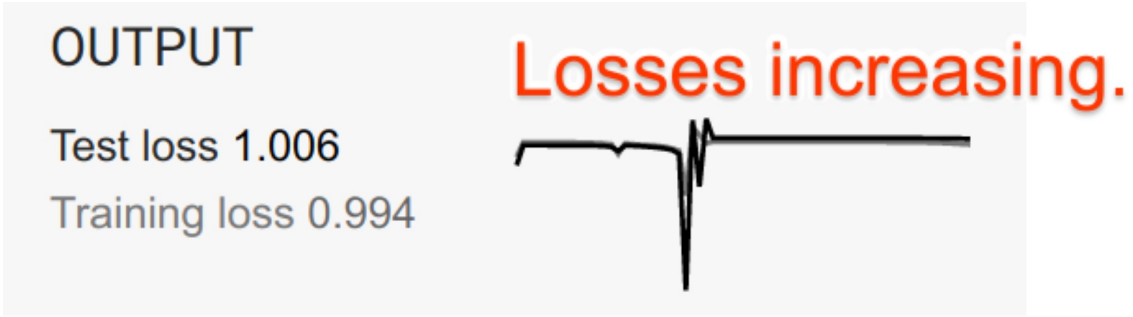
When the model is getting trained:

- Both the losses should gradually decrease.



- The rate of decrease is proportional to the learning rate.

Evaluation Procedure



If the losses are not going down, that means the model is not training, possibly because:

- The learning rate is too high, or
- There are no informative features describing the training examples.

Evaluation Procedure

If the losses are not going down, that means the model is not training, possibly because:

- The learning rate is too high, or
- There are no informative features describing the training examples.

Evaluation Procedure

If the model is getting trained properly:

- Both the training and test losses go down.
- Both of them should be close to 0.

Evaluation Procedure

If the model is getting trained properly:

- Both the training and test losses go down.
- Both of them should be close to 0.

Evaluation Procedure

If the model is getting trained properly:

- Both the training and test losses go down.
- Both of them should be close to 0.

A tour

Let's take a tour of TensorFlow Playground now.