Intro to Deep Learning

1) The neuron model
2) Gradient descent
3) Architectures (mostly for vision)
4) **Model evaluation**

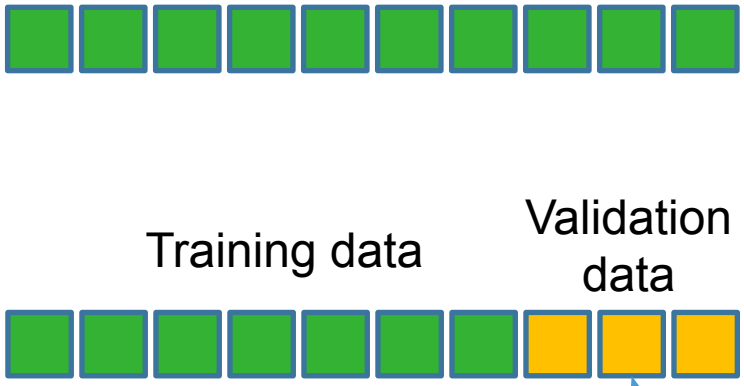
Training Error and Generalization Error

- Training error: model error on the training data
- Generalization error: model error on new data
- Example: practice a future exam with past exams
  - Doing well on past exams (training error) doesn't guarantee a good score on the future exam (generalization error)
  - Student A learns 100% of the answers by heart
  - Student B understands the reasons for 90% of answers

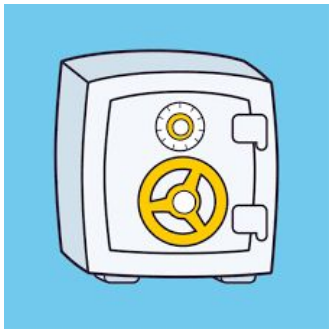
Validation Dataset and Test Dataset

- Validation dataset: a dataset used to select the model
  - E.g. Take out 50% of the training data
  - Should not be mixed with the training data (#1 mistake)
- Test dataset: a dataset that can be used once, e.g.
  - A future exam
  - Dataset used in private leaderboard in Kaggle

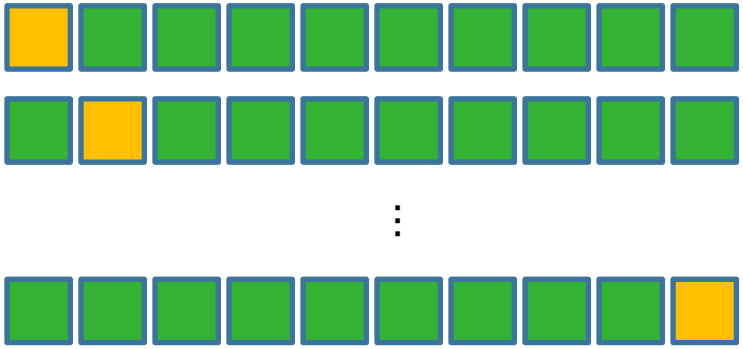
Validation



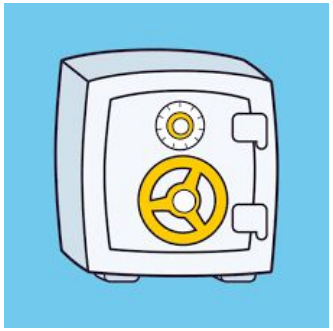
Test data



K-fold Cross Validation



Test data



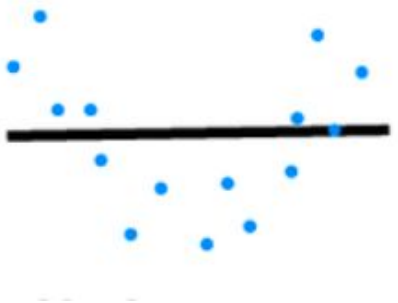
Model chosen based on its average performance across all folds

Underfitting
Overfitting

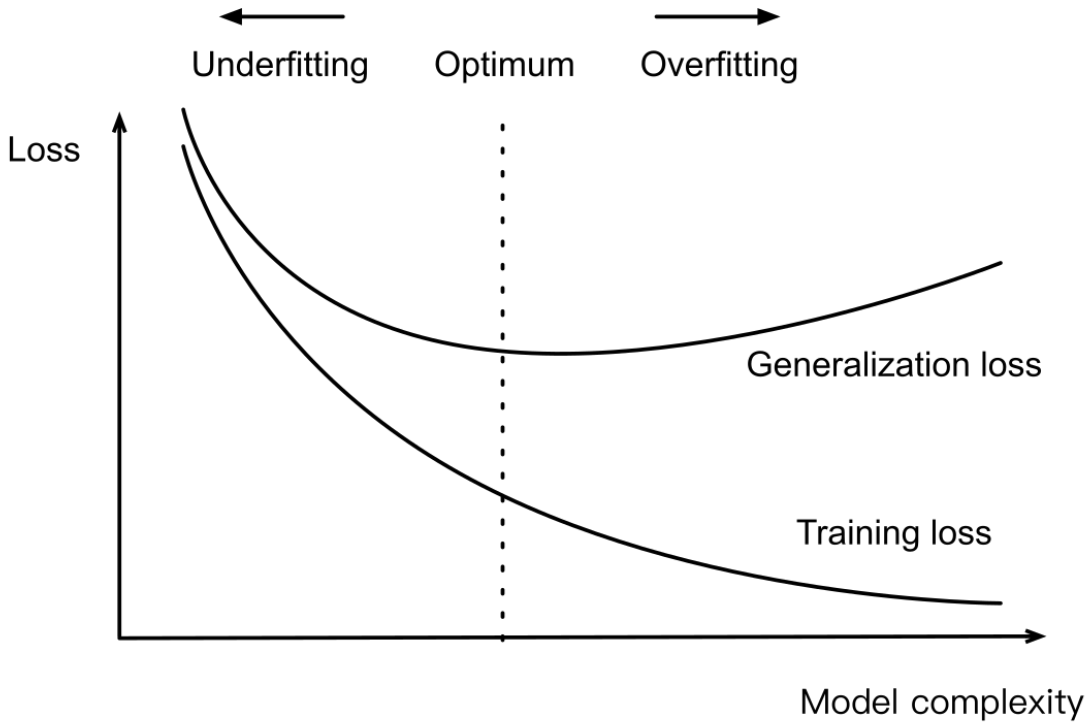
Image credit: hackernoon.com

Model Capacity

- The ability to fit variety of functions
- Low capacity models struggles to fit training set
  - Underfitting
- High capacity models can memorize the training set
  - Overfitting



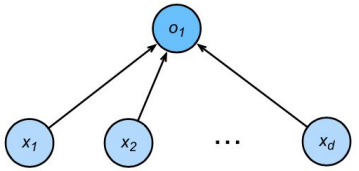
Influence of Model Complexity



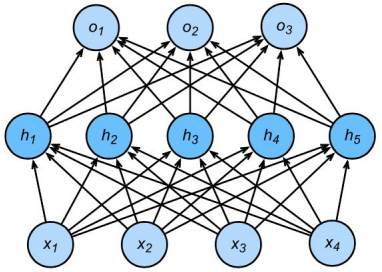
Estimate Model Capacity

- It's hard to compare complexity between different algorithms
  - e.g. tree vs neural network
- Given an algorithm family, two main factors matter:
  - The number of parameters
  - The values taken by each parameter

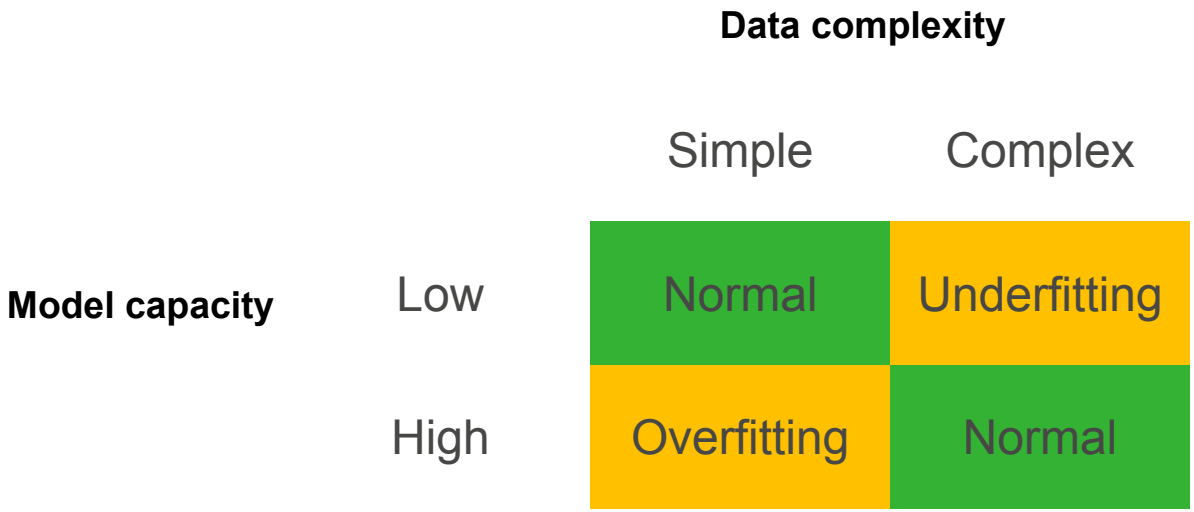
$d + 1$



$(d + 1)m + (m + 1)k$

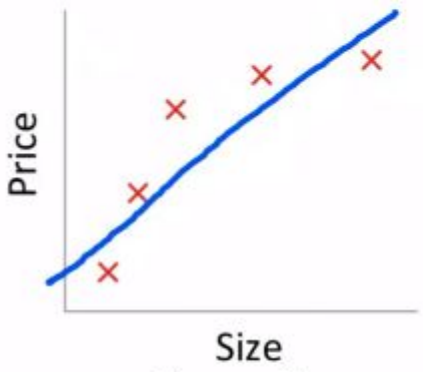


Underfitting and Overfitting



Regularization with weight decay

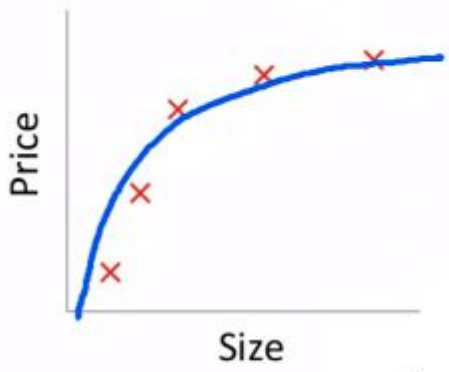
Bias/variance



$f(x) = b + w_1 x$

High bias
(underfit)

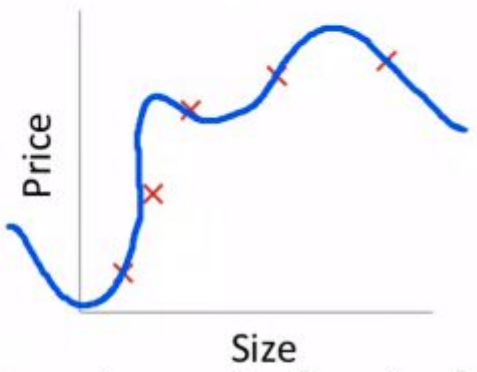
$d = 1$



$f(x) = b + w_1 x + w_2 x^2$

"Just right"

$d = 2$



$f(x) = b + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

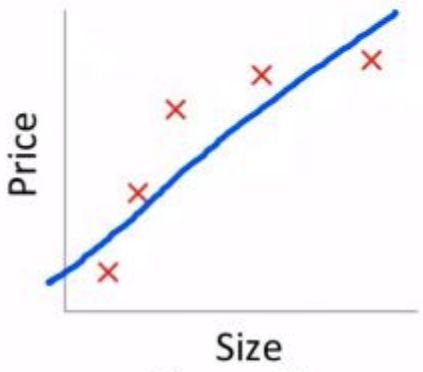
High variance
(overfit)

$d = 4$

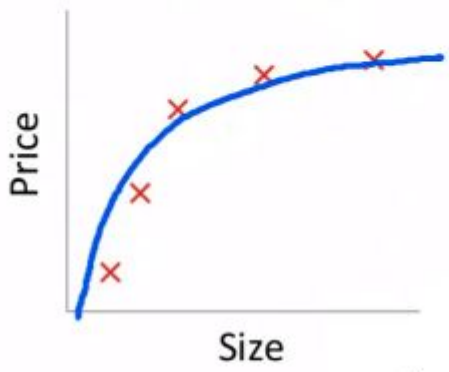
Andrew Ng

Regularization with weight decay

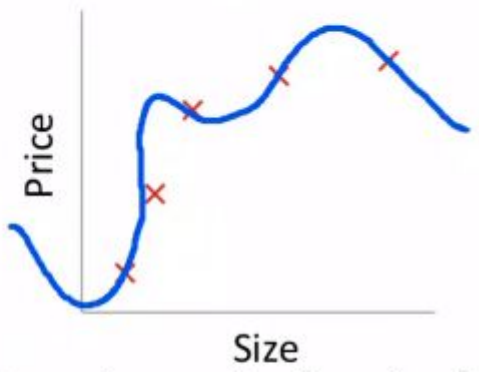
Bias/variance



$$f(x) = b + w_1 x$$



$$f(x) = b + w_1 x + w_2 x^2$$

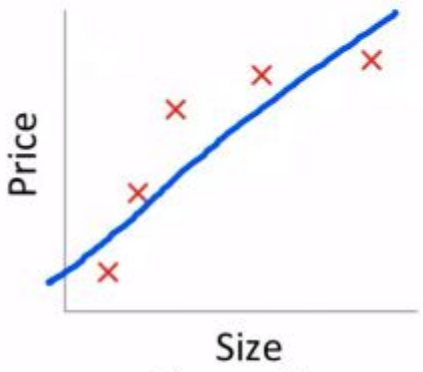


$$f(x) = b + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

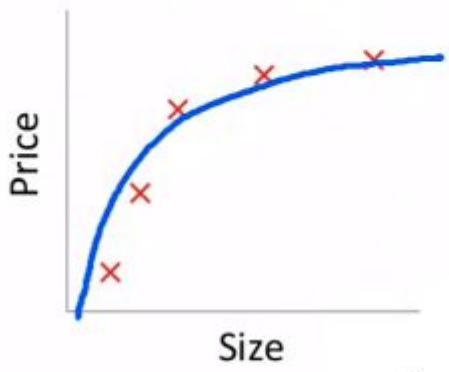
$$\boldsymbol{L}(\boldsymbol{w}, b) = \sum_{i=1}^{d} (f(x) - y)^2$$

Regularization with weight decay

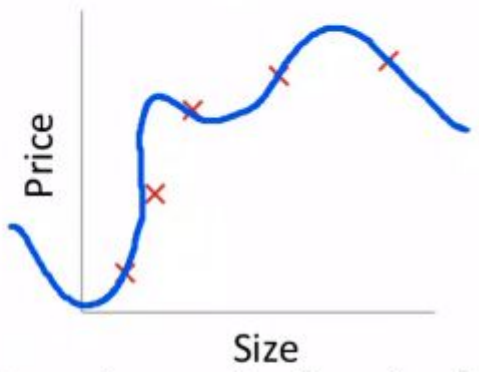
Bias/variance



$$f(x) = b + w_1 x$$



$$f(x) = b + w_1 x + w_2 x^2$$



$$f(x) = b + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

$$\boldsymbol{L}(\boldsymbol{w}, b) = \sum_{i=1}^{d} (f(x) - y)^2 + \lambda \|\boldsymbol{w}\|^2$$

Regularization with weight decay

- L2-norm regularization:

$$\lambda\|\boldsymbol{w}\|^2 = \lambda(w_1^2 + w_2^2 + w_3^2 + w_4^2)$$

- L1-norm regularization:

$$\lambda\|\boldsymbol{w}\| = \lambda(|w_1| + |w_2| + |w_3| + |w_4|)$$

Regularization with weight decay

- L2-norm regularization:

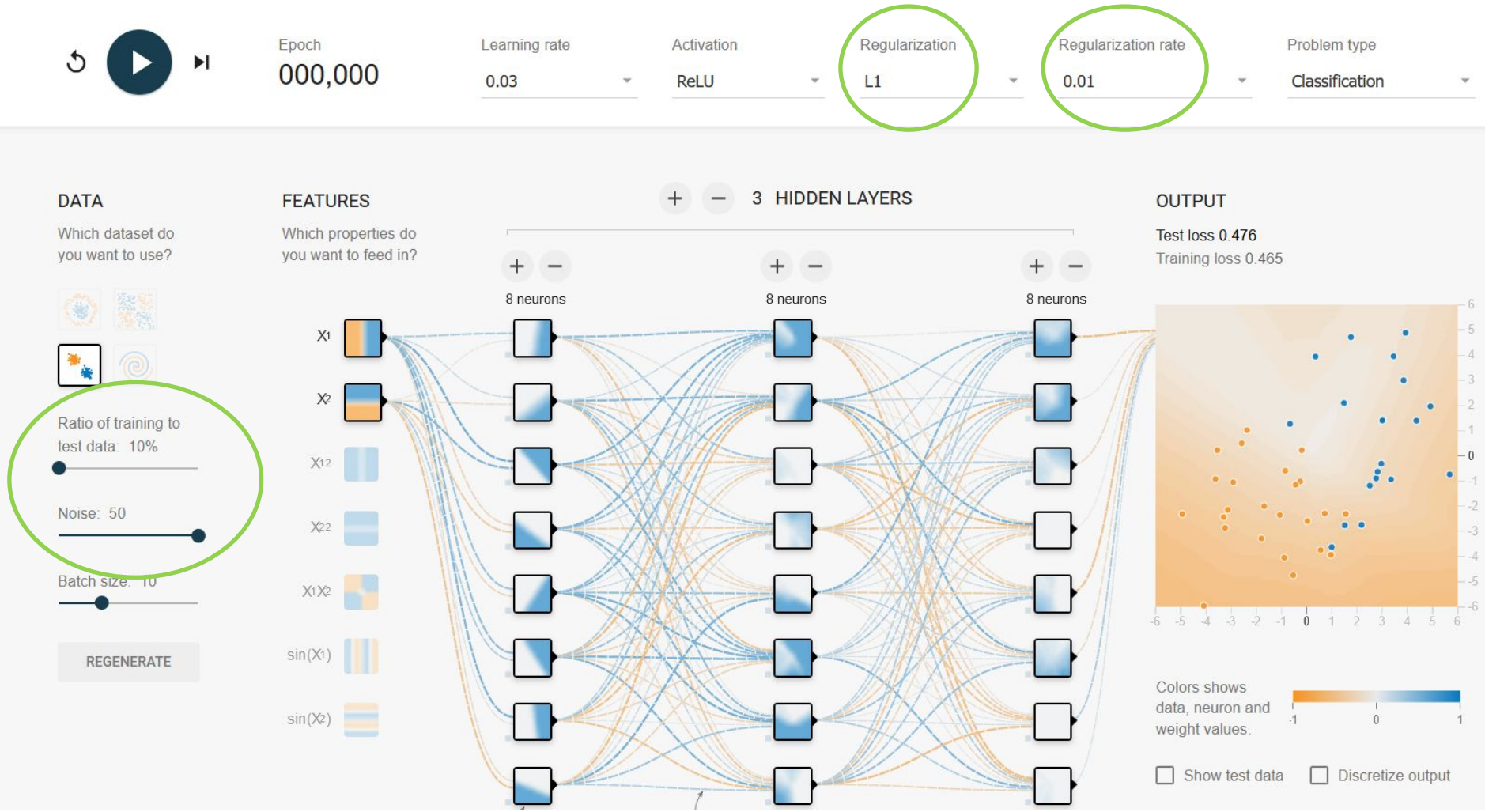
$$\lambda\|\boldsymbol{w}\|^2 = \lambda(w_1^2 + w_2^2 + w_3^2 + w_4^2)$$

- L1-norm regularization:

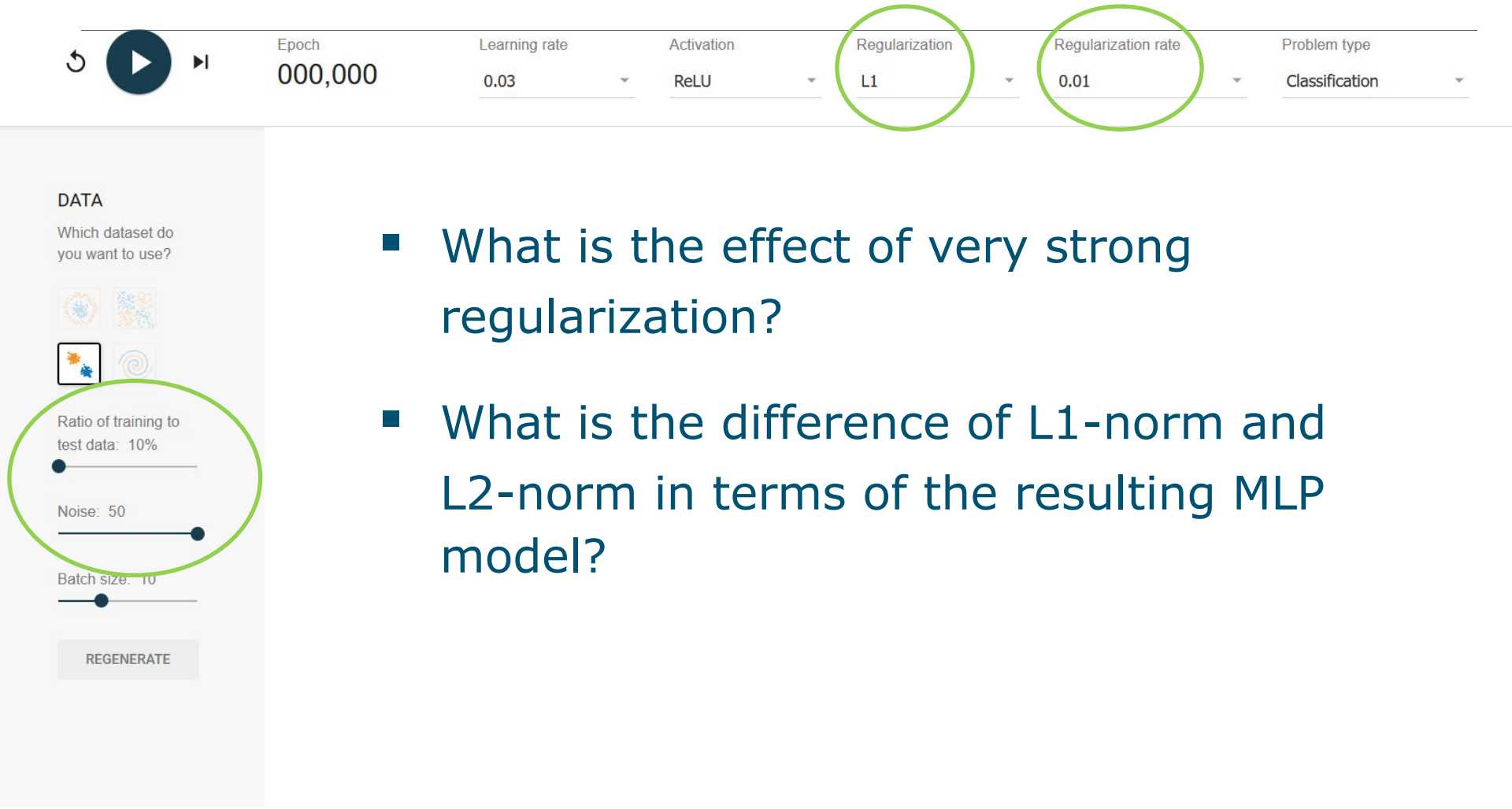
$$\lambda\|\boldsymbol{w}\| = \lambda(|w_1| + |w_2| + |w_3| + |w_4|)$$

http://playground.tensorflow.org/

Regularization with weight decay

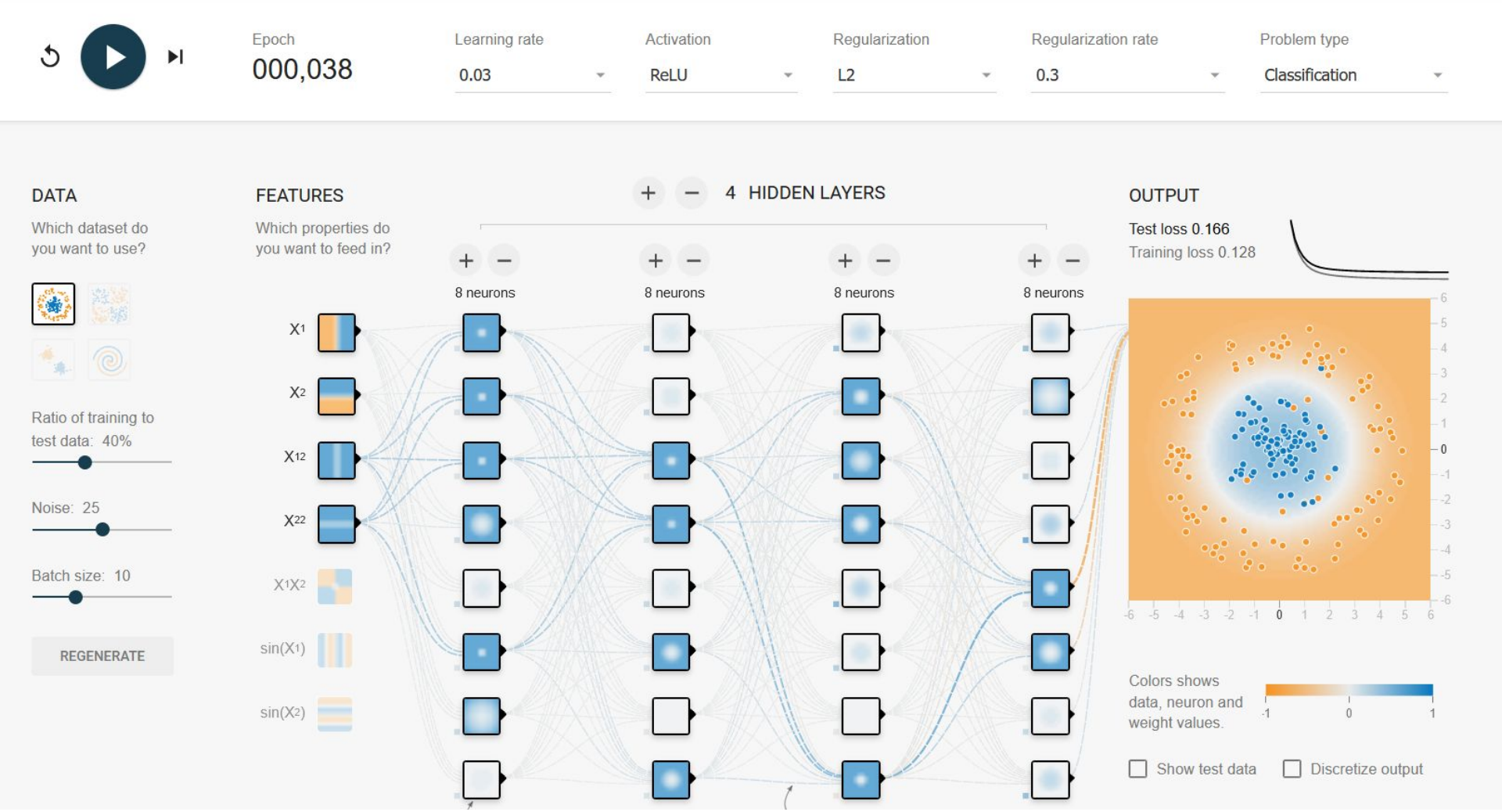


Regularization with weight decay

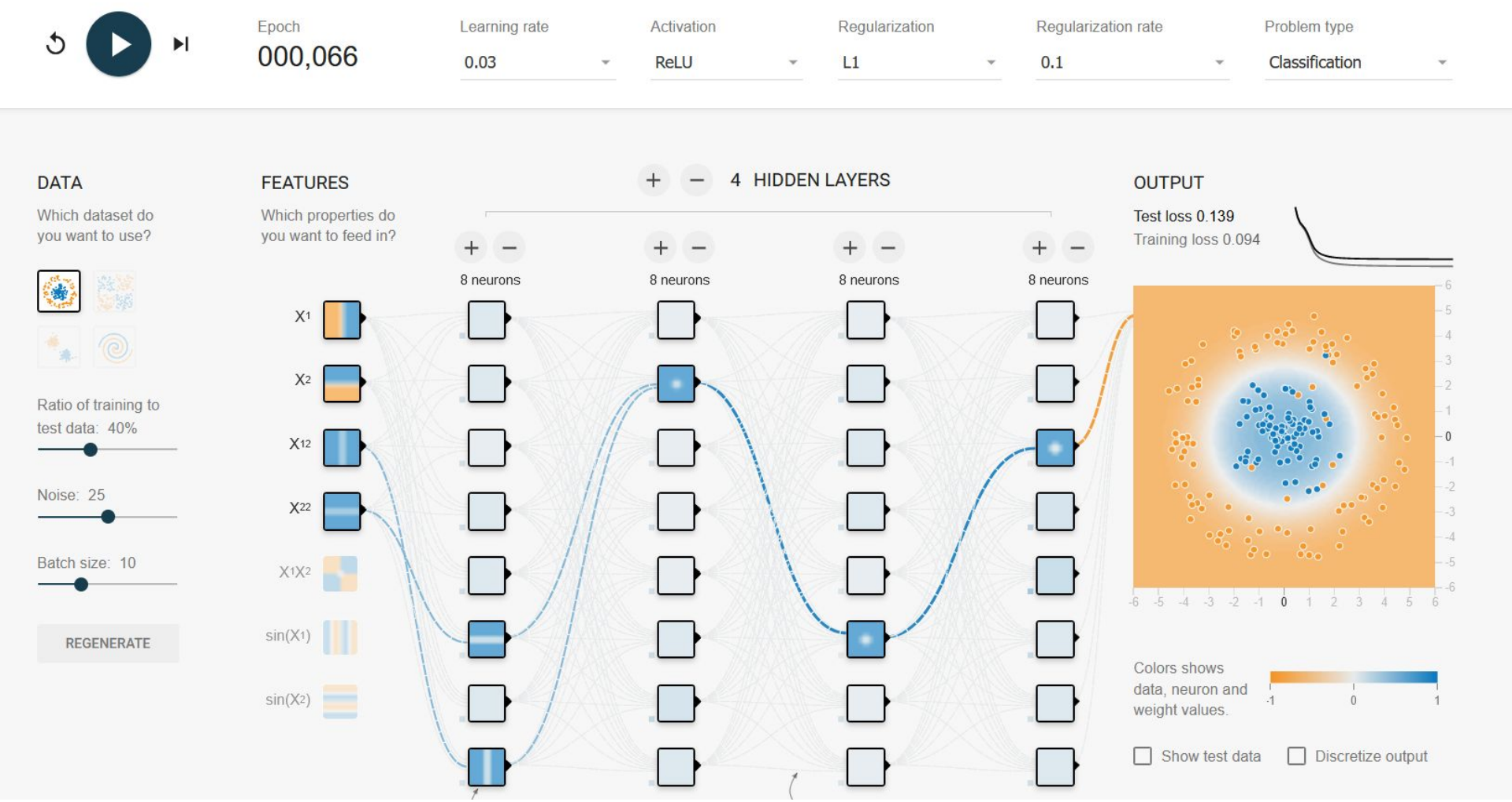


- What is the effect of very strong regularization?
- What is the difference of L1-norm and L2-norm in terms of the resulting MLP model?

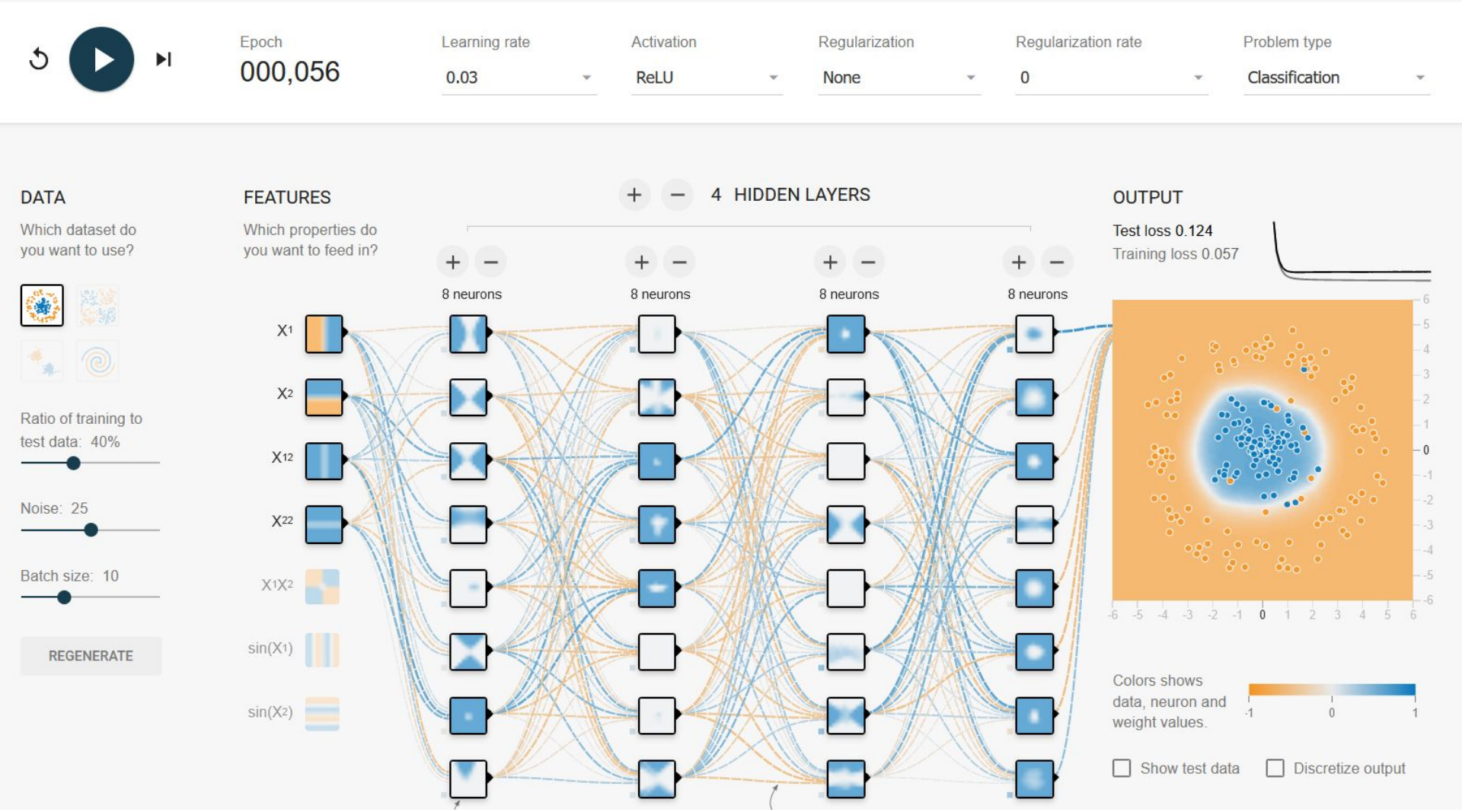
L2 regularization



L1 regularization



No regularization



Further things to explore

Using the same example (with the ring data):

- Do you need a hidden layer? Why?
- What happens if you use a linear activation function?
- What happens if you use too many hidden layers using a sigmoid activation?
- What happens if the learning rate is too small? And too large?