Deep Learning for Business

Deep Learning Project with TensorFlow Playground

Prof. Jong-Moon Chung

Deep Learning for Business

Deep Learning Project with TensorFlow Playground

Introduction to TensorFlow Playground

Introduction to TensorFlow Playground

TensorFlow Playground

- Web application written in d3.js
- Best web application to learn about

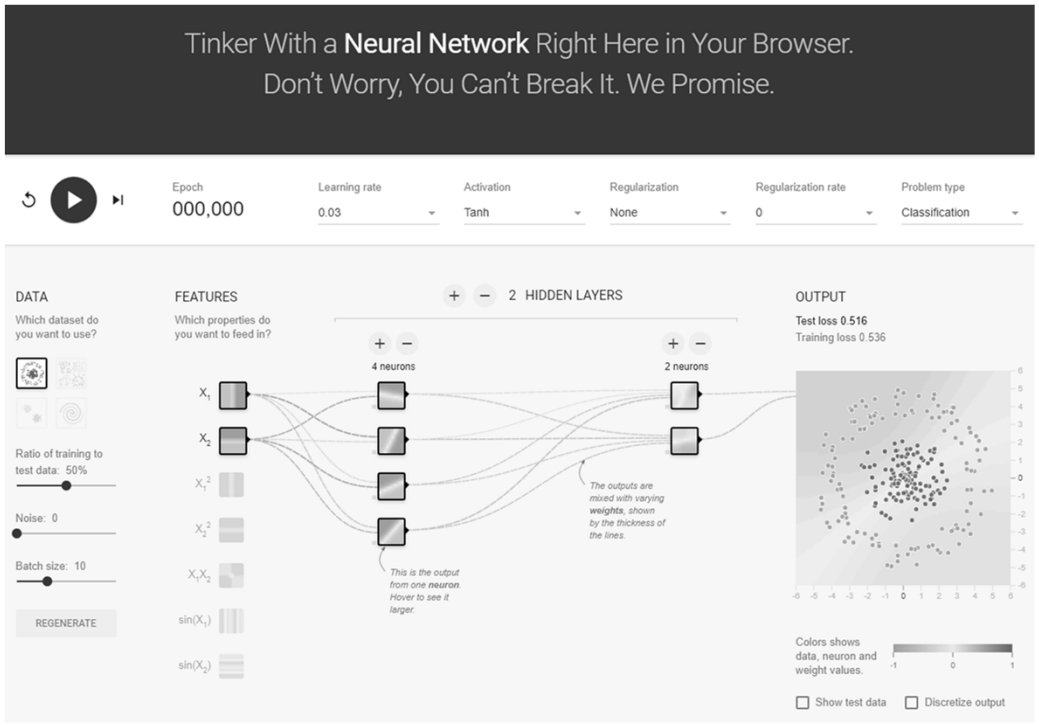
  NNs (Neural Networks) without math
- In your web browser, you can create a NN and immediately see your results
- Licensed under the Apache License 2.0, January 2004 (http://www.apache.org/licenses/)

Introduction to TensorFlow Playground

- Creators of TensorFlow Playground
  - Daniel Smilkov and Shan Carter
- Based on a continuation of many previous works
  - Most notably Andrej Karpathy's convnet.js demo and Chris Olah's articles on NNs (Neural Networks)
- Contributing members
  - D. Sculley for help with the original idea
  - Fernanda Viégas and Martin Wattenberg and the **Big Picture** and **Google Brain** teams for feedback and guidance
- Thank you for making this great web application!

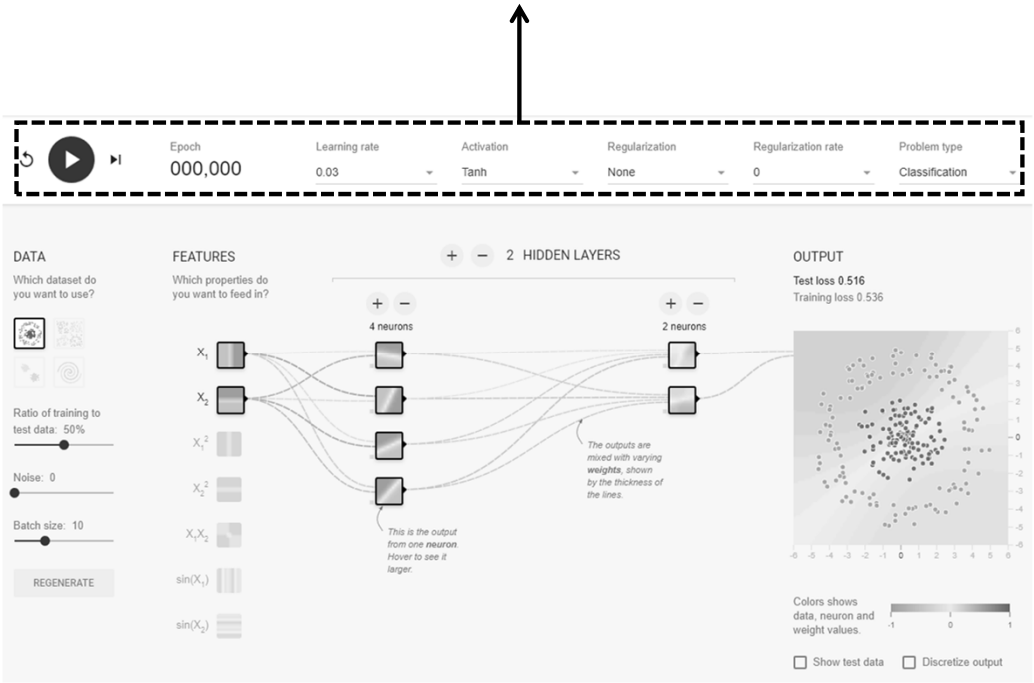
Introduction to TensorFlow Playground

– Go to http://playground.tensorflow.org



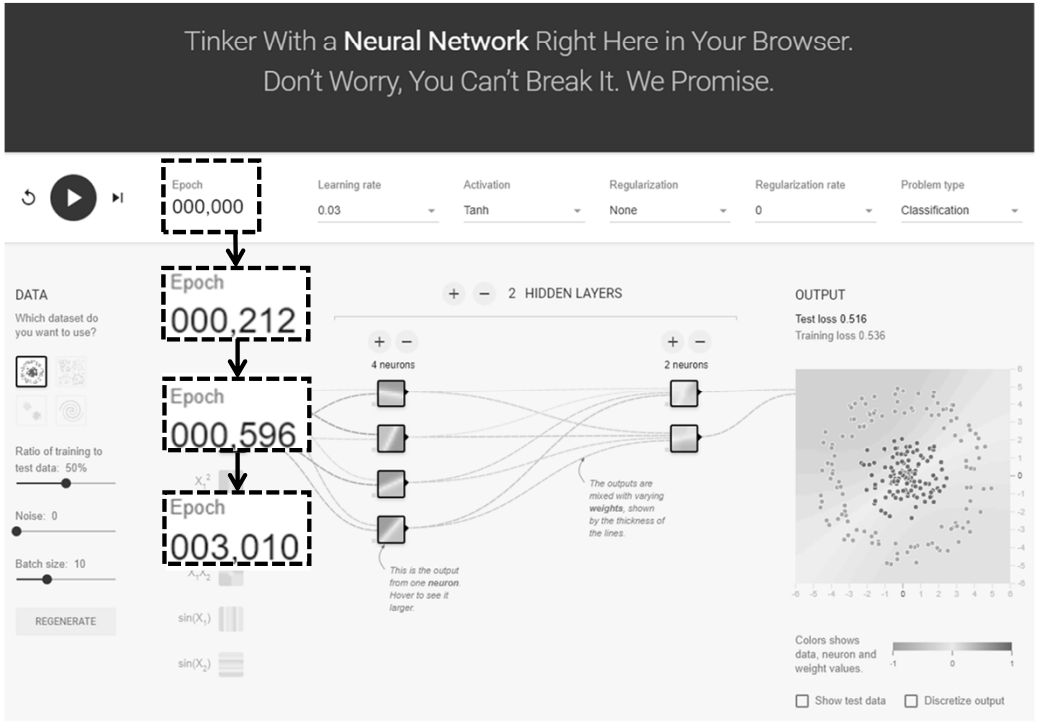
Introduction to TensorFlow Playground

- Epoch, Learning rate, Activation,
  Regularization rate, Problem type



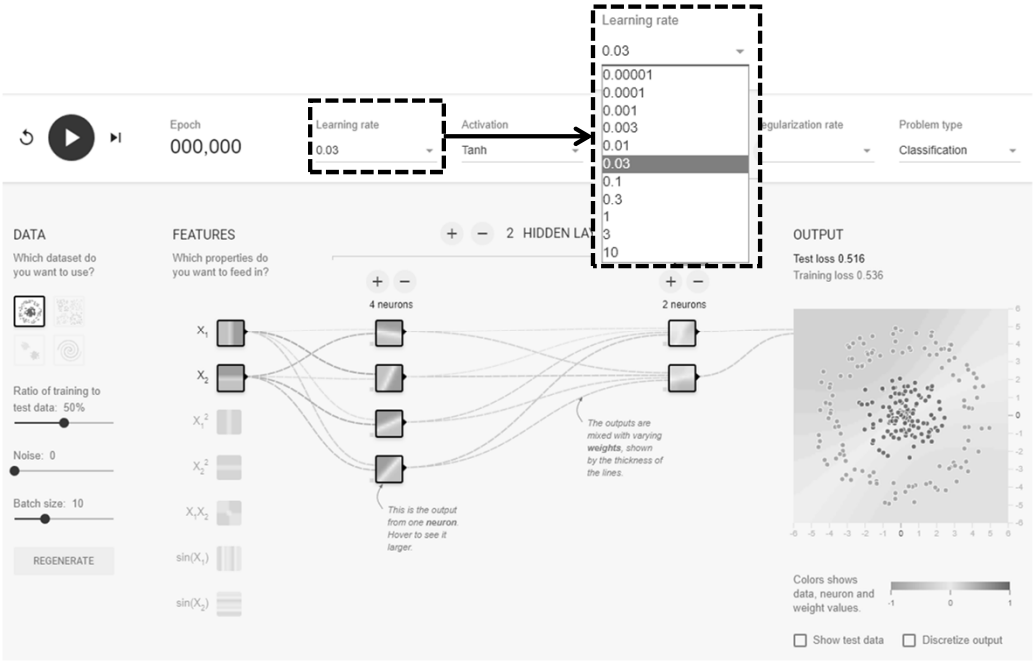
Introduction to TensorFlow Playground

- Every time training is conducted for a whole training set, the Epoch number increases



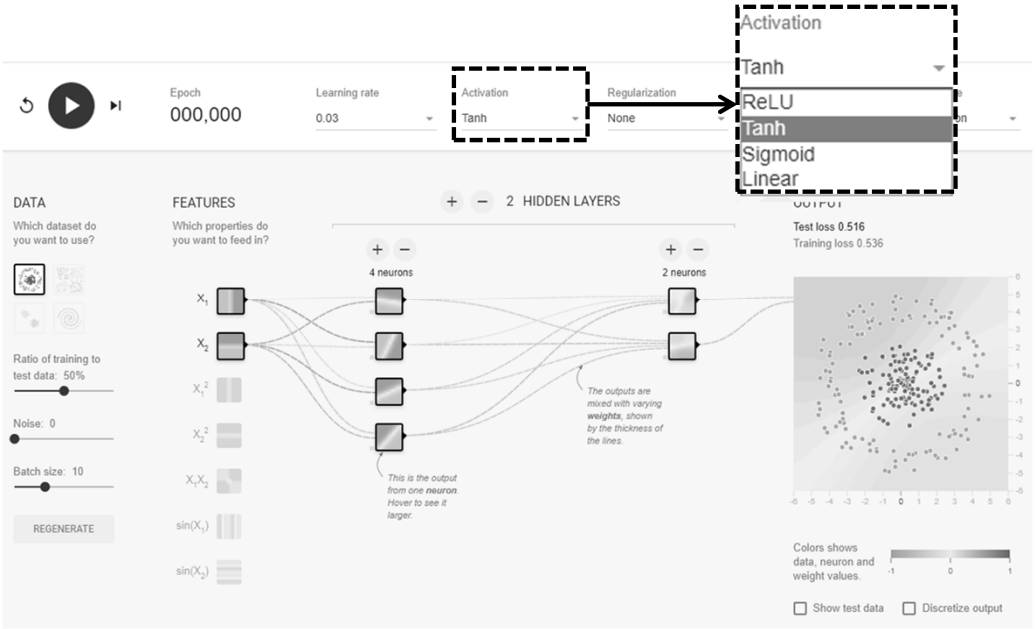
Introduction to TensorFlow Playground

- Learning rate determines the learning speed
- Select the appropriate learning rate



Introduction to TensorFlow Playground

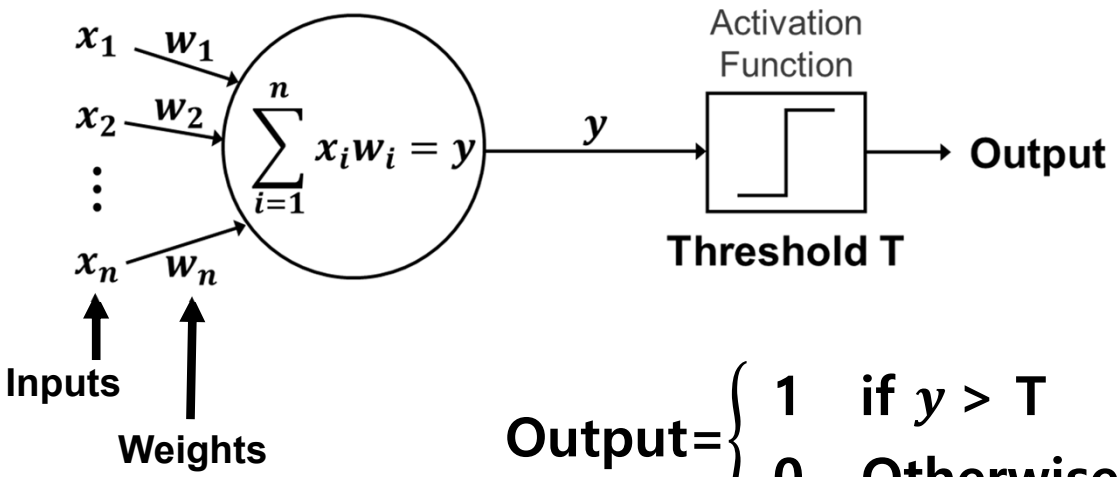
- Activation function type needs to be selected
- More information on activation functions are in the lectures of Module 4



Review of Module 4

ANN (Artificial Neural Network) Neuron

- **Activation Function with Hard Output**



$$\text{Output} = \begin{cases} 1 & \text{if } y > \text{T} \\ 0 & \text{Otherwise} \end{cases}$$

Review of Module 4

Soft Output Activation Functions

- ReLU (Rectified Linear Unit) $g(y) = \max(0, y)$

- Tanh (Hyperbolic Tangent) $t(y) = \frac{1 - \exp(-2y)}{1 + \exp(-2y)}$

- Sigmoid (Logistic Sigmoid) $\sigma(y) = \frac{1}{1 + \exp(-y)}$

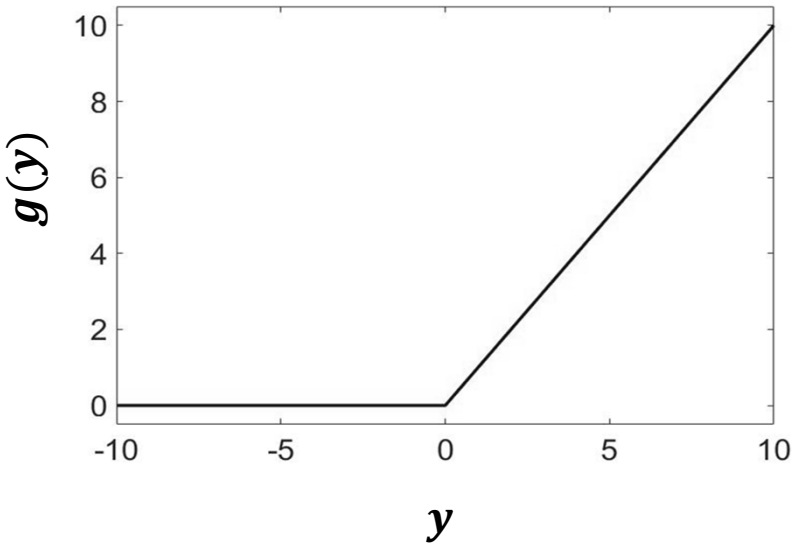
- Linear (e.g., α=1) $l(y) = \alpha y$

Review of Module 4

Soft Output Activation Functions

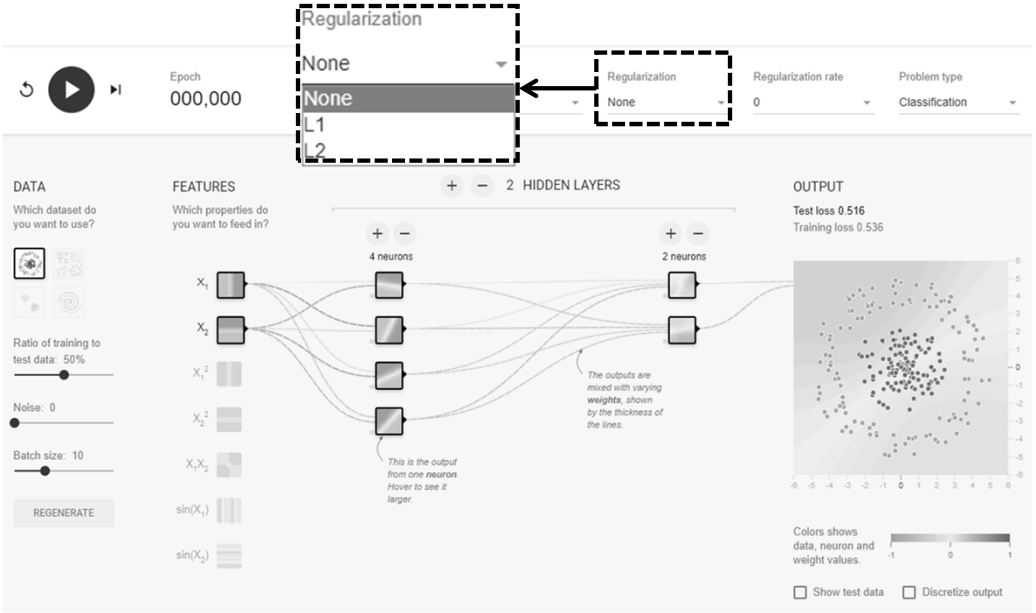
– ReLU (Rectified Linear Unit)

$$g(y) = \max(0, y)$$



Introduction to TensorFlow Playground

- Regularization is used to prevent overfitting
- TensorFlow Playground provides two types of regularization: L1, L2



Introduction to TensorFlow Playground

Regularization L1 and L2

- Regularization slowly increases/reduces the weights of strong/weak connections to make the pattern classification sharper

- L1 and L2 are popular regularization methods

- Dropout is also a regularization method
  - Dropout is explained in the lectures of Module 5
    "Deep Learning with CNN (Convolutional Neural Network)"

Introduction to TensorFlow Playground

L1 Regularization

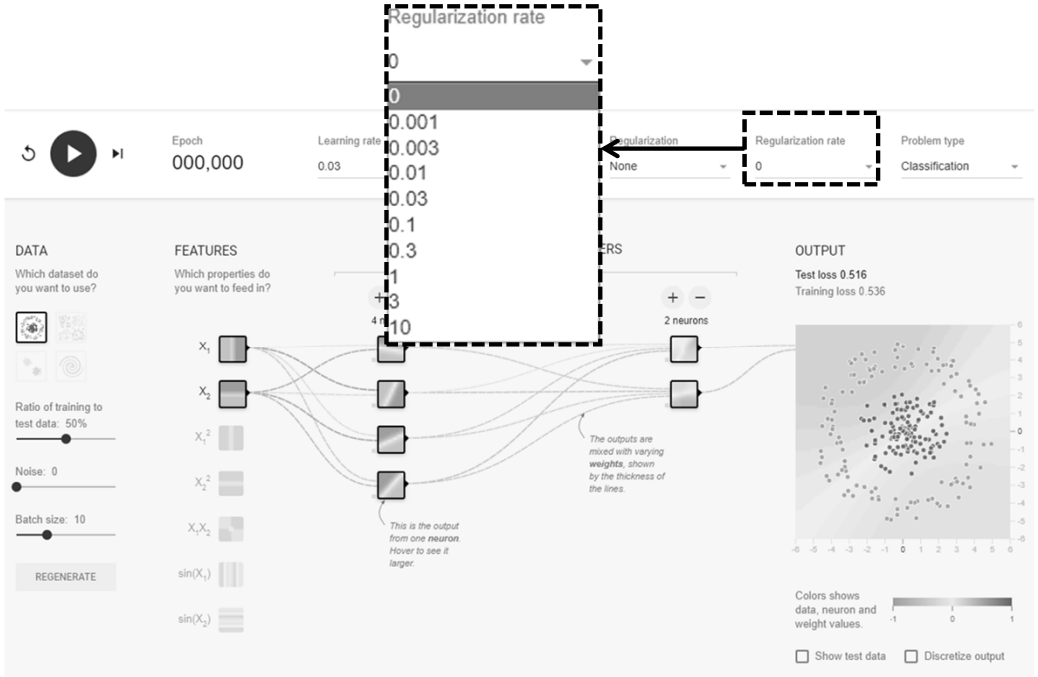
- L1 is effective in sparse feature spaces, where there is a need to select a few among many
- L1 will make selections and assign big weight values, and will make the weights of the non-selected ones very small (or zero)

L2 Regularization

- L2 is effective with inputs that are correlated
- L2 will control the weight values corresponding to the level of correlation

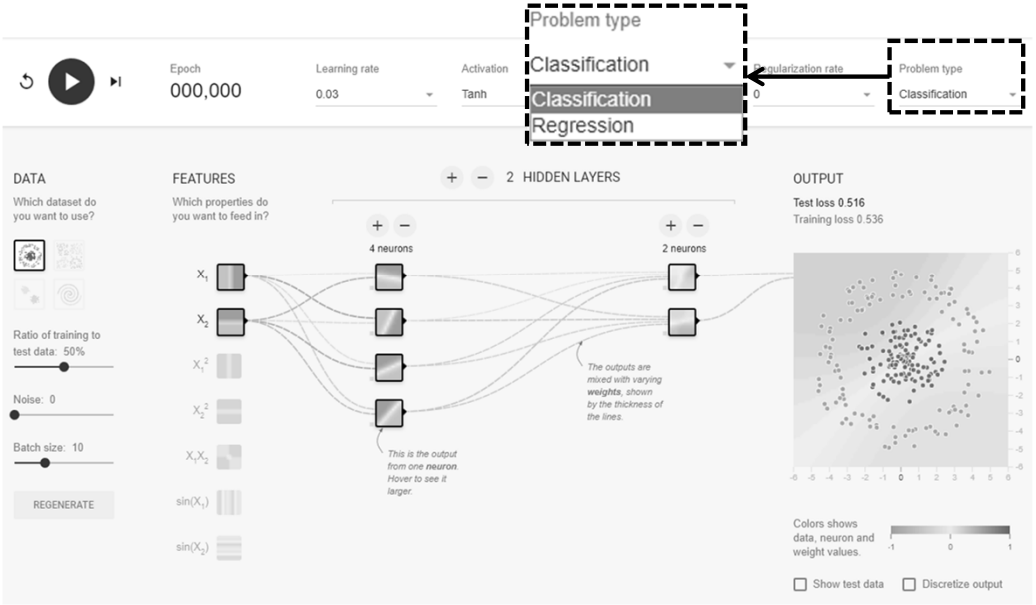
Introduction to TensorFlow Playground

- A higher Regularization Rate will make the weights more limited in range



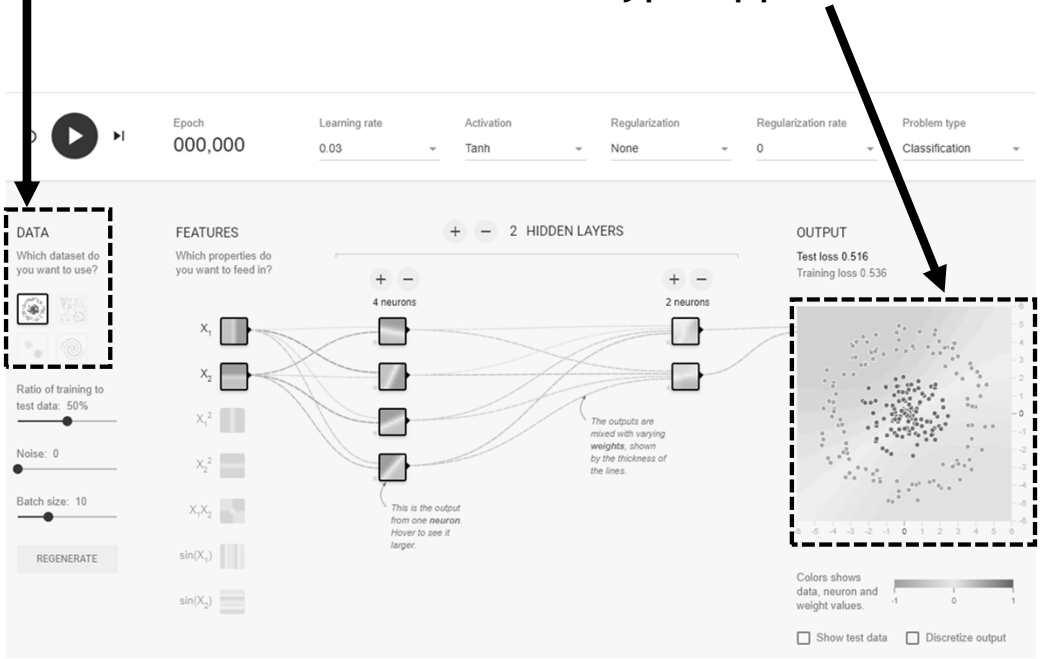
Introduction to TensorFlow Playground

- Select among the two types of problems
  - Classification
  - Regression



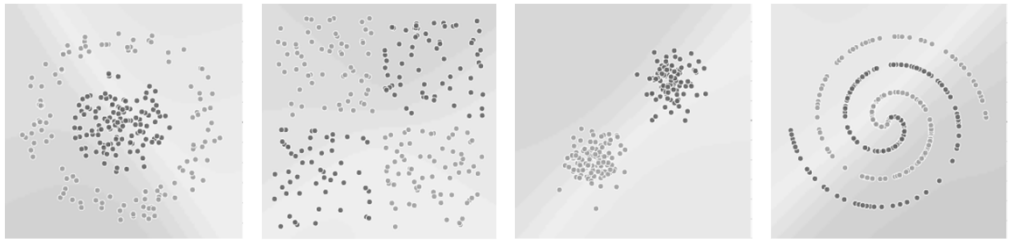
Introduction to TensorFlow Playground

- Select the type of Data set
  - Picture of the Data set type appears

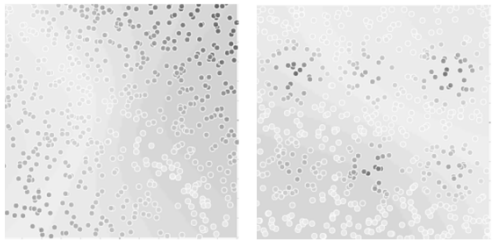


Introduction to TensorFlow Playground

- Four types of Classification

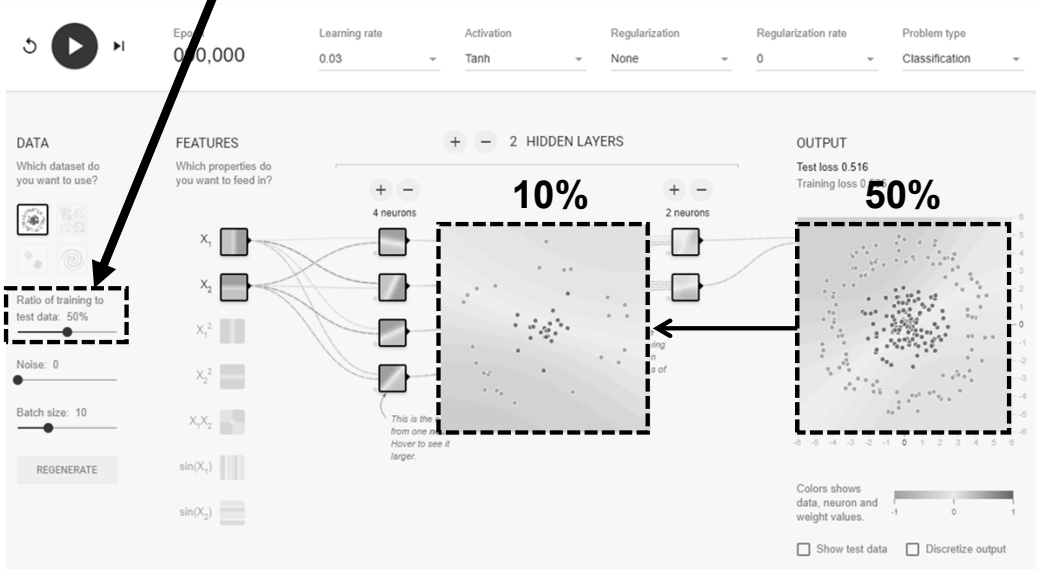


- Two types of Regression



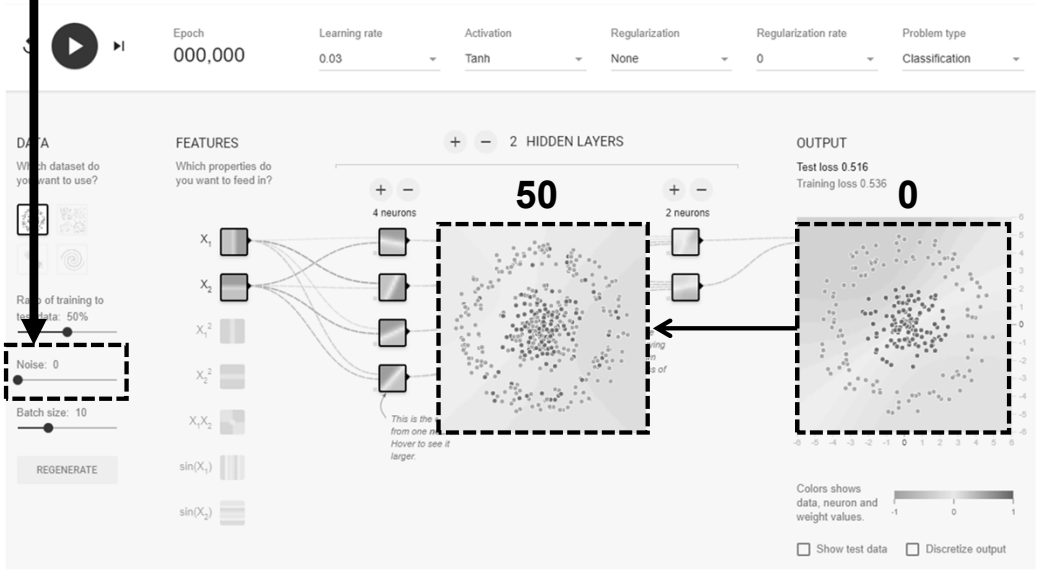
Introduction to TensorFlow Playground

- Blue and orange dots form data sets
  - Orange dot = -1, Blue dot = +1
- Using the **Ratio of training to test data**, the percentage of the training set can be controlled



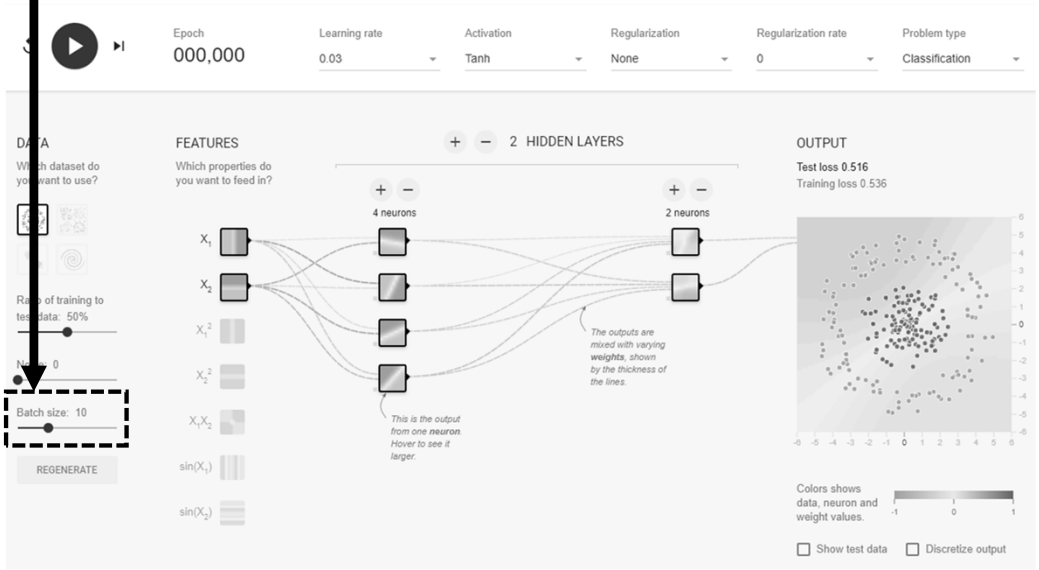
Introduction to TensorFlow Playground

- Noise level of the data set can be controlled
  - Data pattern becomes more irregular as the noise increases



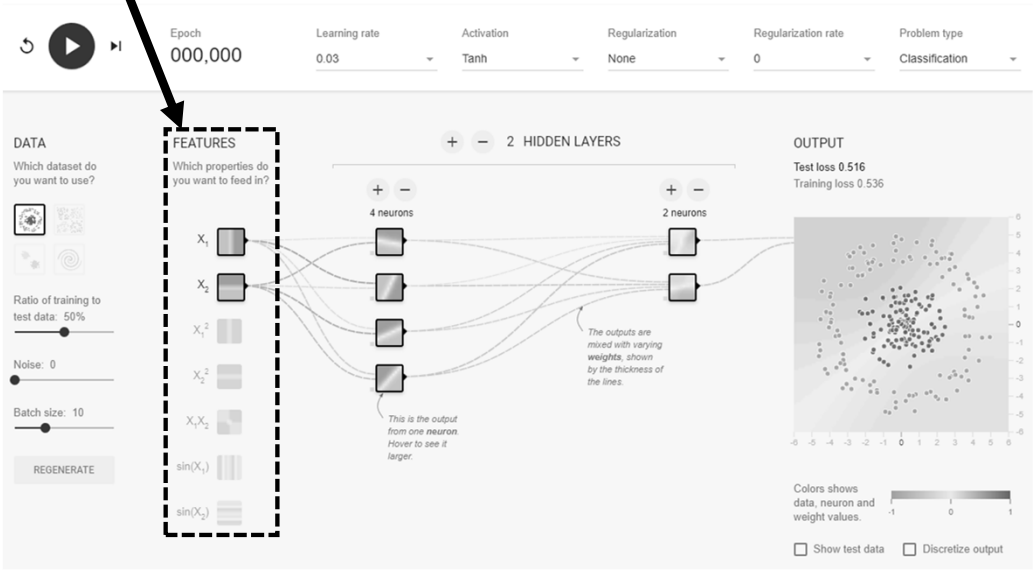
Introduction to TensorFlow Playground

- **Batch** size determines the data amount to use for each training iteration



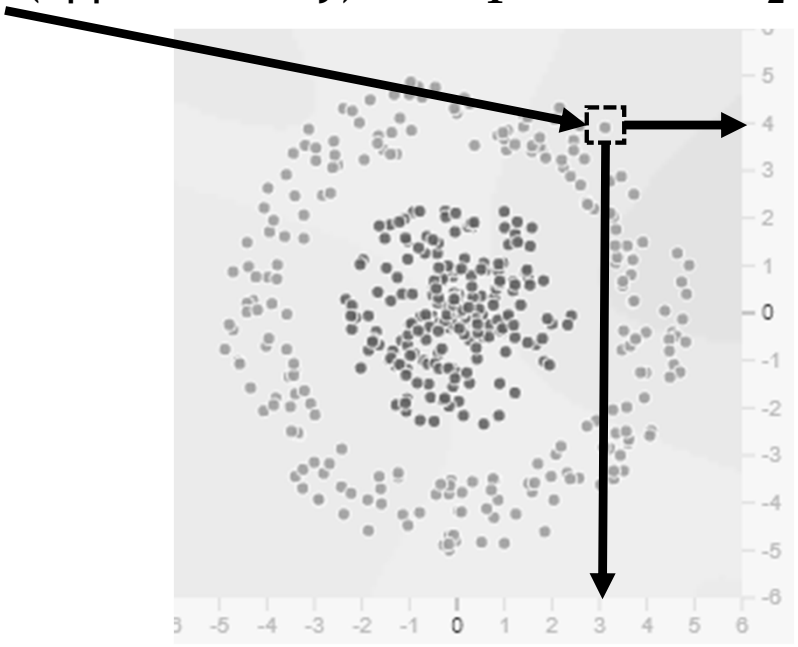
Introduction to TensorFlow Playground

- **Feature** selection
  - X_1 is a value on the horizontal axis
  - X_2 is a value on the vertical axis



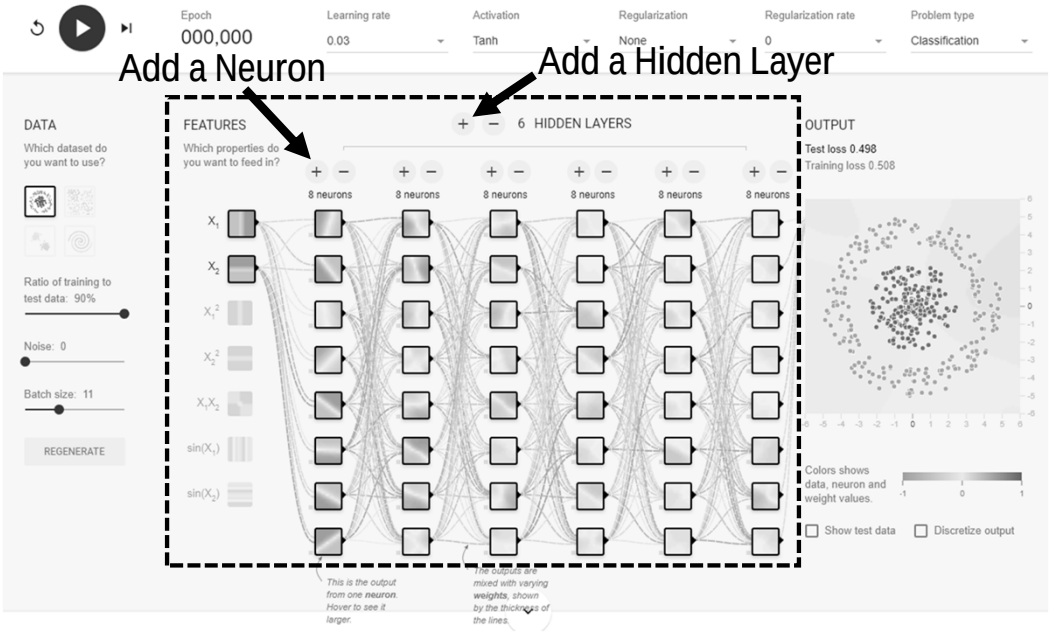
Introduction to TensorFlow Playground

- Example of X_1 and X_2
  - This **dot** (approximately) has $\boldsymbol{X_1 \approx 3.1}$ and $\boldsymbol{X_2 \approx 4}$



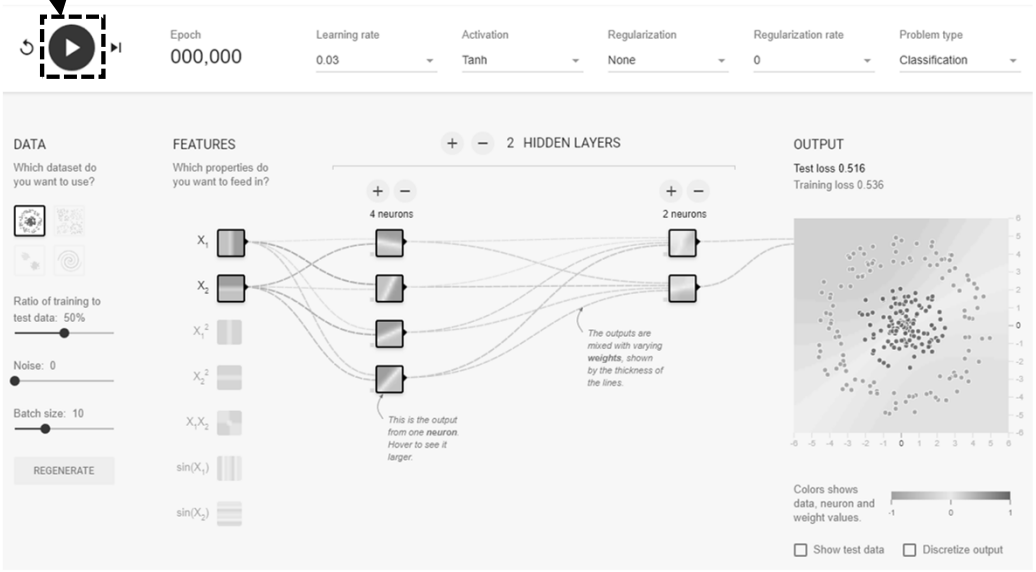
Introduction to TensorFlow Playground

- Hidden layer structure
  - Up to **6 hidden layers** can be set
  - Up to **8 neurons** per hidden layer can be set



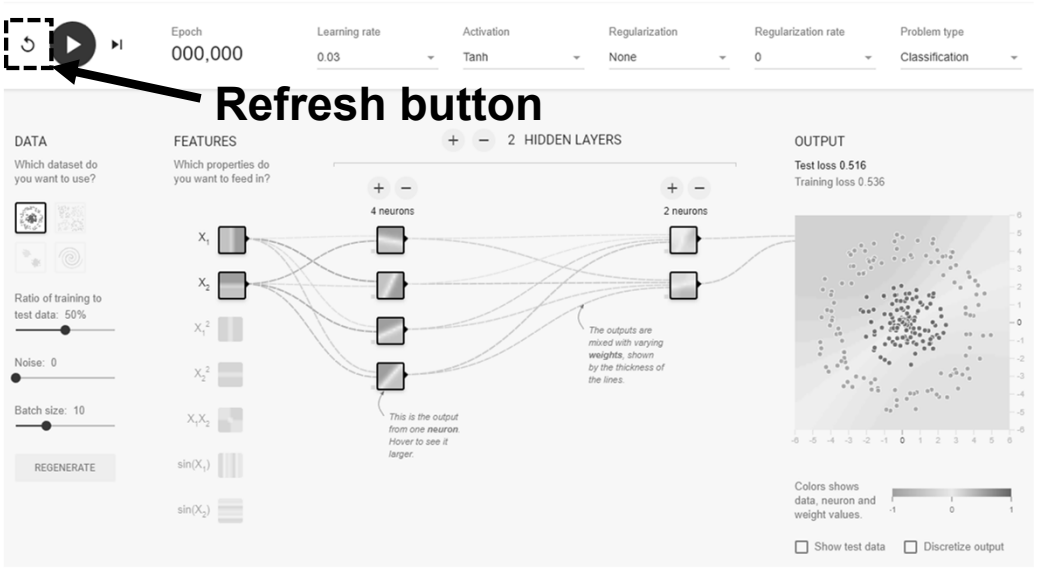
Introduction to TensorFlow Playground

- Pressing the arrow ▶ button starts the NN training
  - Epoch will be increased by one
  - Backpropagation will be used to train the NN



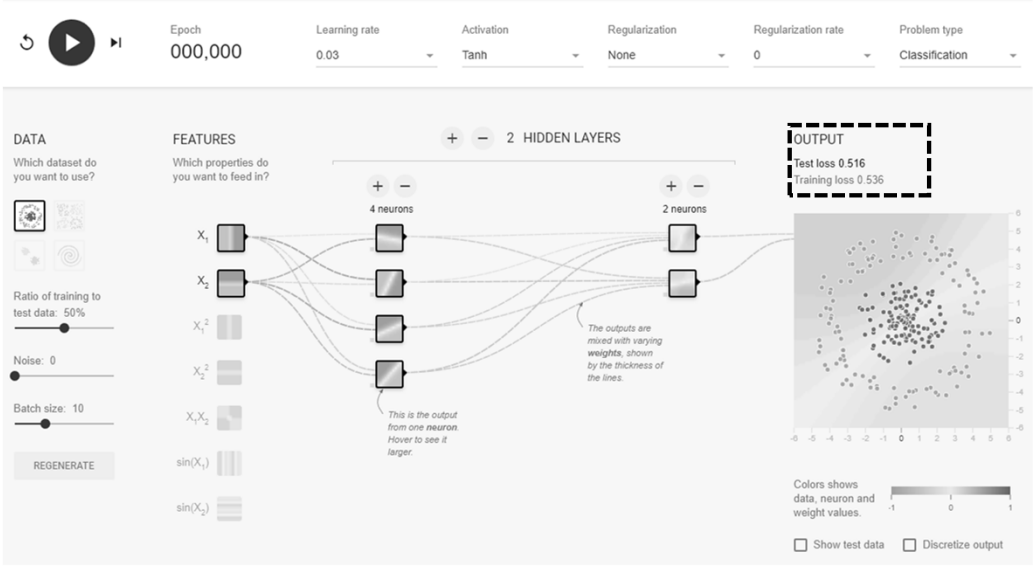
Introduction to TensorFlow Playground

- Pressing the arrow ▶ button starts the NN training
  - Epoch will be increased by one
  - Backpropagation will be used to train the NN



Introduction to TensorFlow Playground

- NN minimizes the Test Loss and Training Loss
- Test Loss (black) and Training Loss (grey) changes will be shown in small performance curves
- If the Loss is reduced, the curves will go down



Deep Learning for Business

Deep Learning Project with TensorFlow Playground

References

References

- A Neural Network Playground – TensorFlow [Online]. Available: http://playground.tensorflow.org
- Google Cloud Big Data and Machine Learning Blog [Online]. Available: https://cloud.google.com/blog/big-data/2016/07/understanding-neural-networks-with-tensorflow-playground
- Google Open Source [Online]. Available: https://opensource.google.com/projects/tensorflow-playground