Google Cloud

The Art of ML

Fereshteh Mahvar

Machine Learning on
Google Cloud Platform

The Art of ML

Hyperparameter Tuning

A Pinch of Science

The Science of Neural Networks

Embeddings

Custom Estimator

Learn how to...

Generalize your model

Tune batch size and learning rate for better model performance

Optimize your model

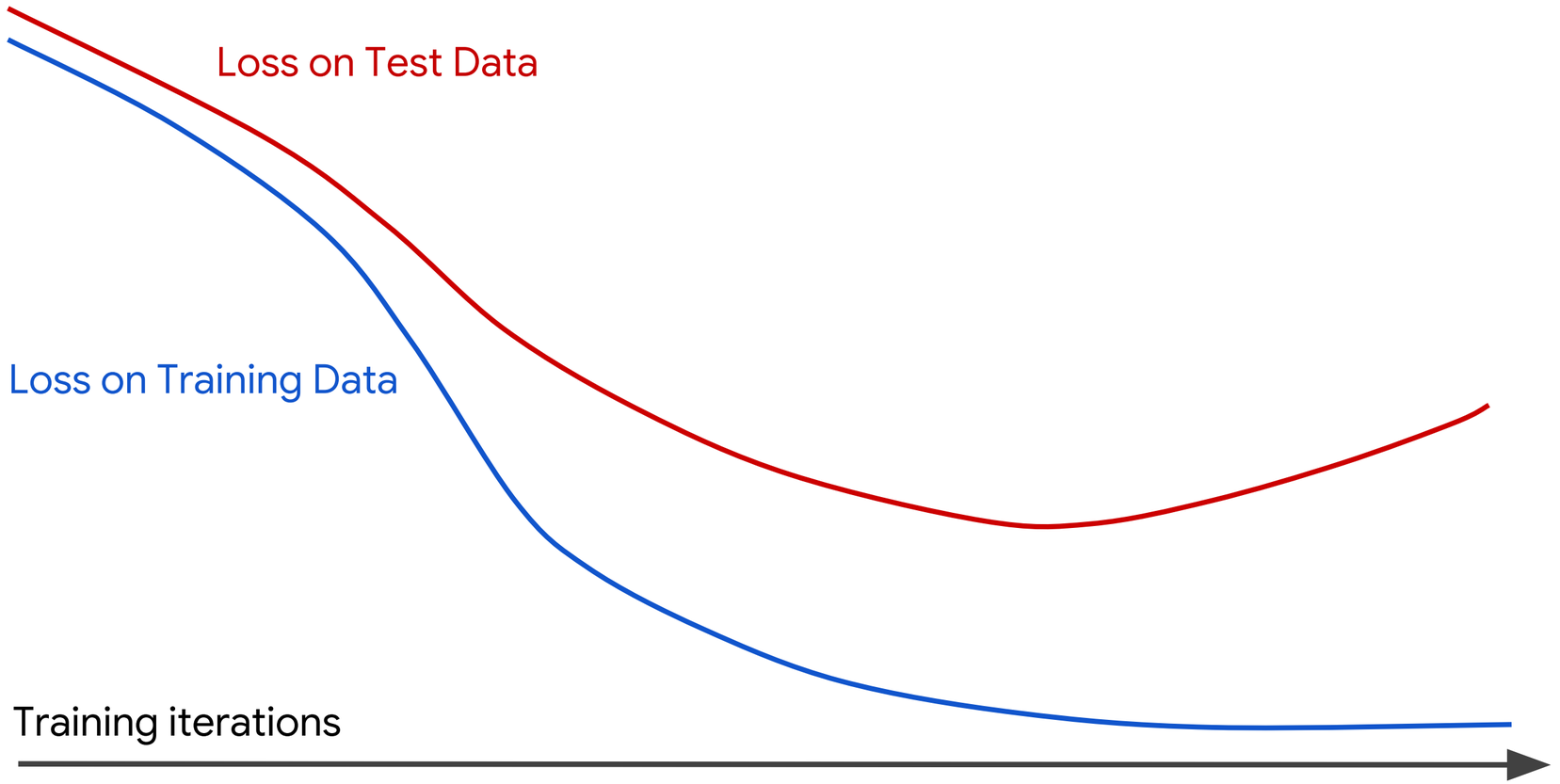
Apply the concepts in TensorFlow code

Google Cloud

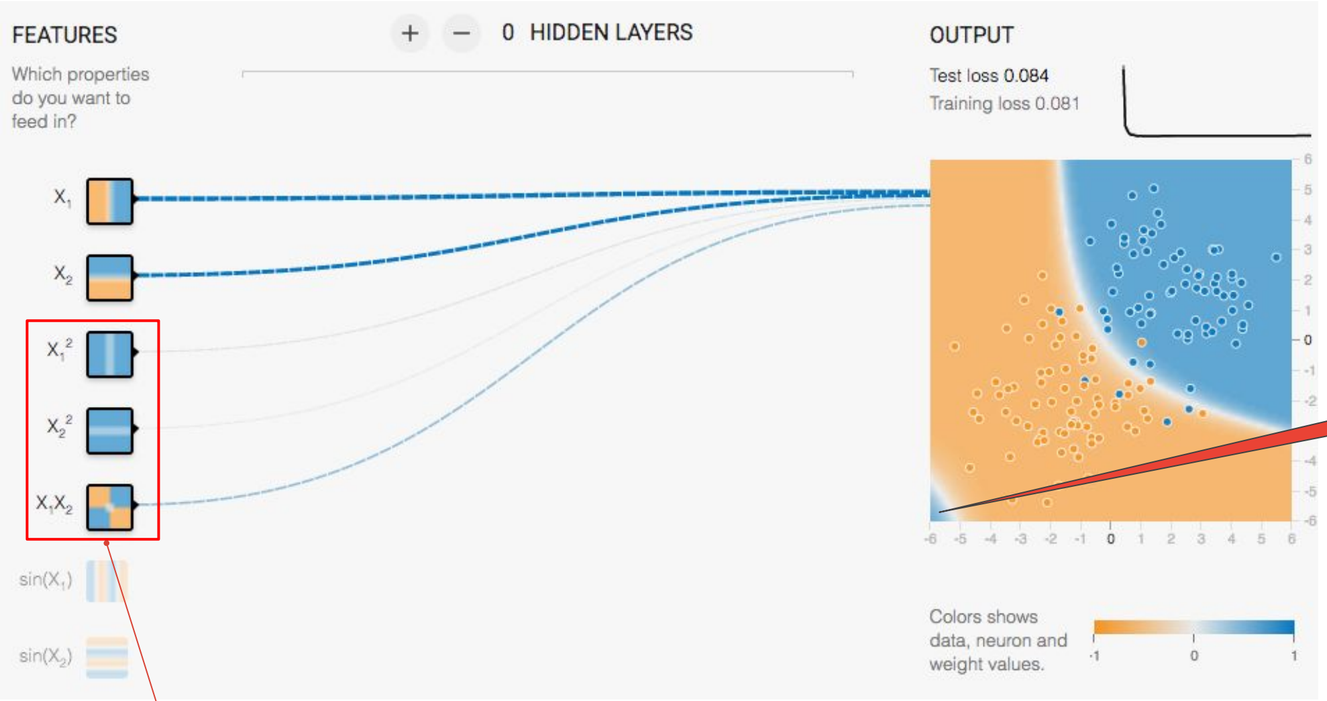
Regularization

Fereshteh Mahvar

What is happening here? How can we address this?



Remember the splotch of blue?
Why does it happen?



https://goo.gl/ofiHCT

Why?

Is the model behavior surprising? What's the issue?

Try removing cross-product features. Does performance improve?

The simpler the better

Don’t cook with every spice in the spice rack!

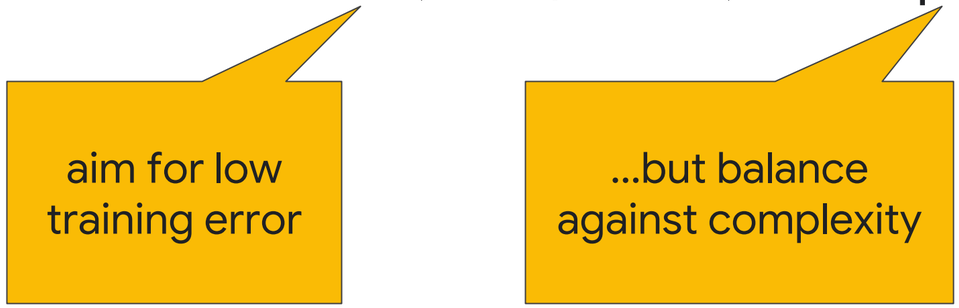
Occam's razor

When presented with competing hypothetical answers to a problem, one should select the one that makes the fewest assumptions. The idea is attributed to William of Ockham (c. 1287–1347).

source:https://en.wikipedia.org/wiki/Occam%27s_razor

Factor in model complexity when calculating error

Minimize: *loss*(Data|Model) + complexity(Model)



Optimal model complexity is data-dependent, so requires hyperparameter tuning.

Regularization is a major field of ML research

Early Stopping

Parameter Norm Penalties

- L1 regularization
- L2 regularization
- Max-norm regularization

Dataset Augmentation

Noise Robustness

Sparse Representations

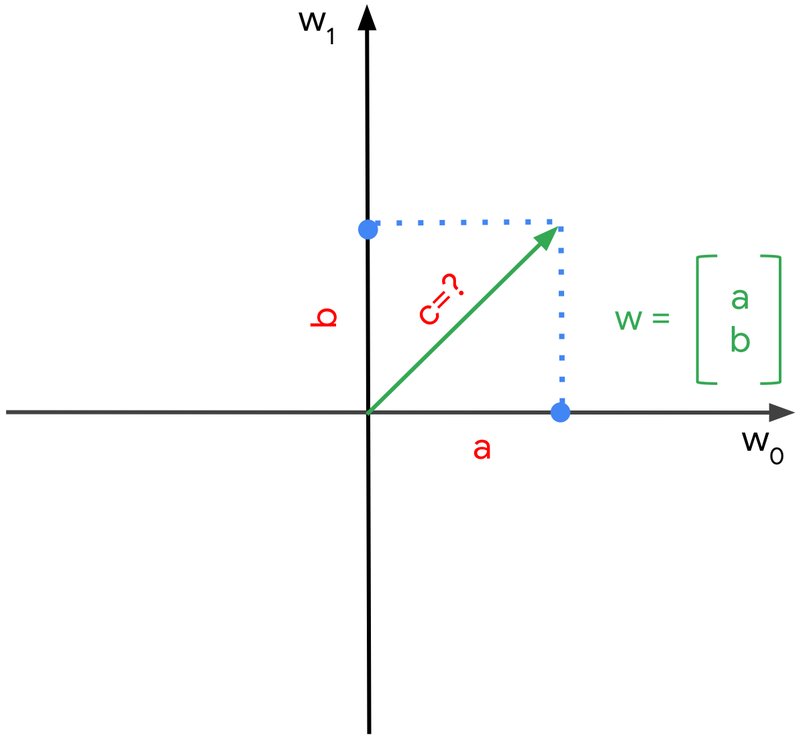
. . .

We will look into these methods.

How can we measure model complexity?

L2 vs. L1 Norm

$$w = \left[w_0, w_1, \ldots, w_n\right]^T$$

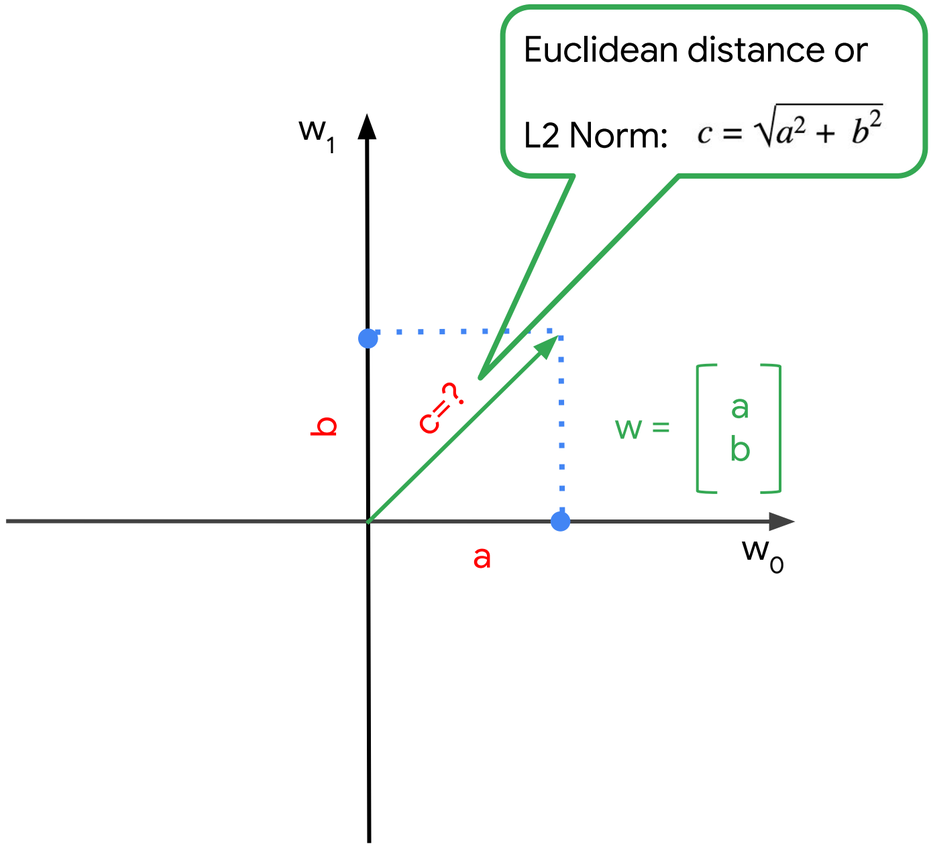


L2 vs. L1 Norm

$$w = [w_0, w_1, \ldots, w_n]^T$$

$$\|w\|_2 = (w_0^2 + w_1^2 + \ldots + w_n^2)^{1/2}$$

$$\|w\|_1 = (|w_0| + |w_1| + \ldots + |w_n|)$$

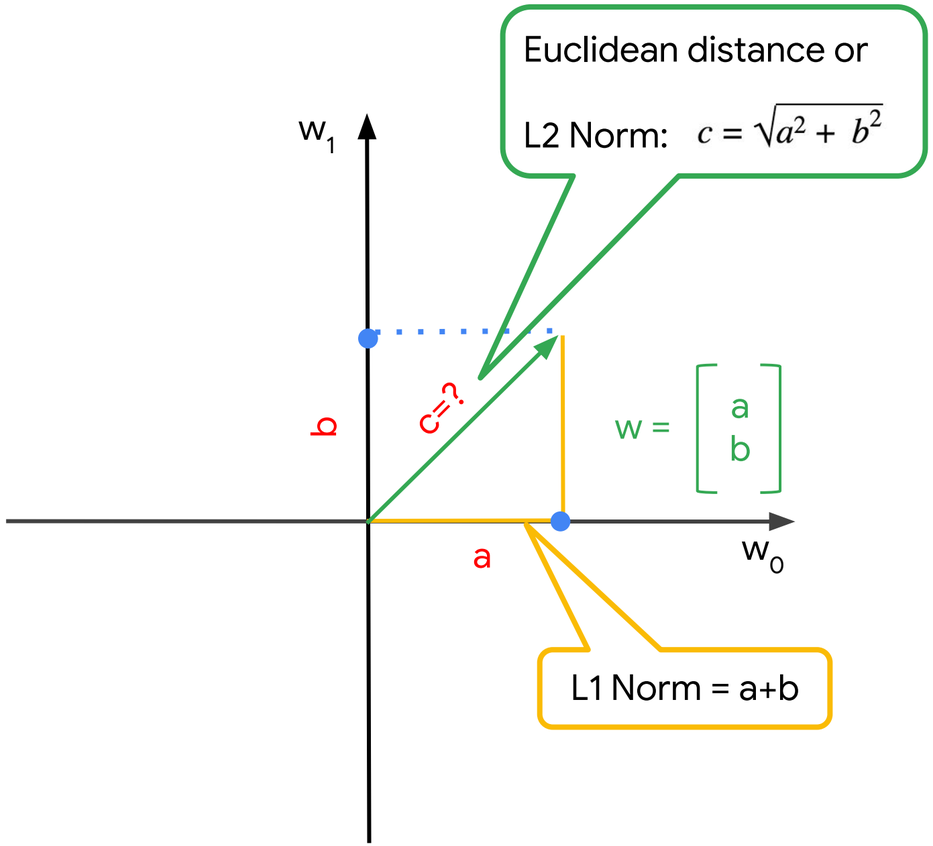


L2 vs. L1 Norm

$$w = [w_0, w_1, \ldots, w_n]^T$$

L2 Norm

$$\|w\|_2 = (w_0^2 + w_1^2 + \ldots + w_n^2)^{1/2}$$



L2 vs. L1 Norm

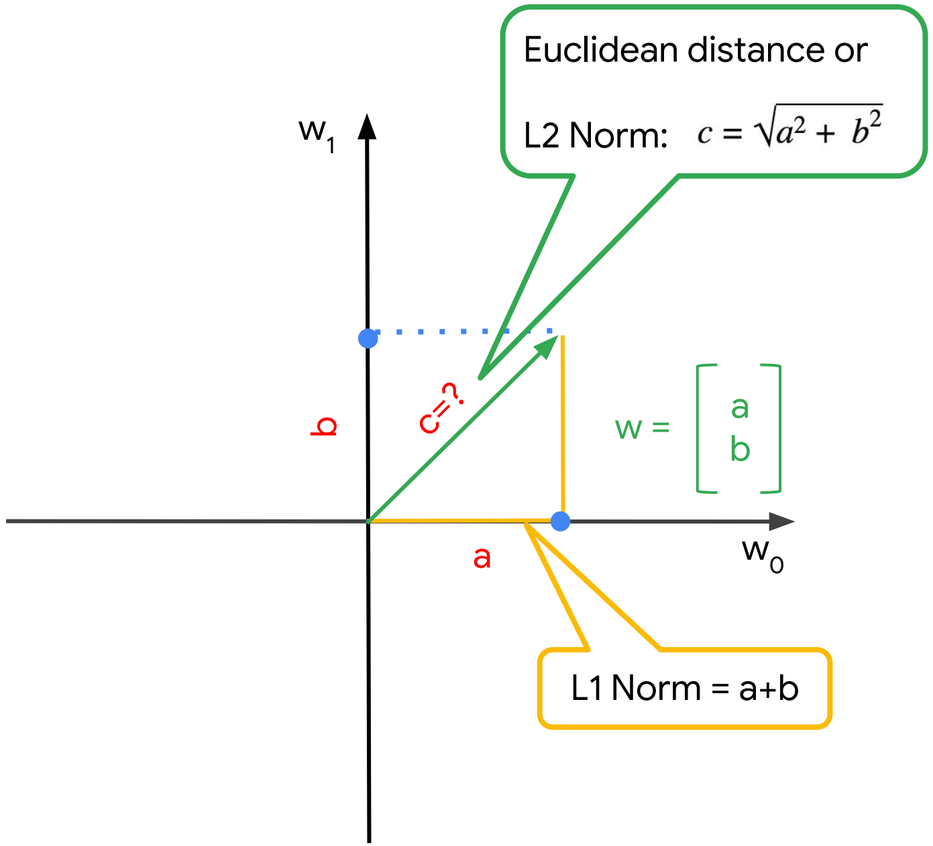
$$w = [w_0, w_1, \ldots, w_n]^T$$

L2 Norm

$$\|w\|_2 = (w_0^2 + w_1^2 + \ldots + w_n^2)^{1/2}$$

L1 Norm

$$\|w\|_1 = (|w_0| + |w_1| + \ldots + |w_n|)$$

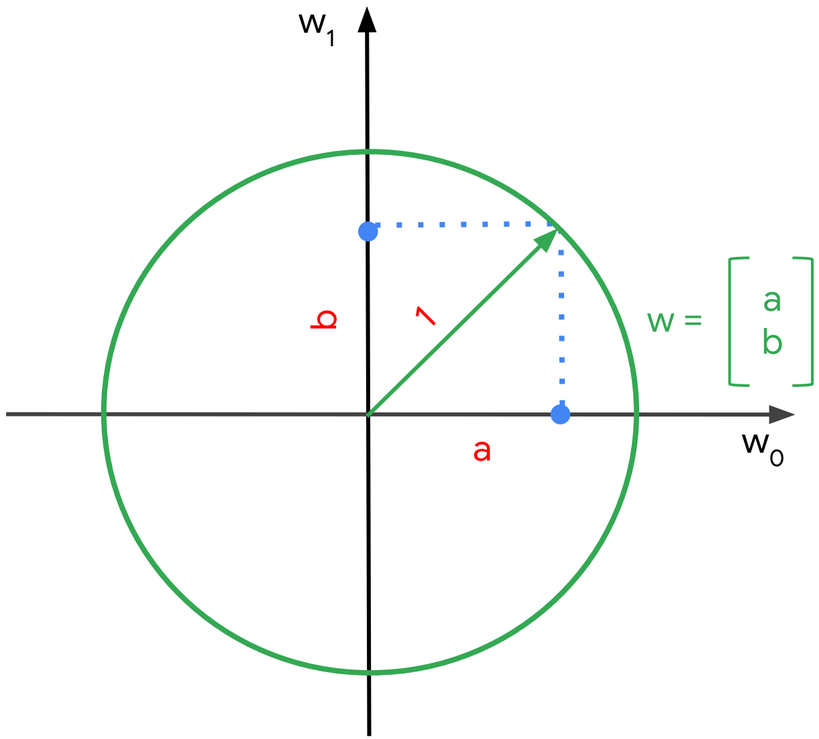


L2 vs. L1 Norm

$$w = \left[w_0, w_1, \ldots, w_n\right]^T$$

L2 Norm

$$\|w\|_2 = (w_0{}^2 + w_1{}^2 + \ldots + w_n{}^2)^{1/2}$$



L2 vs. L1 Norm

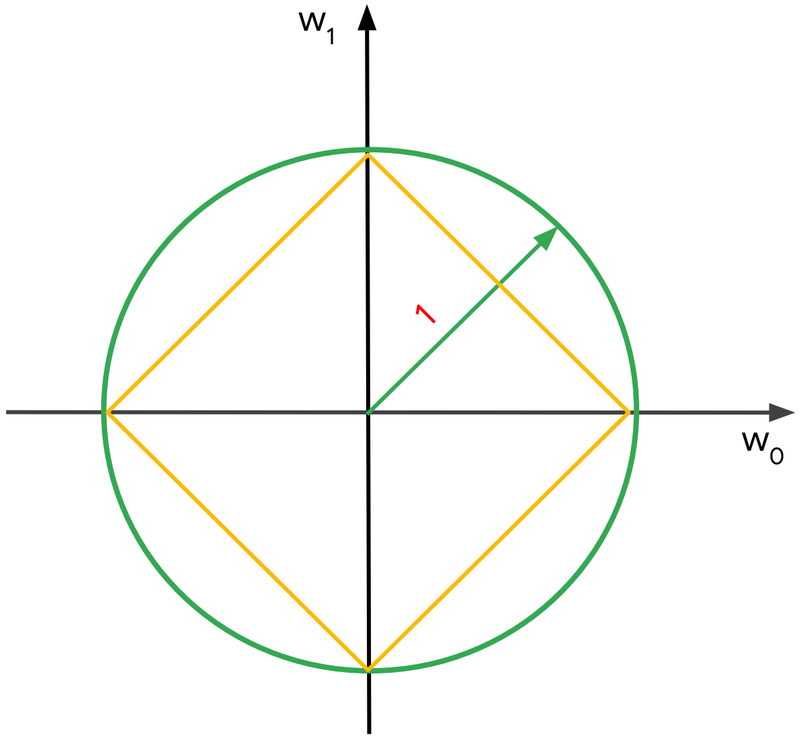
$$w = [w_0, w_1, \ldots, w_n]^T$$

L2 Norm

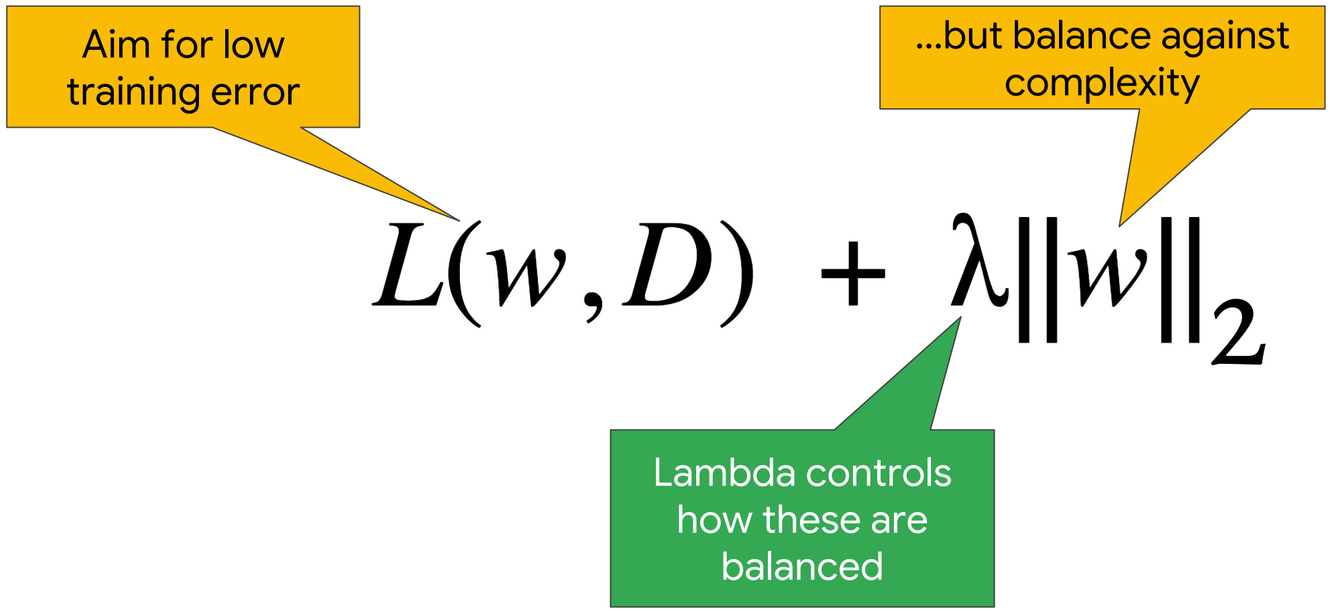
$$\|w\|_2 = (w_0^2 + w_1^2 + \ldots + w_n^2)^{1/2}$$

L1 Norm

$$\|w\|_1 = (|w_0| + |w_1| + \ldots + |w_n|)$$



In L2 regularization, complexity of model is defined by the L2 norm of the weight vector



In L1 regularization, complexity of model is defined by the L1 norm of the weight vector

$$L(w, D) + \lambda \|w\|_1$$

L1 regularization can be used as a feature selection mechanism.

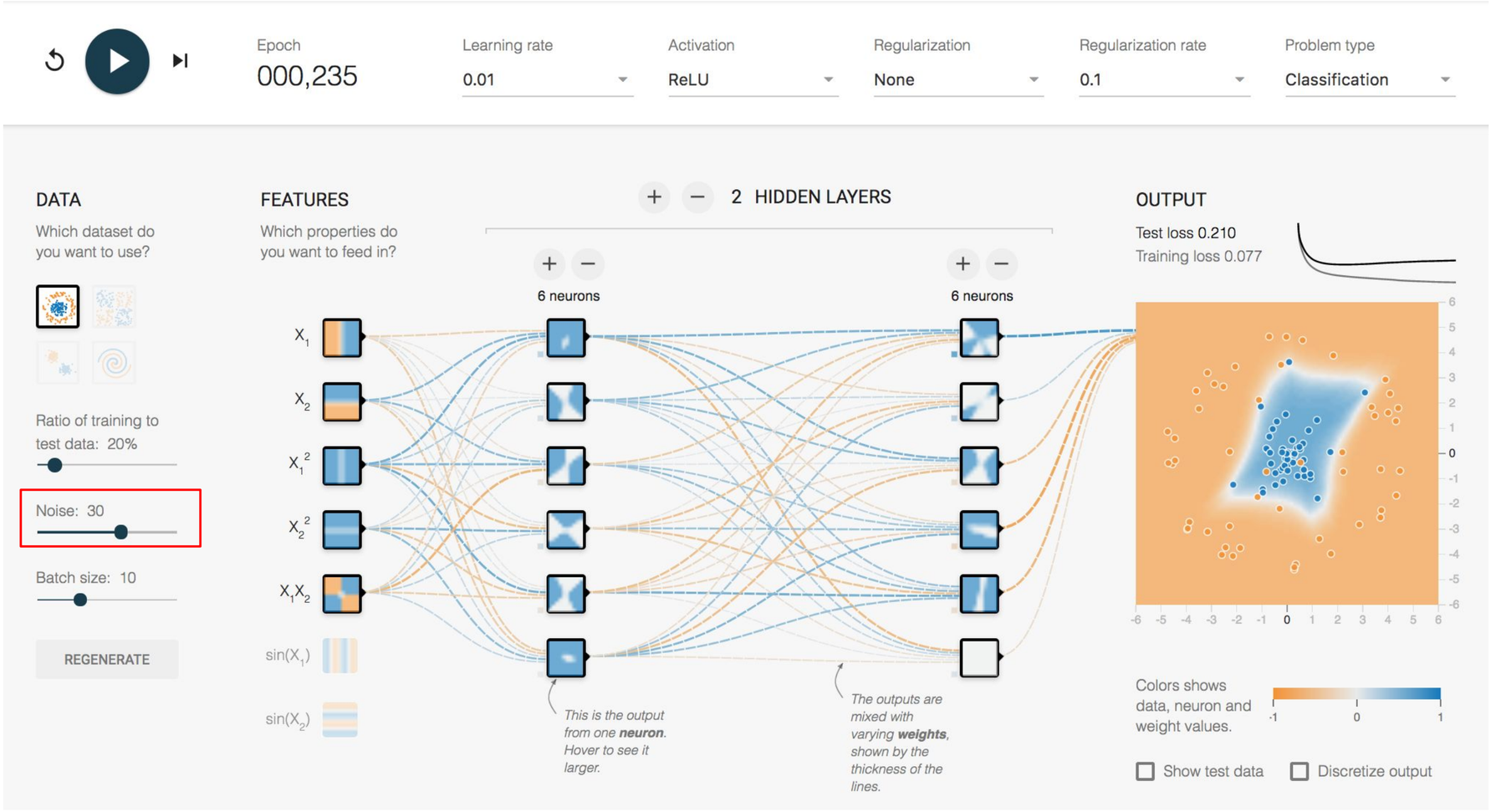
Lab

Regularization

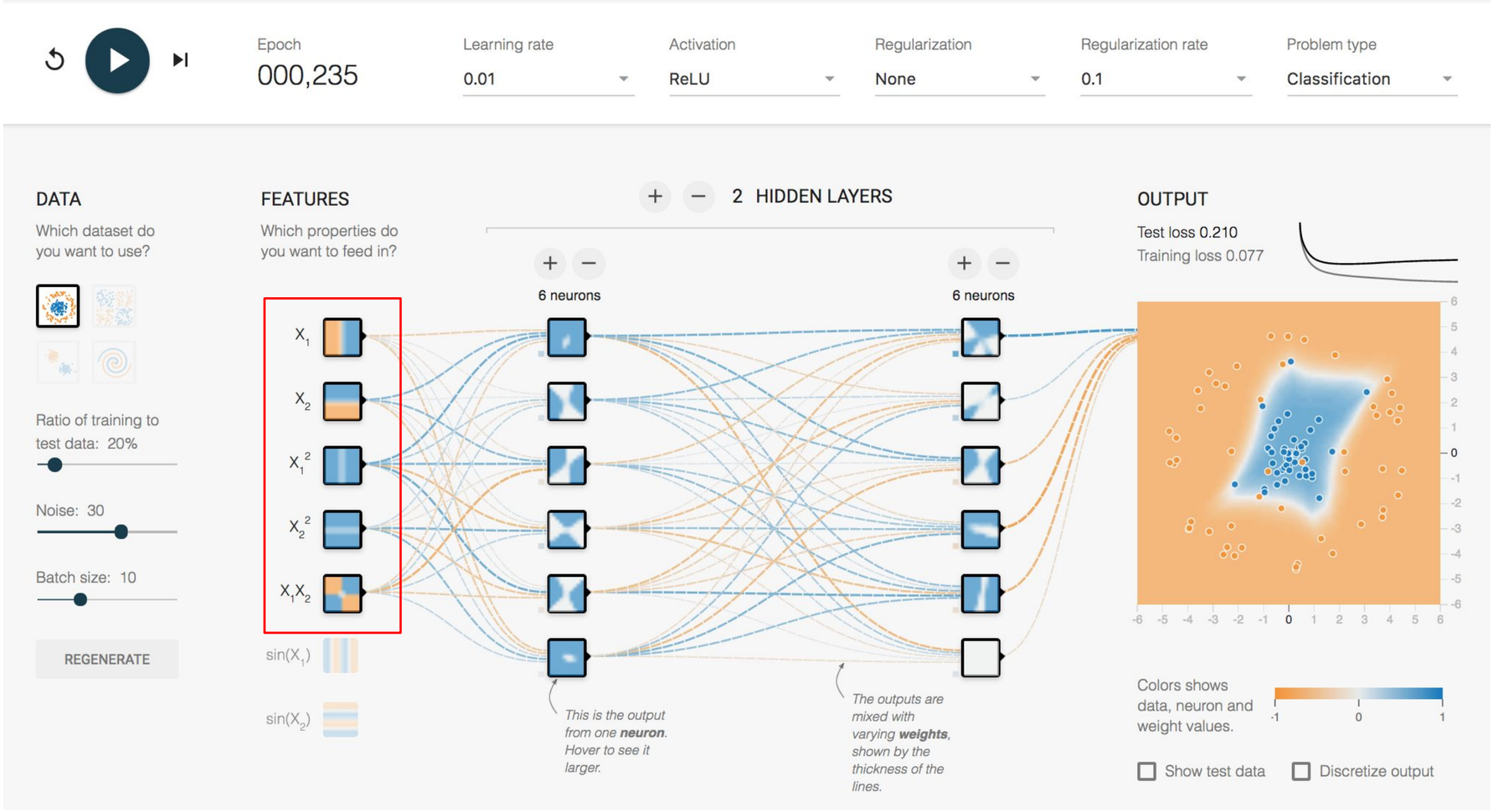
Fereshteh Mahvar

http://goo.gl/4oA1WW

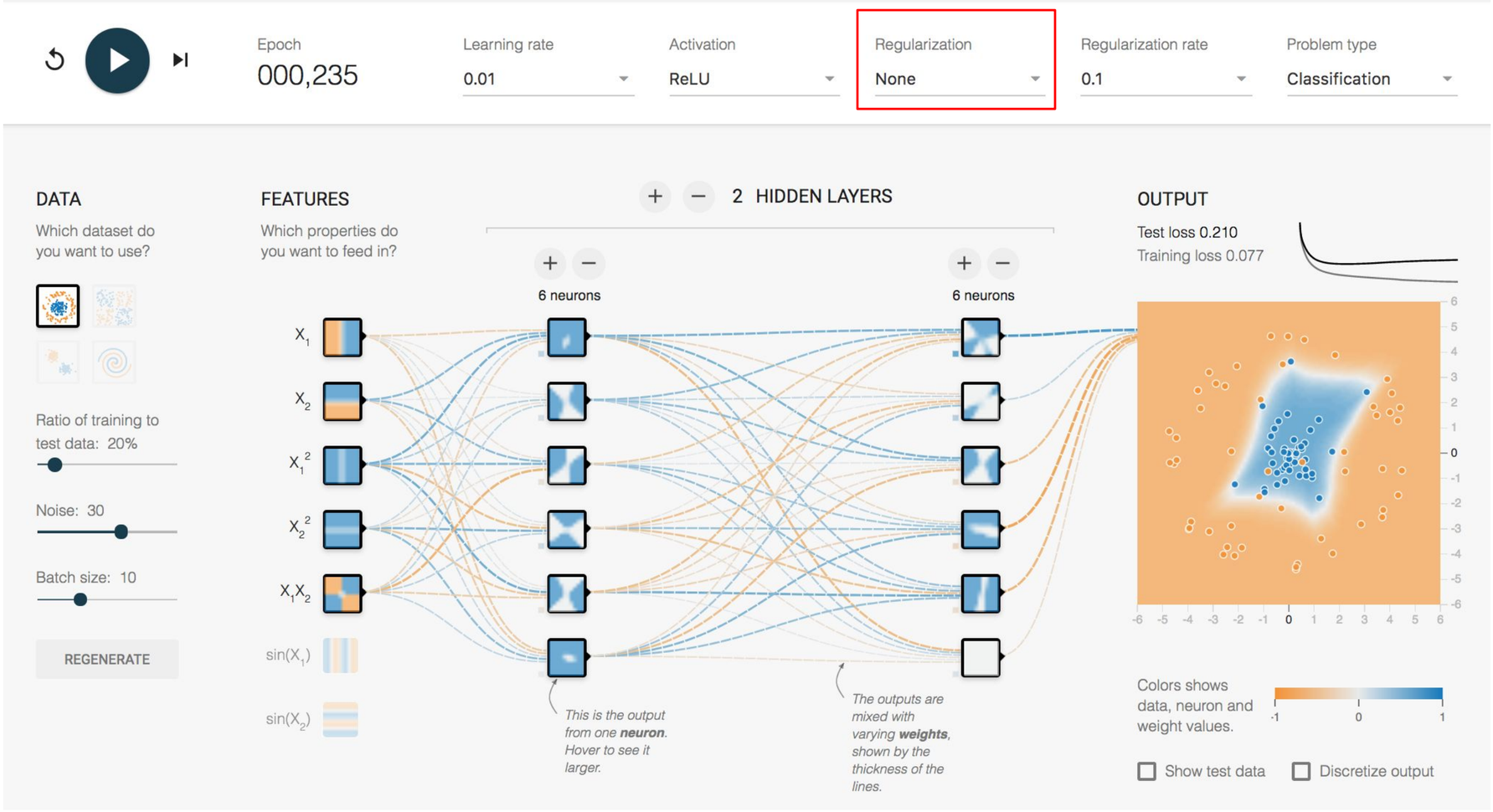
How can you define model complexity?



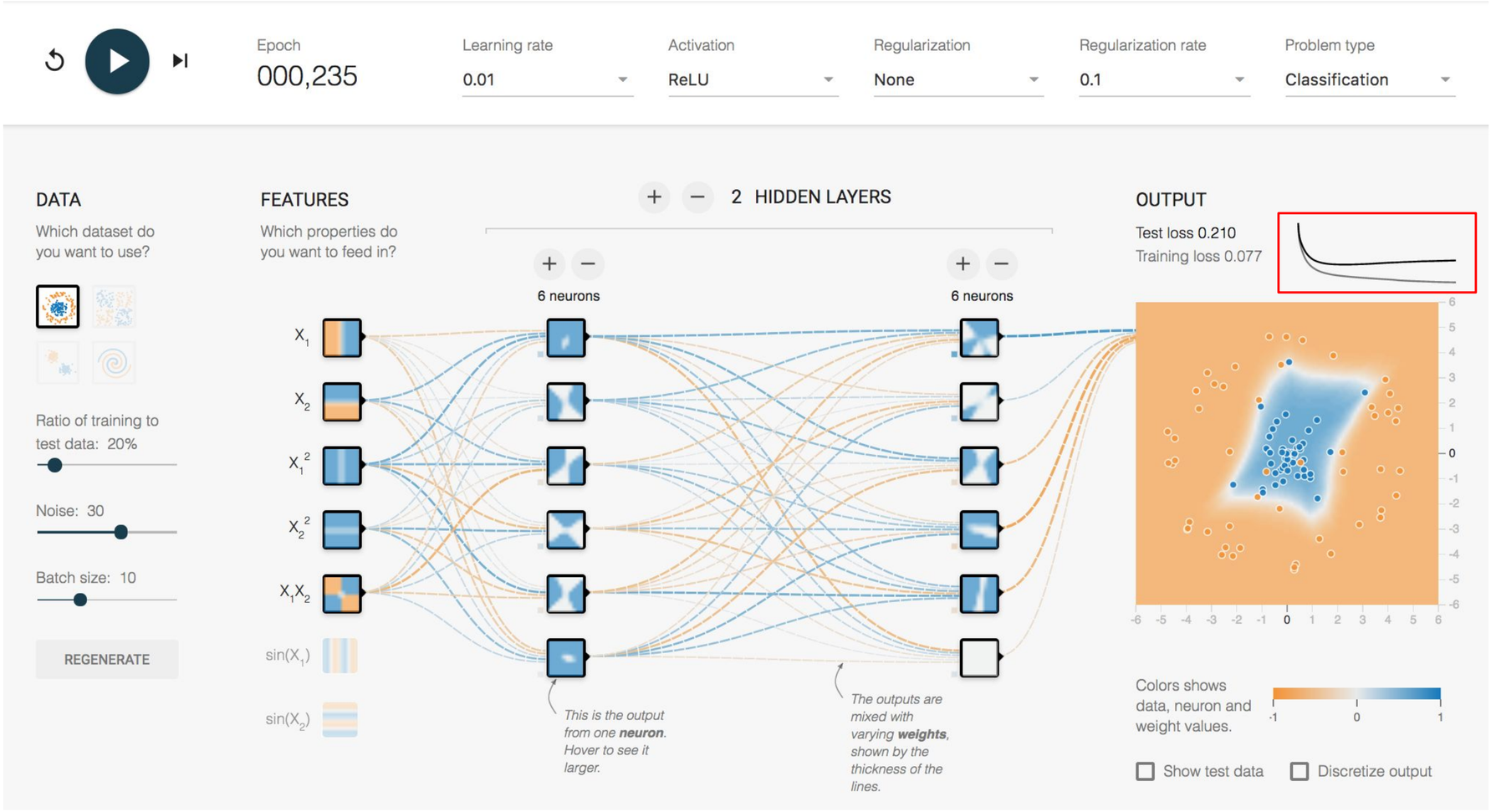
How can you define model complexity?



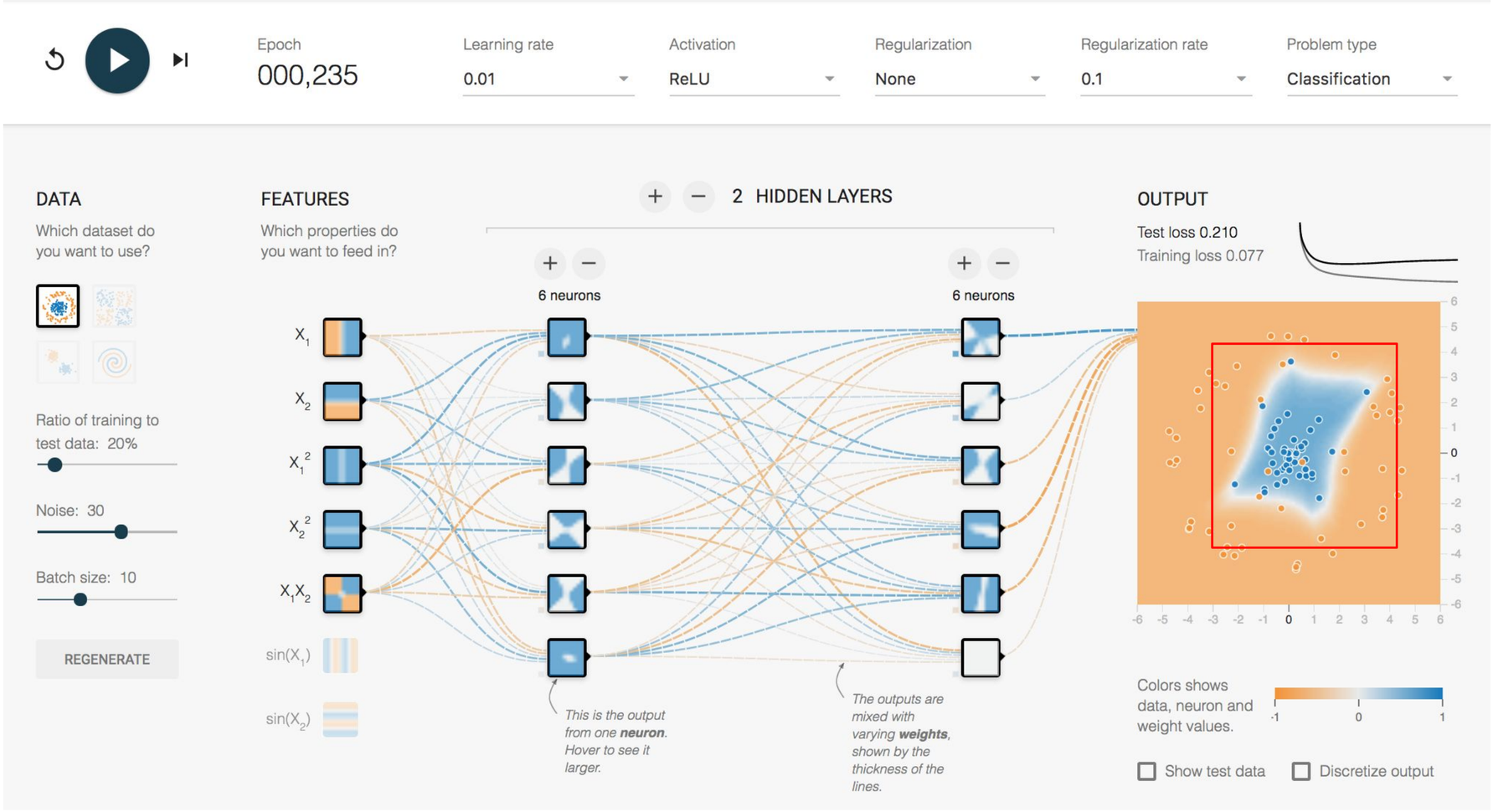
How can you define model complexity?



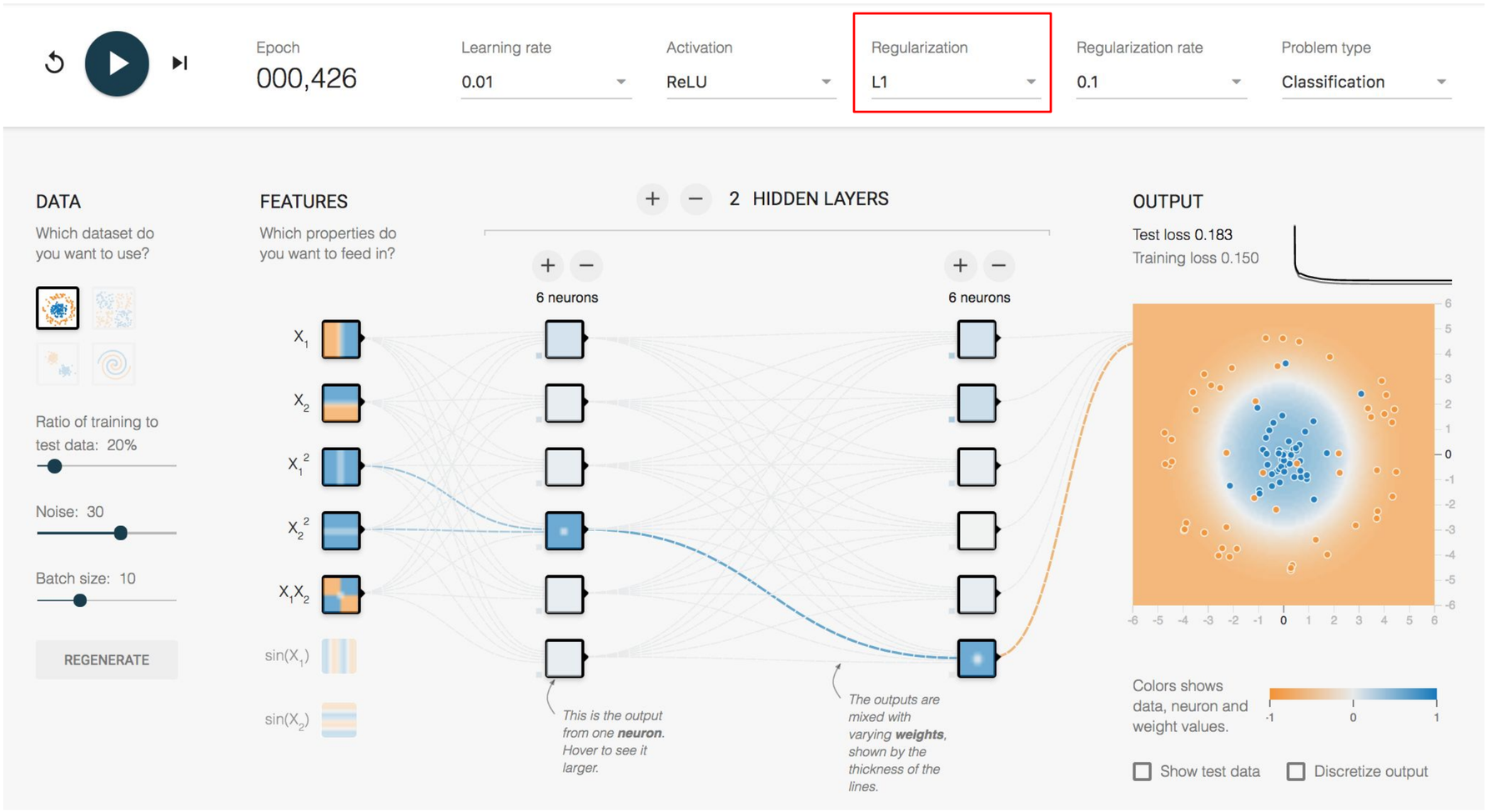
How can you define model complexity?



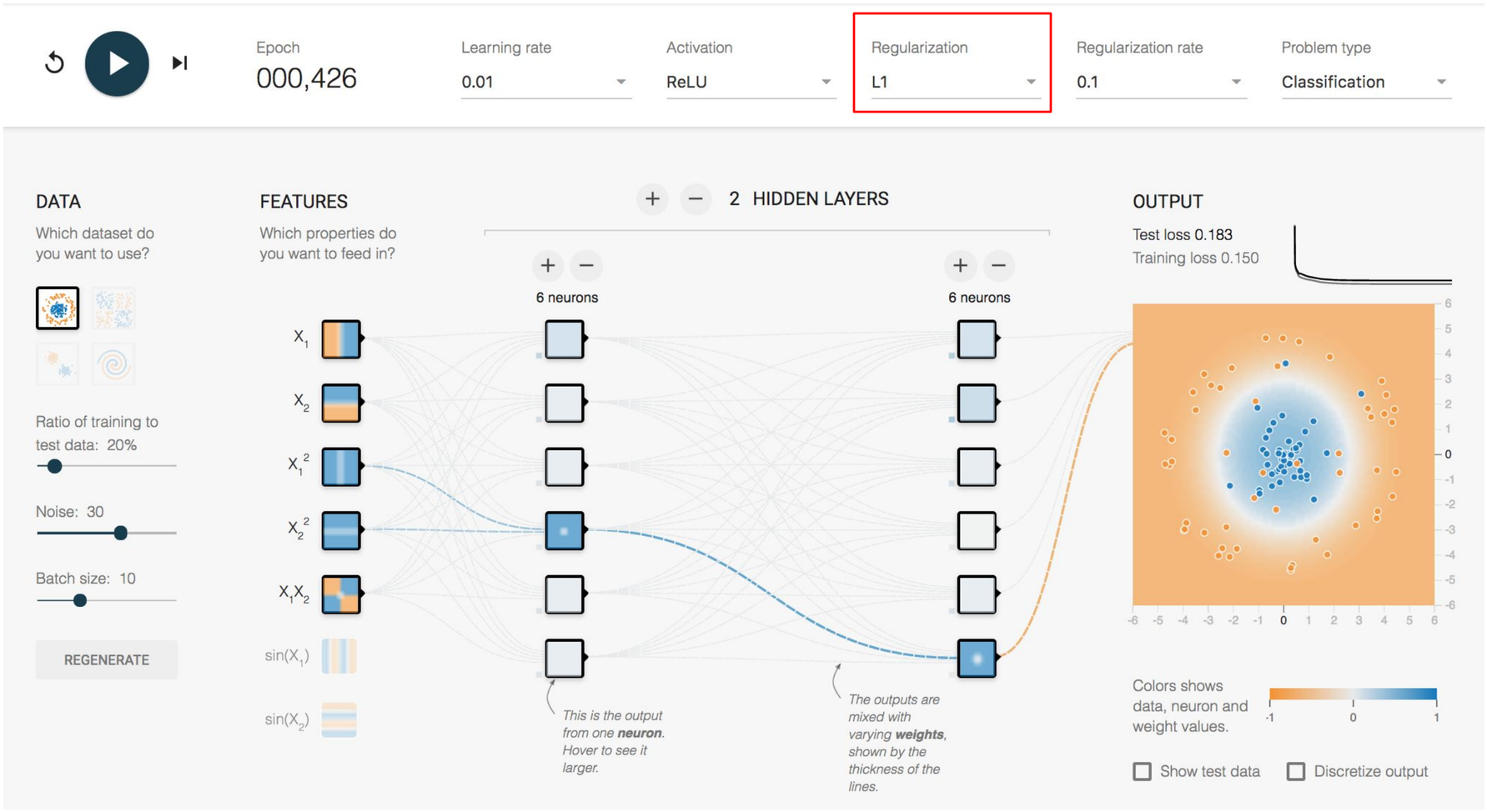
How can you define model complexity?



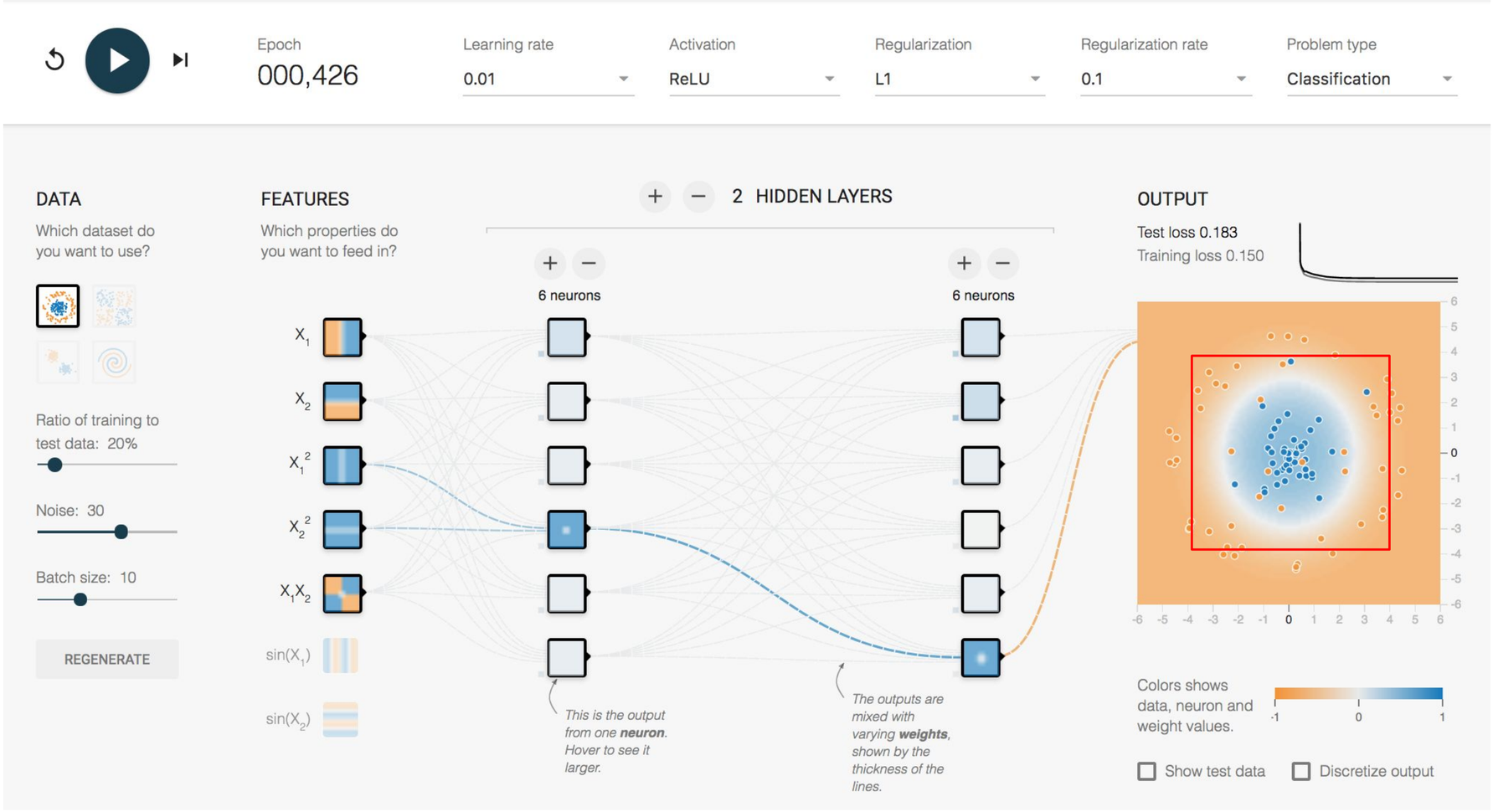
How can you define model complexity?



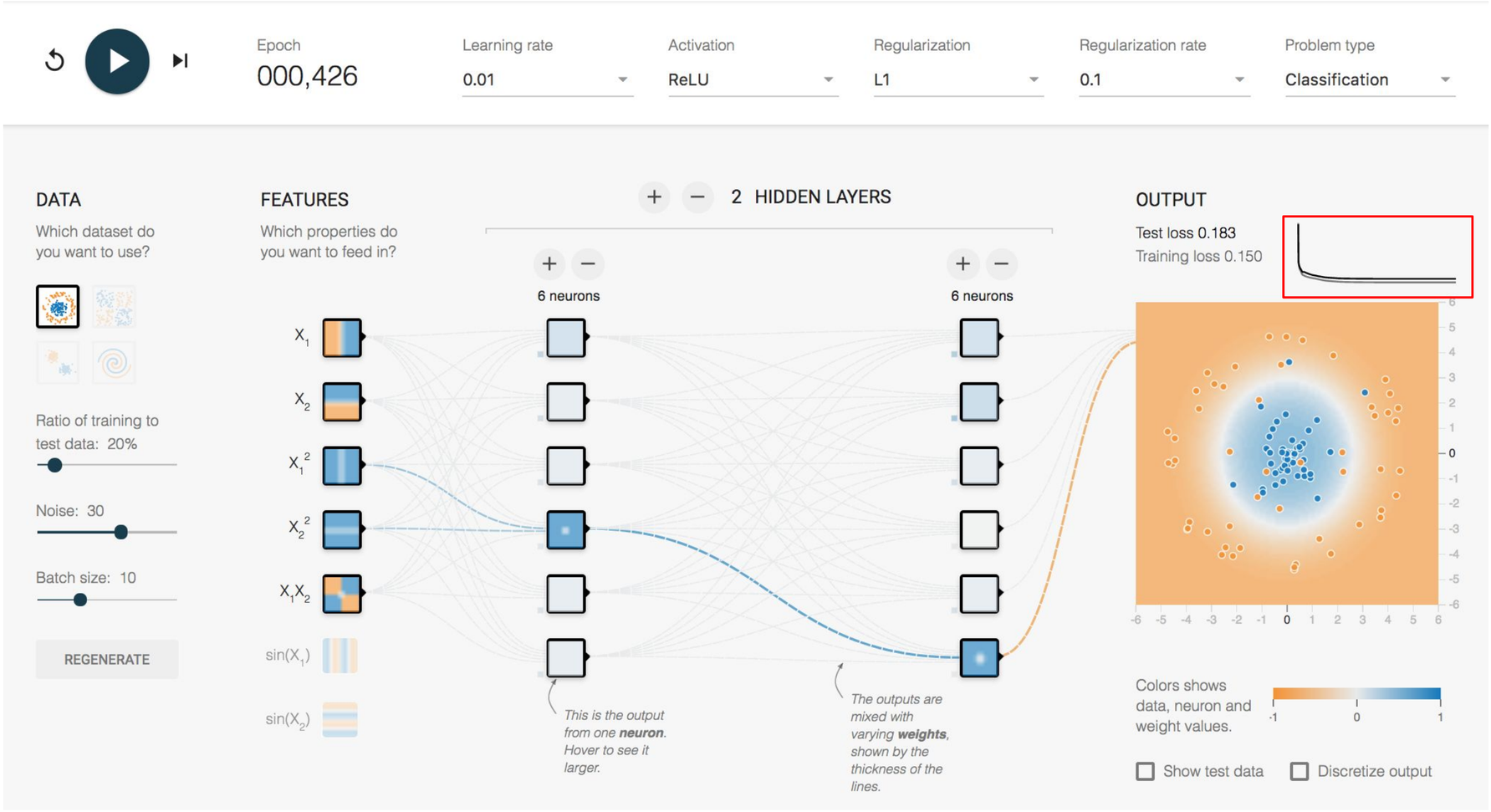
How can you define model complexity?



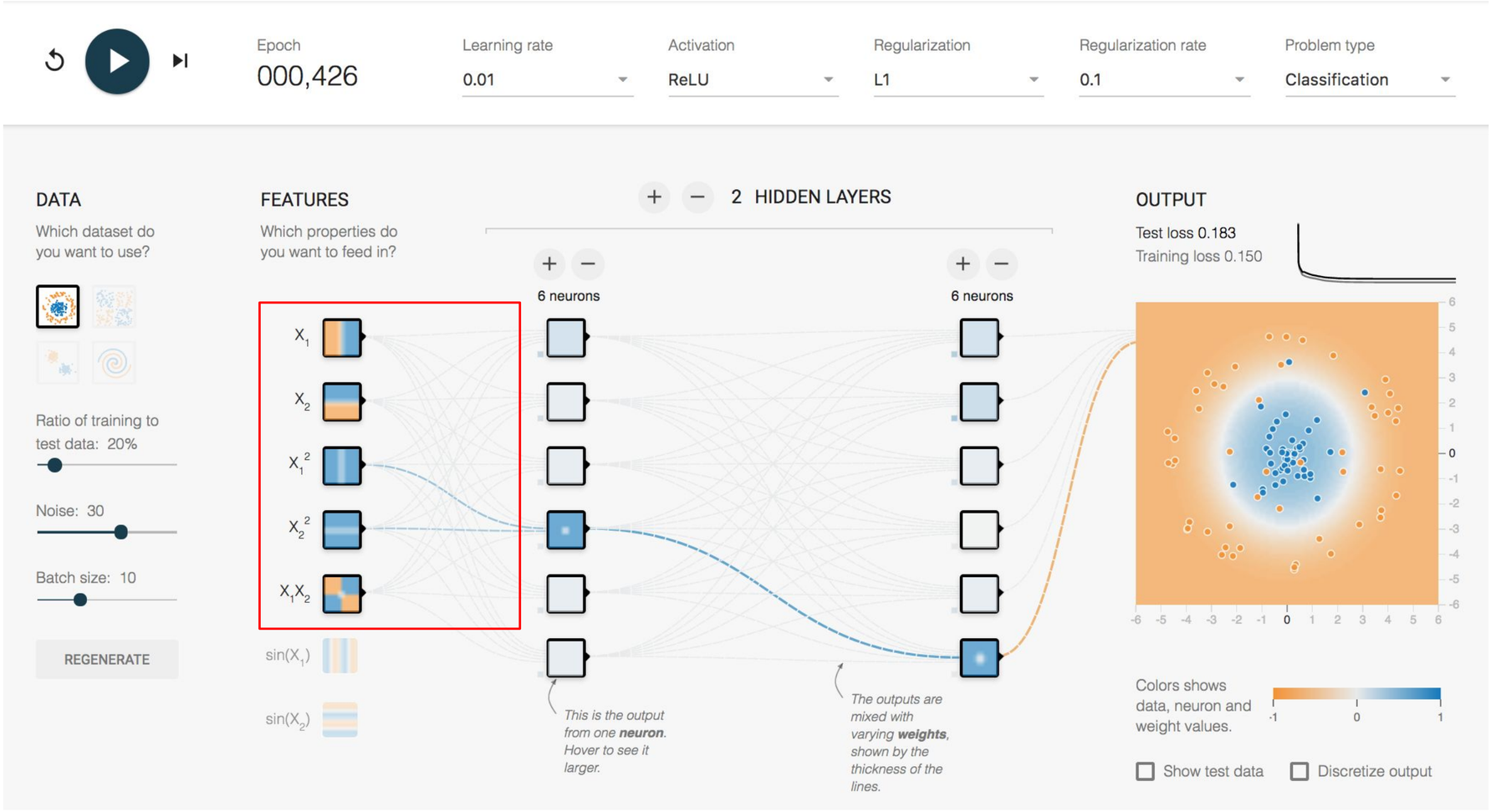
How can you define model complexity?



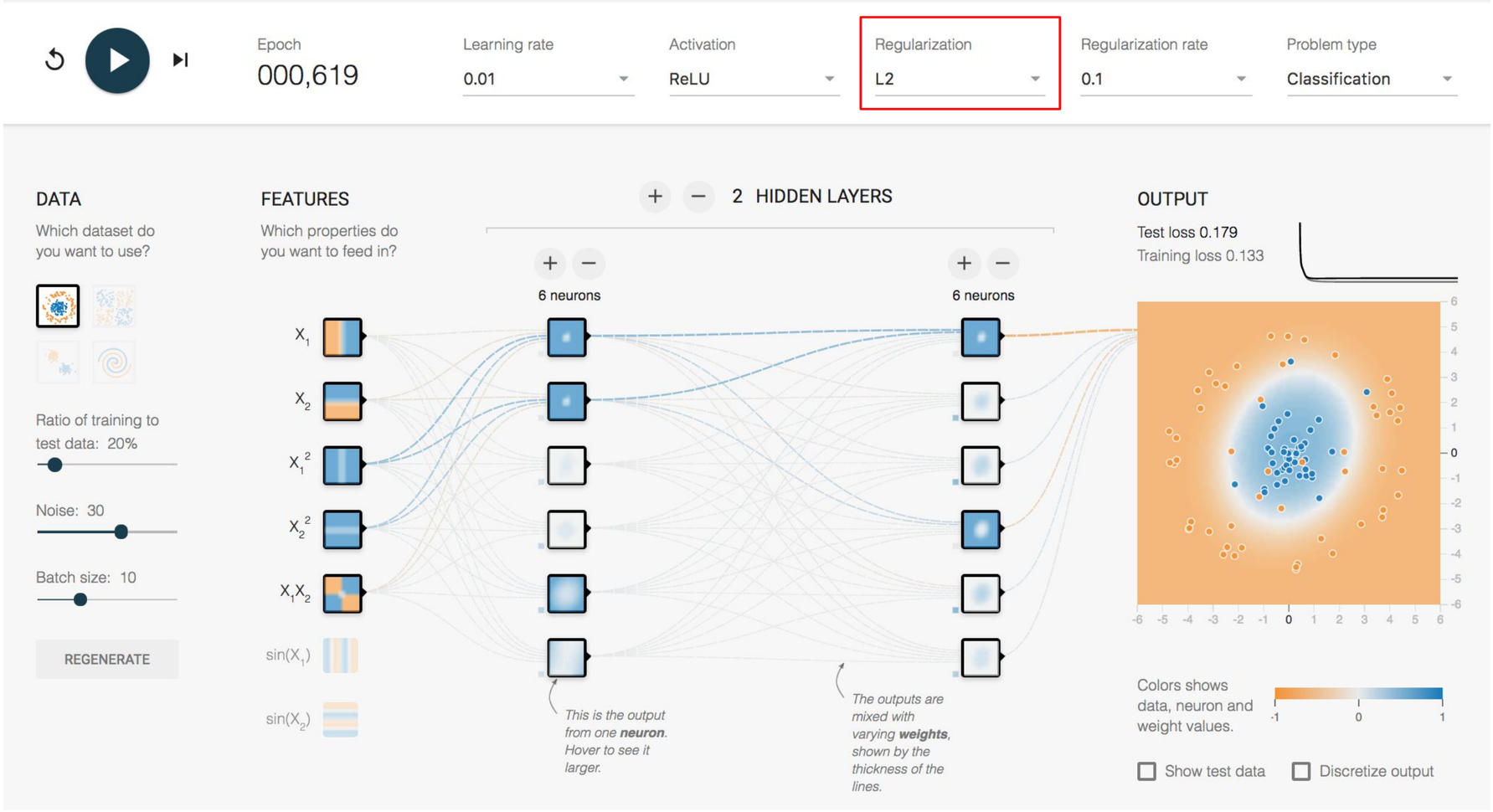
How can you define model complexity?



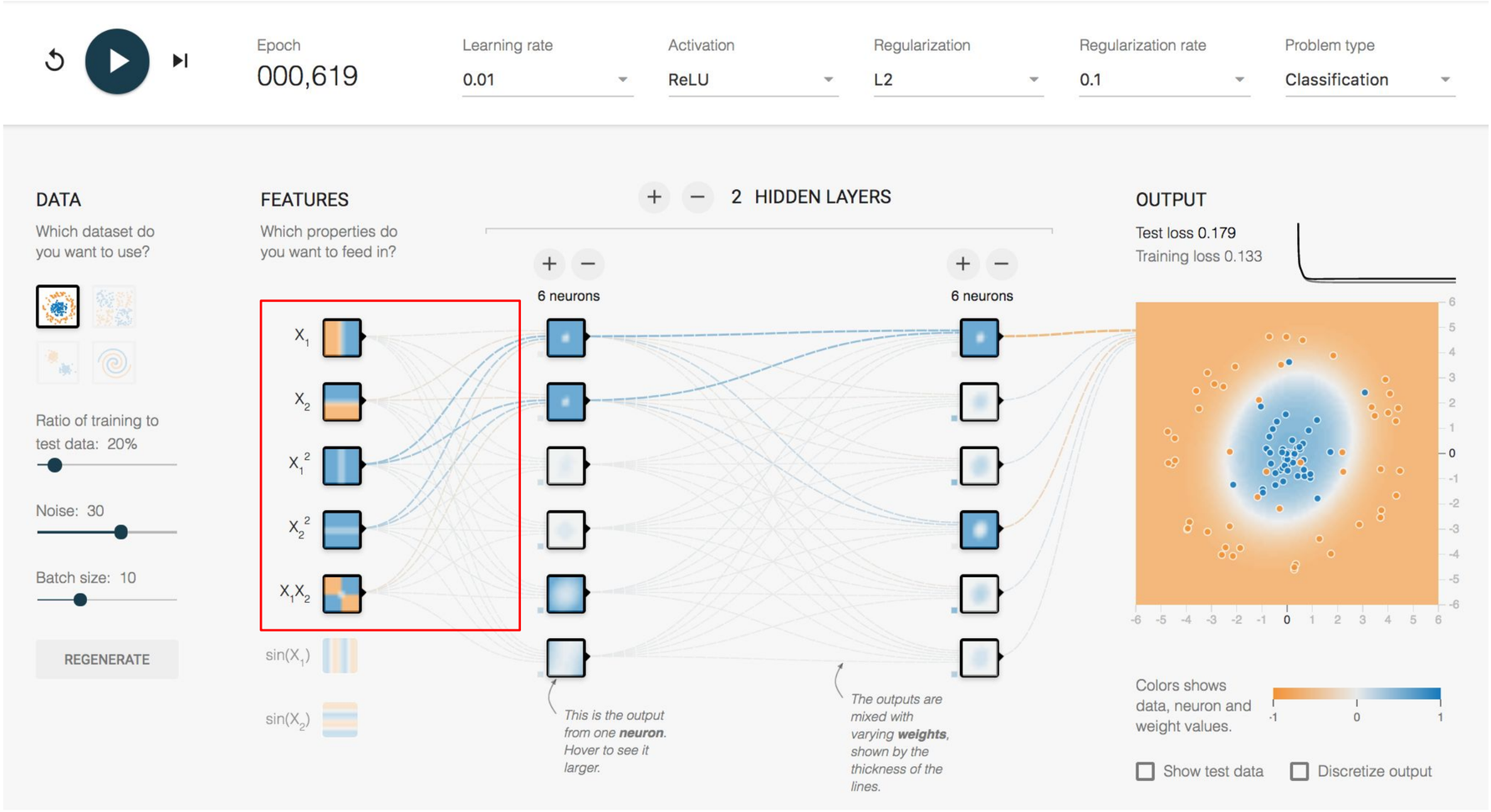
How can you define model complexity?



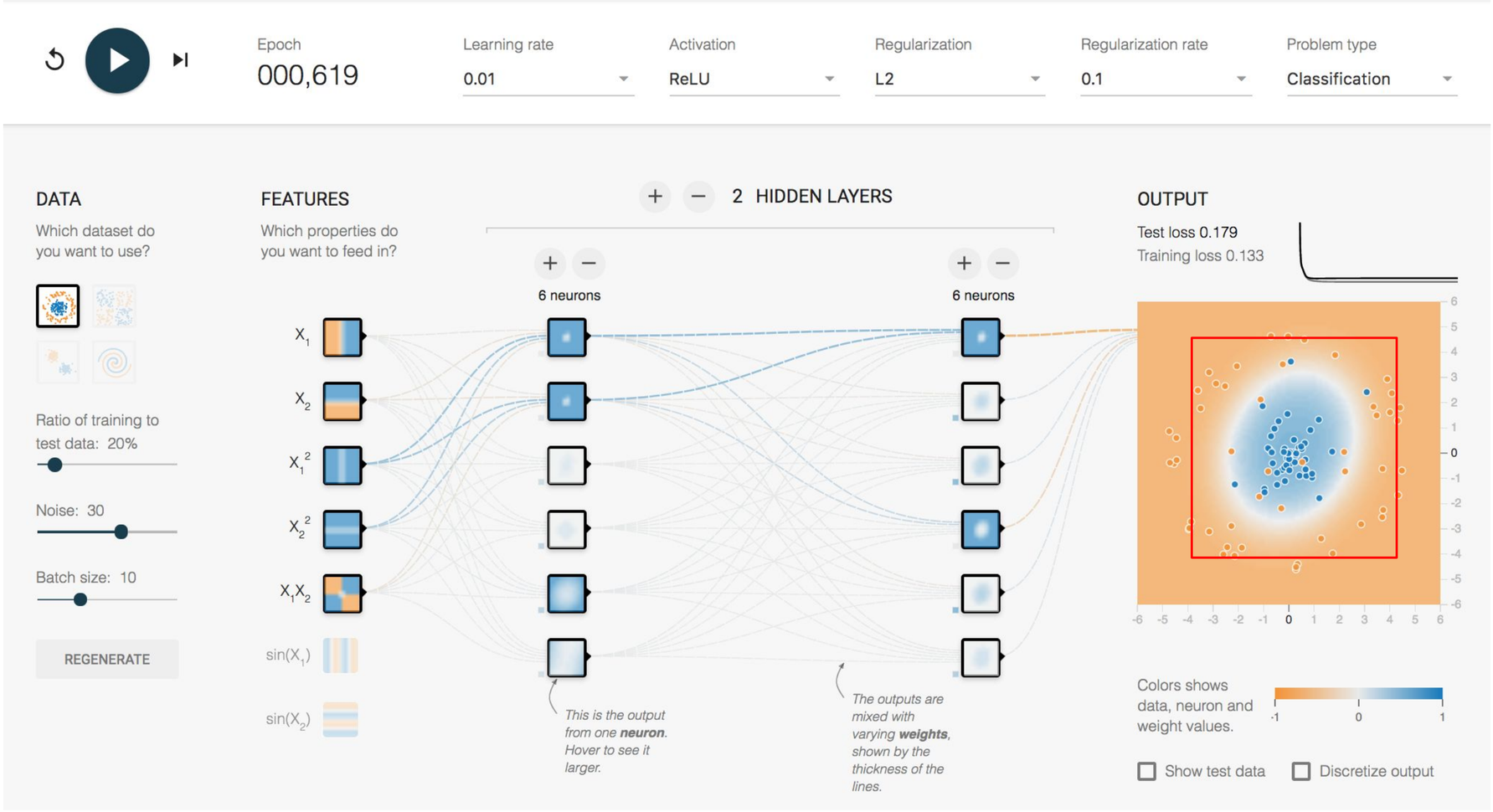
How can you define model complexity?



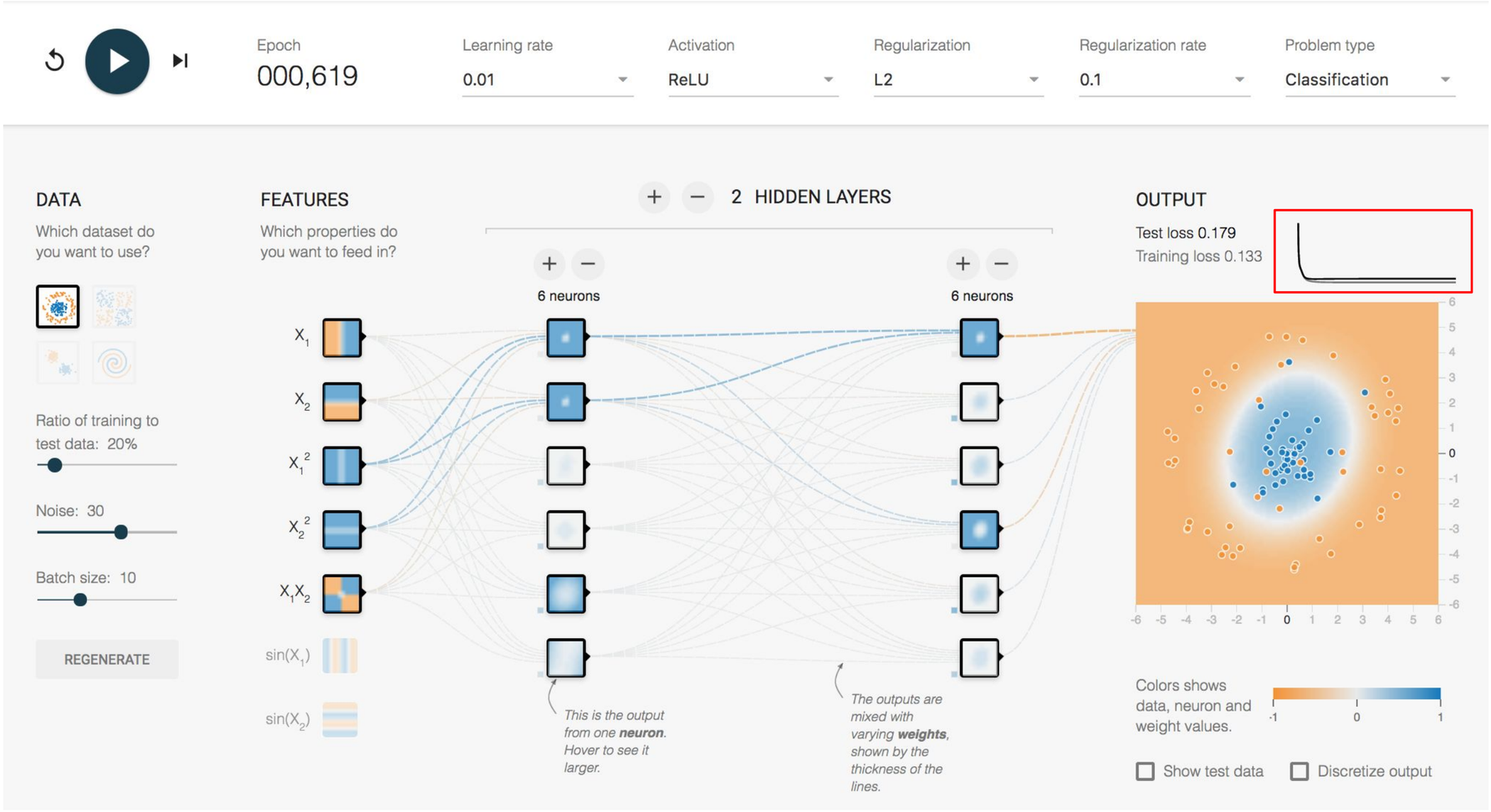
How can you define model complexity?



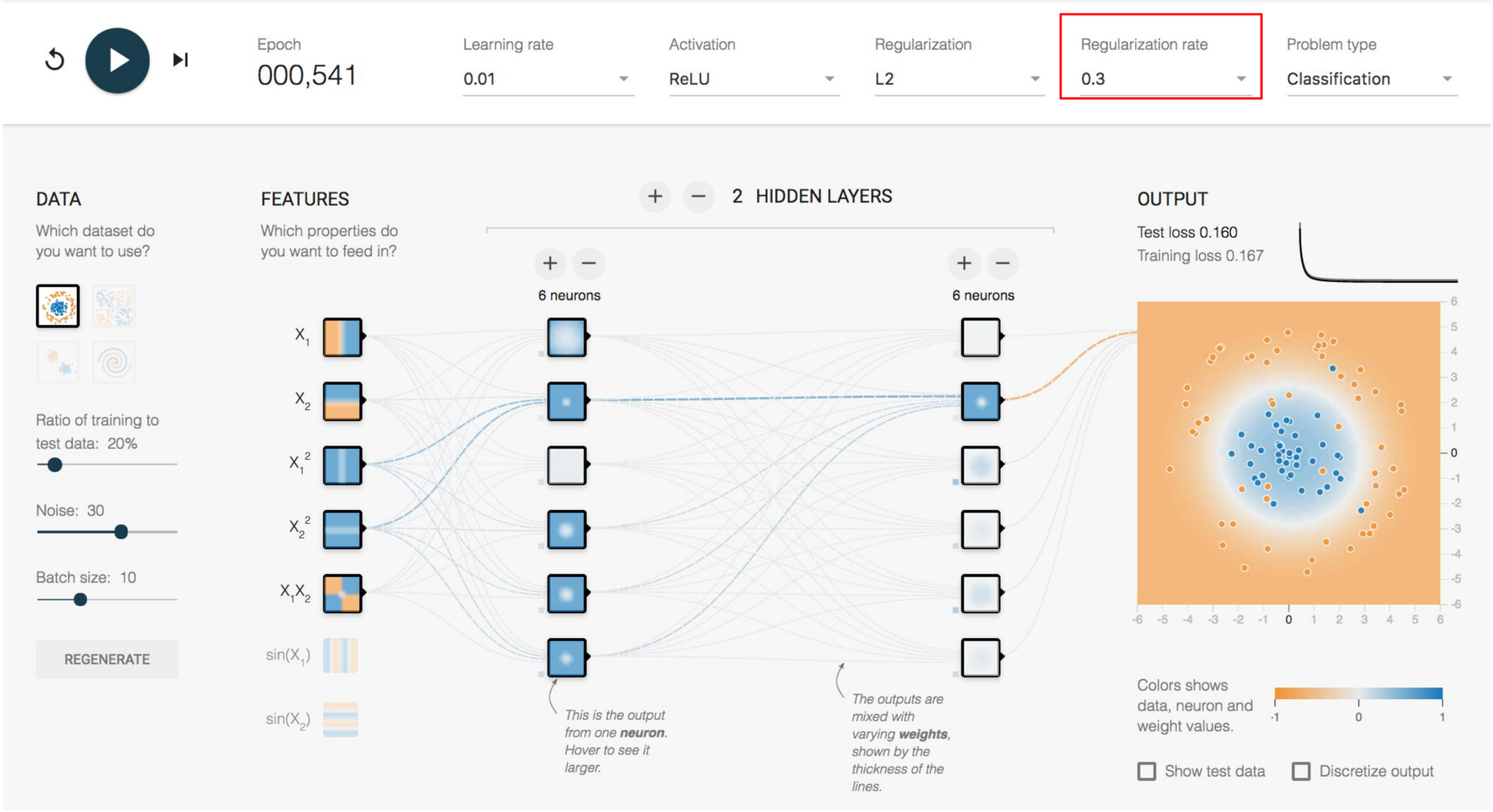
How can you define model complexity?



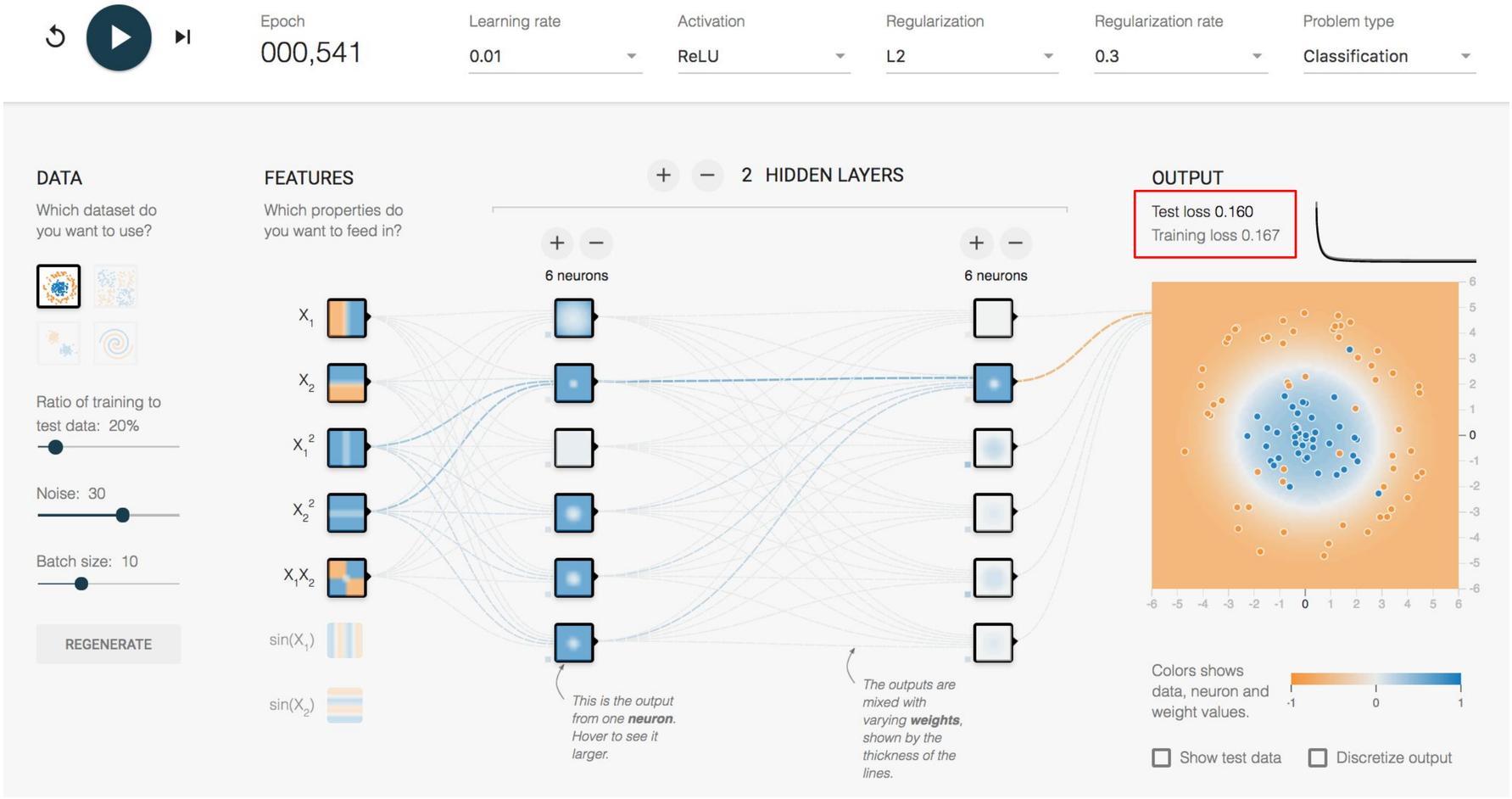
How can you define model complexity?



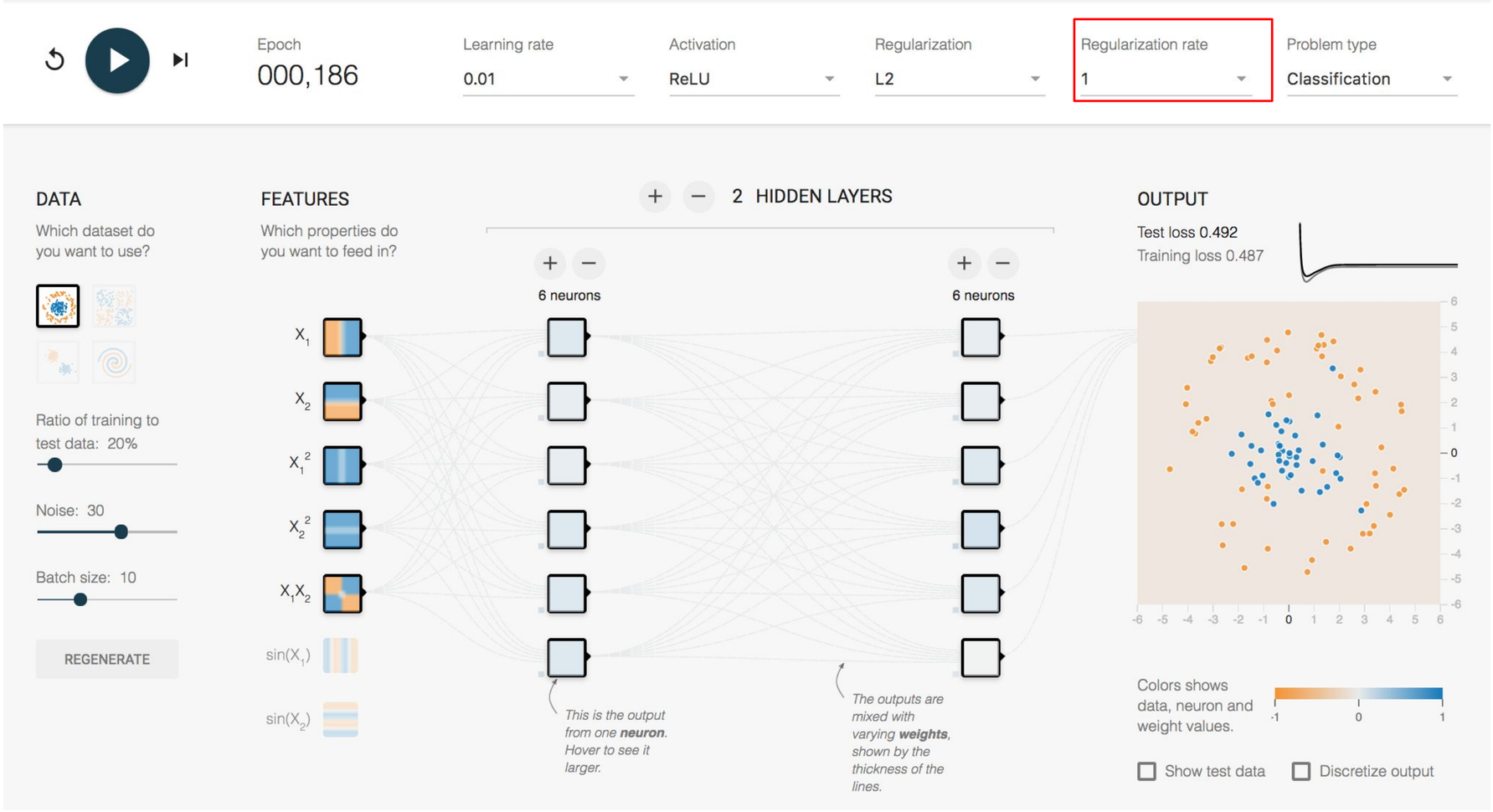
How can you define model complexity?



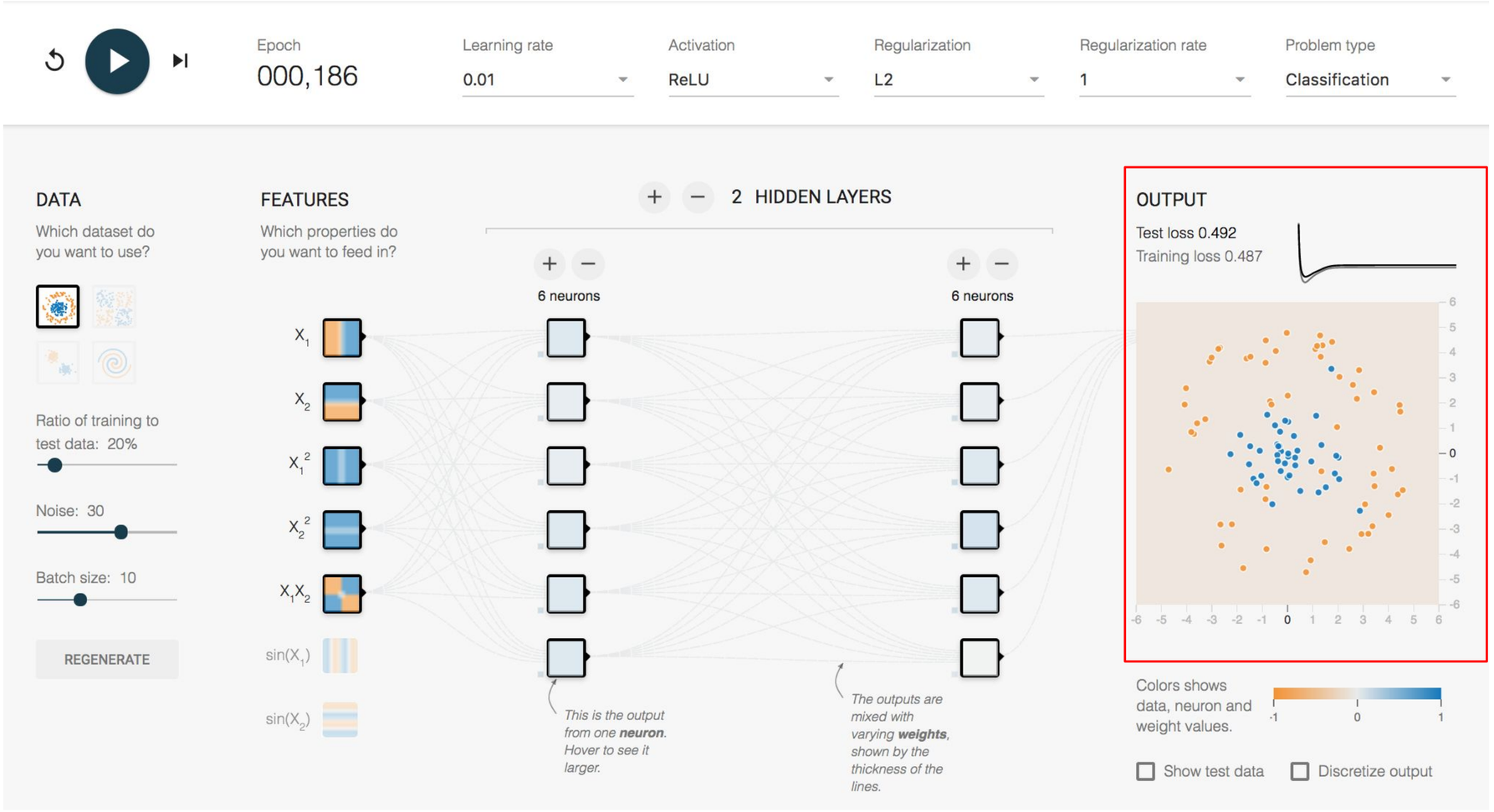
How can you define model complexity?



How can you define model complexity?



How can you define model complexity?

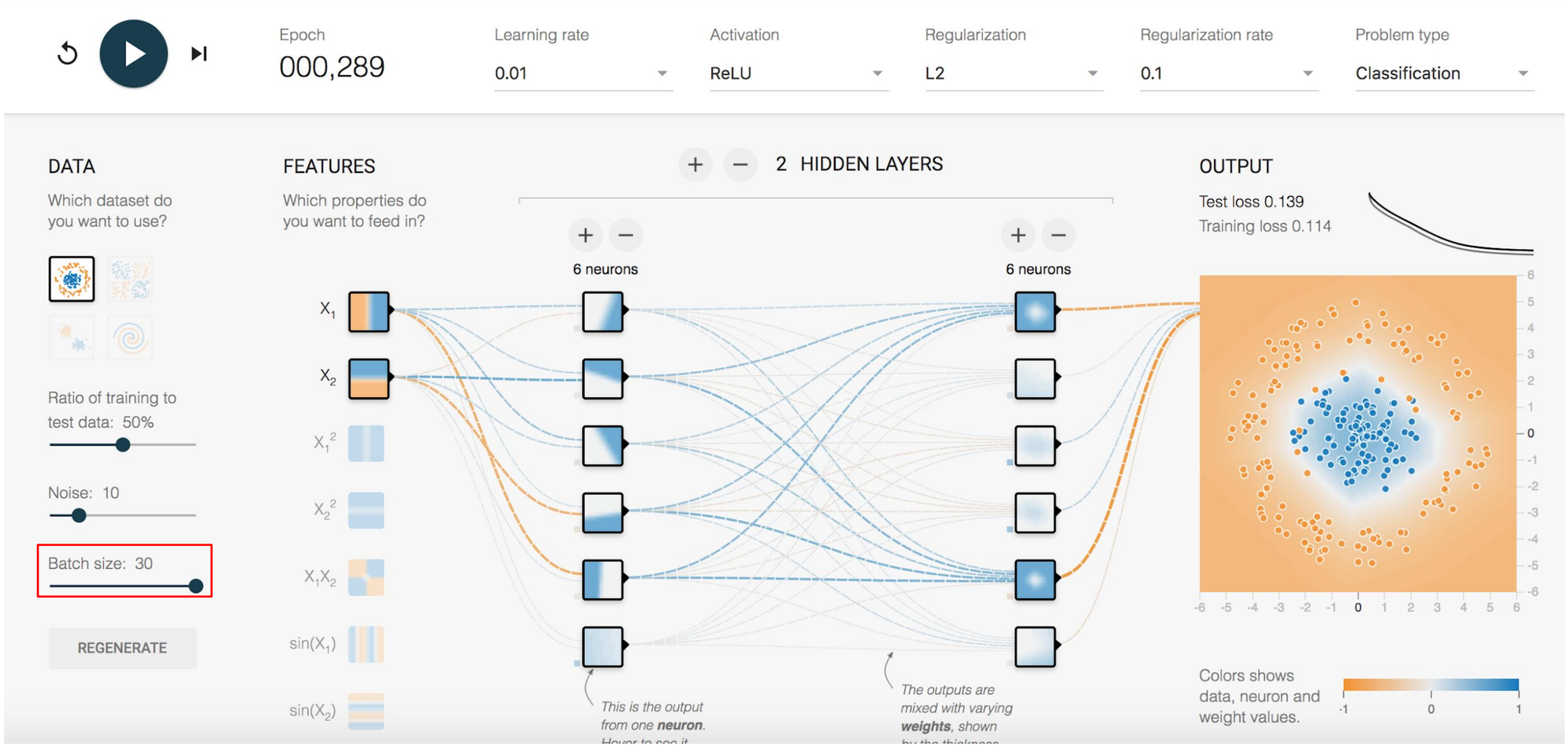


Google Cloud

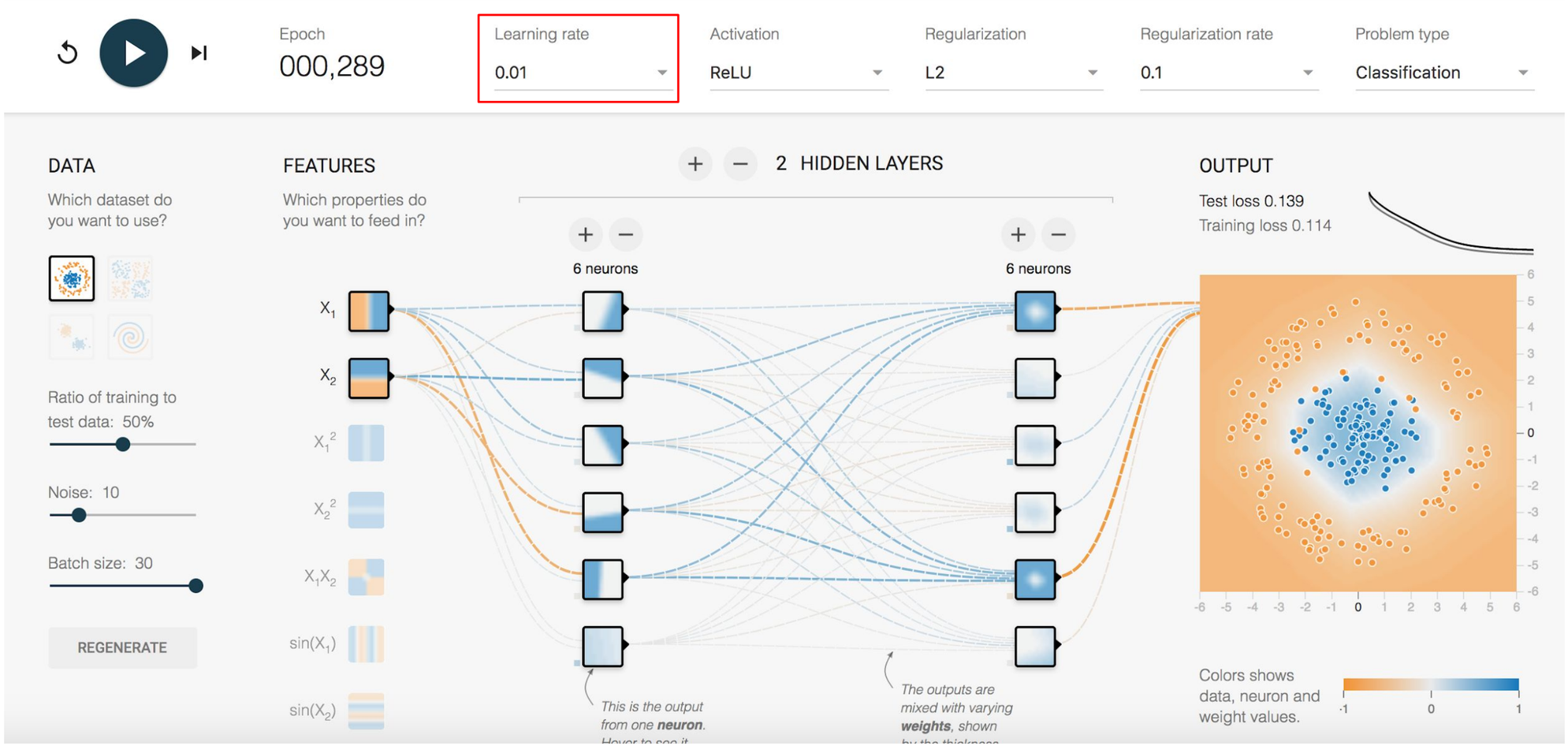
Learning Rate and Batch Size

Fereshteh Mahvar

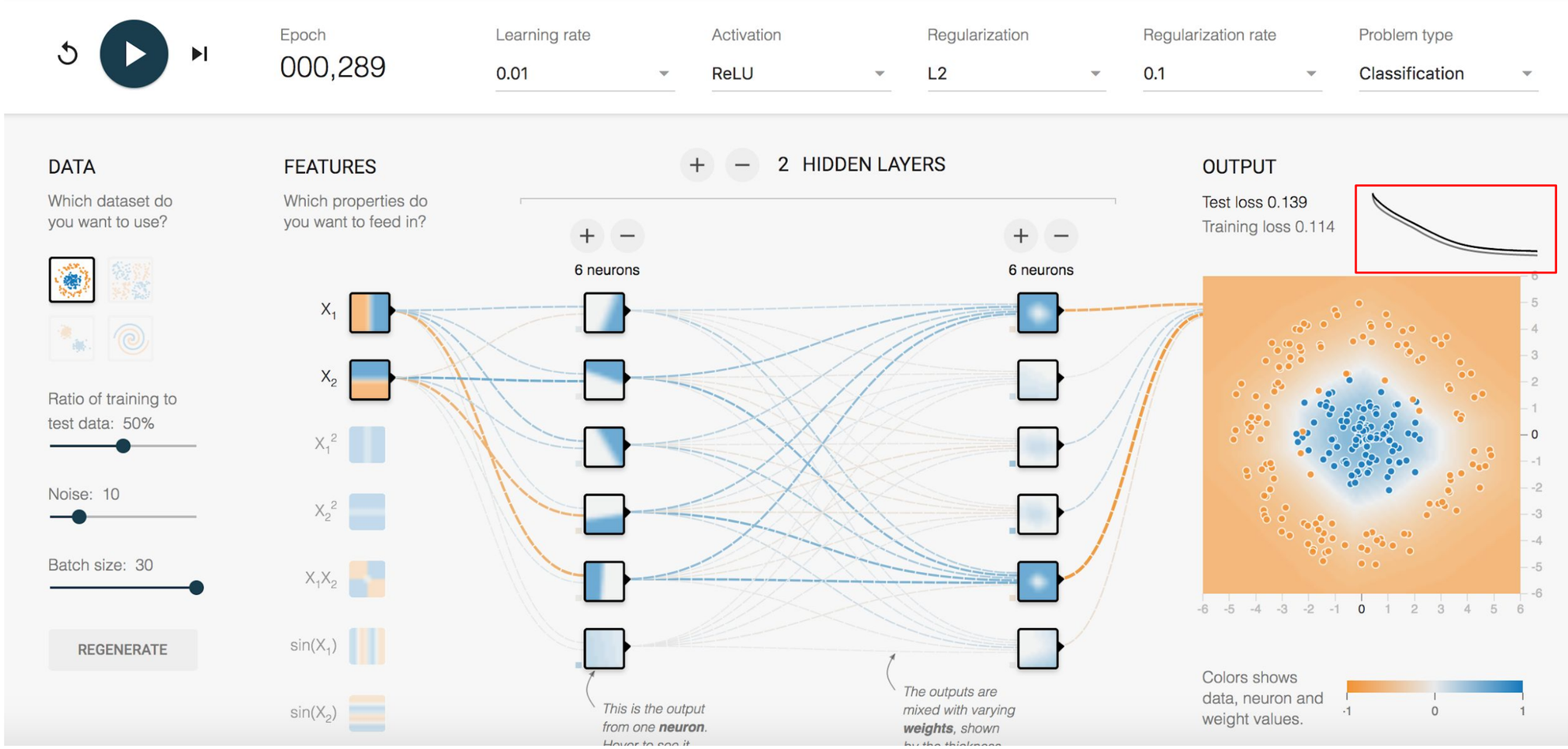
How can you define model complexity?



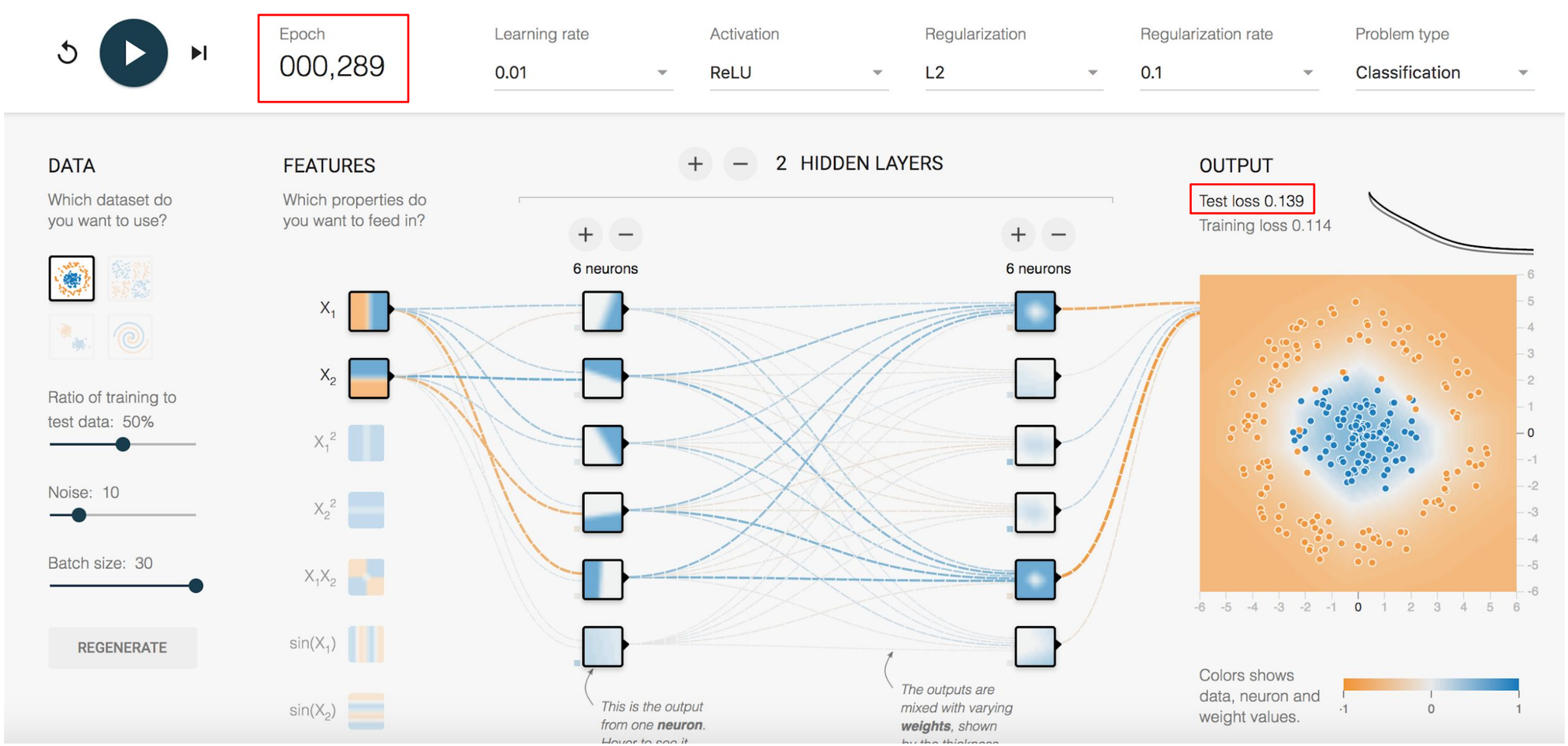
How can you define model complexity?



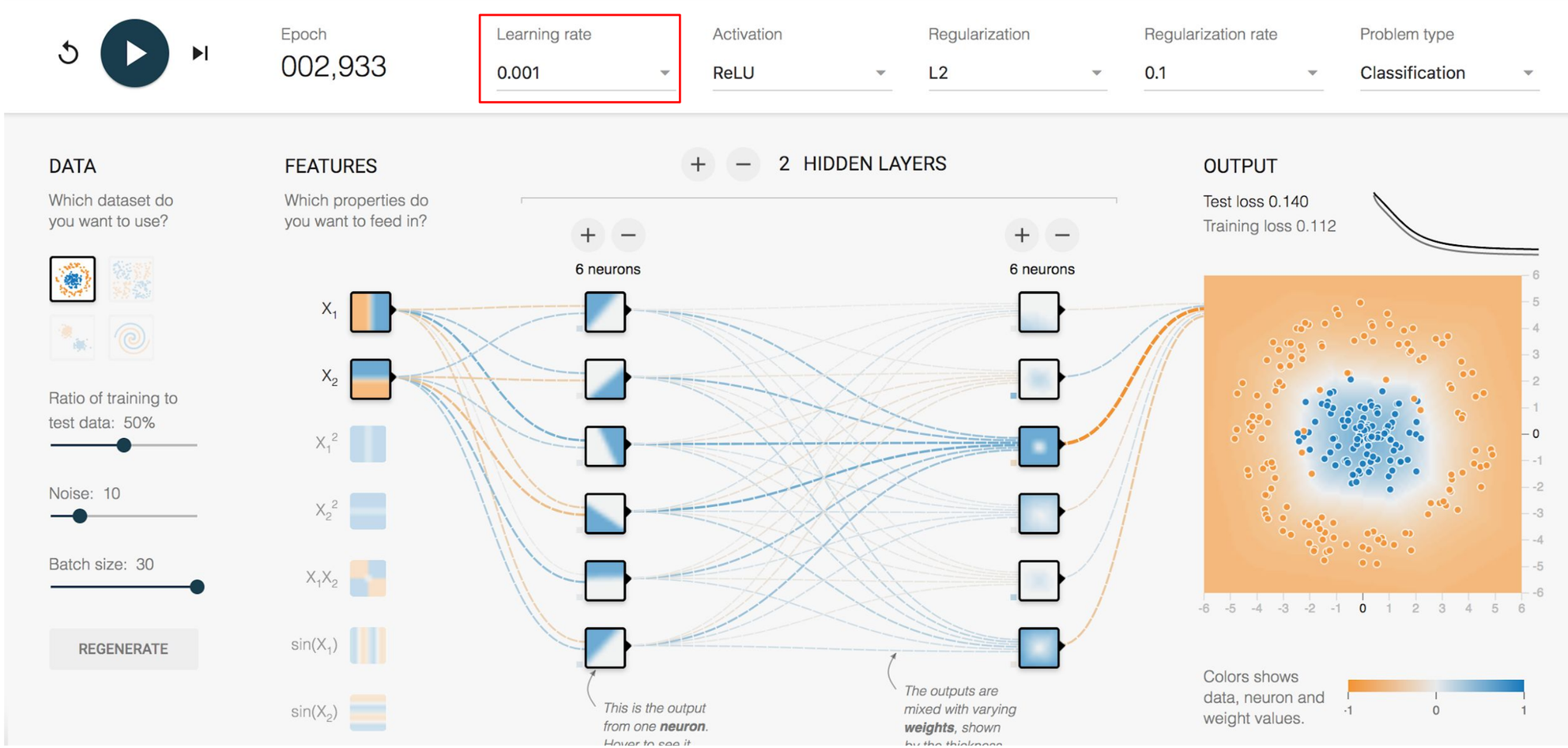
How can you define model complexity?



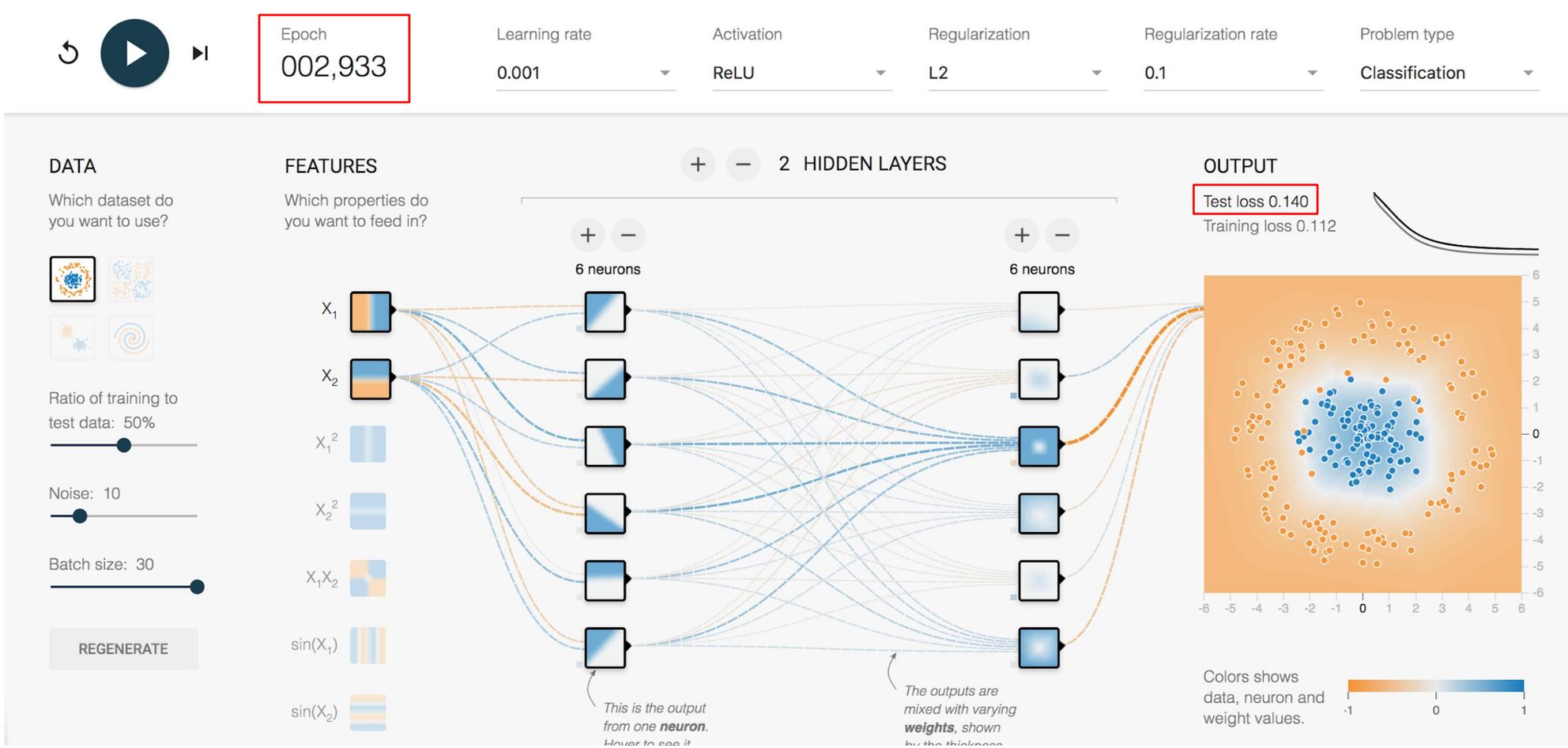
How can you define model complexity?



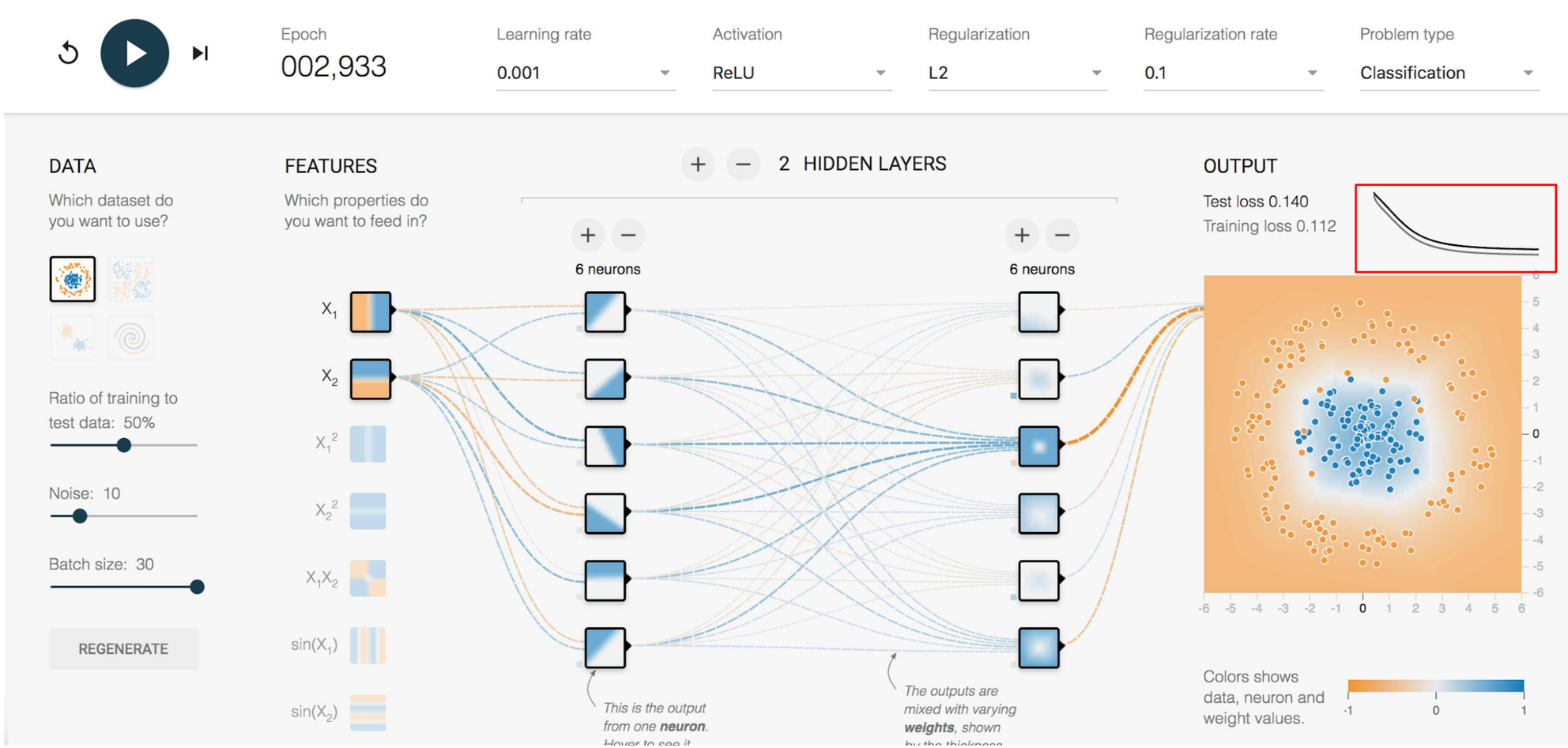
How can you define model complexity?



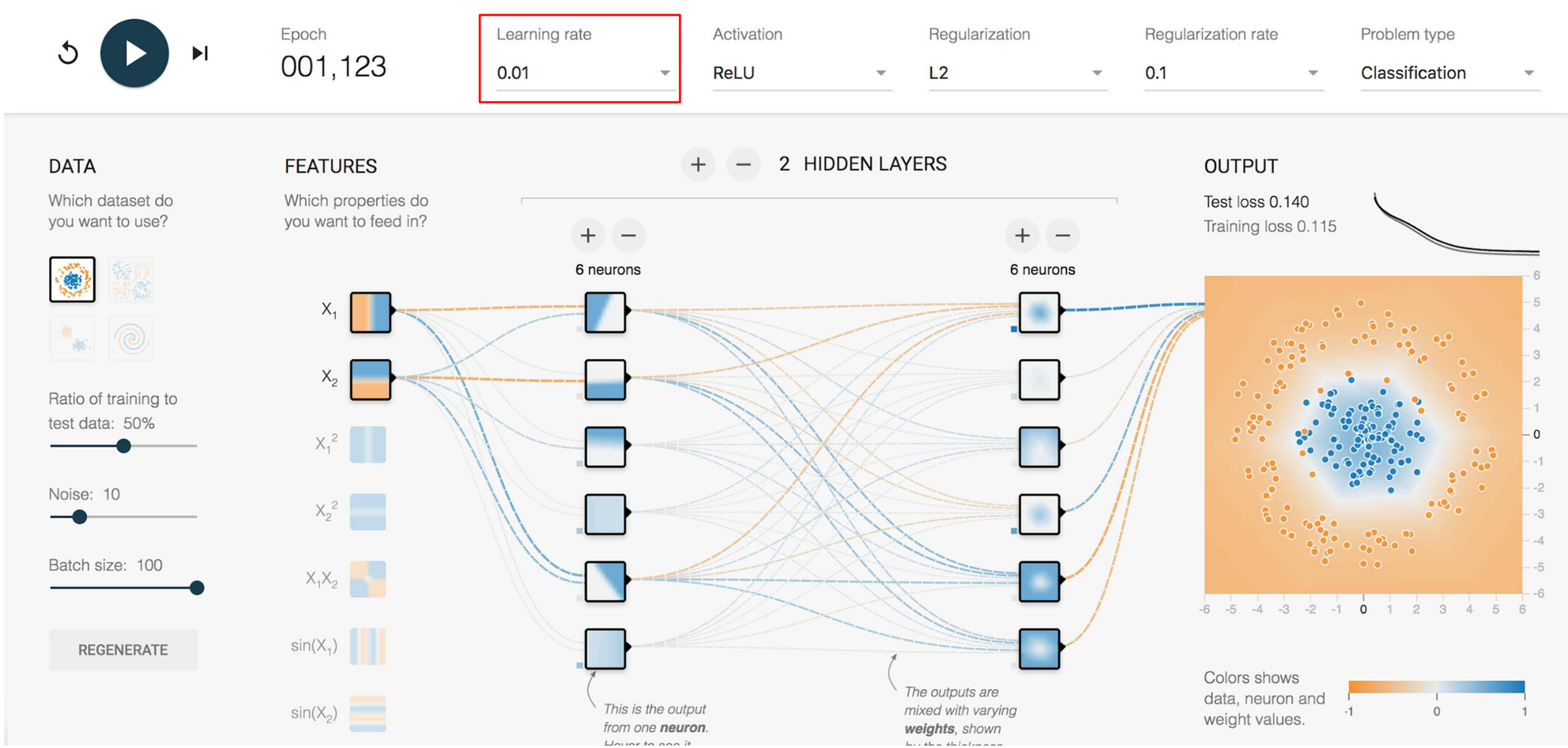
How can you define model complexity?



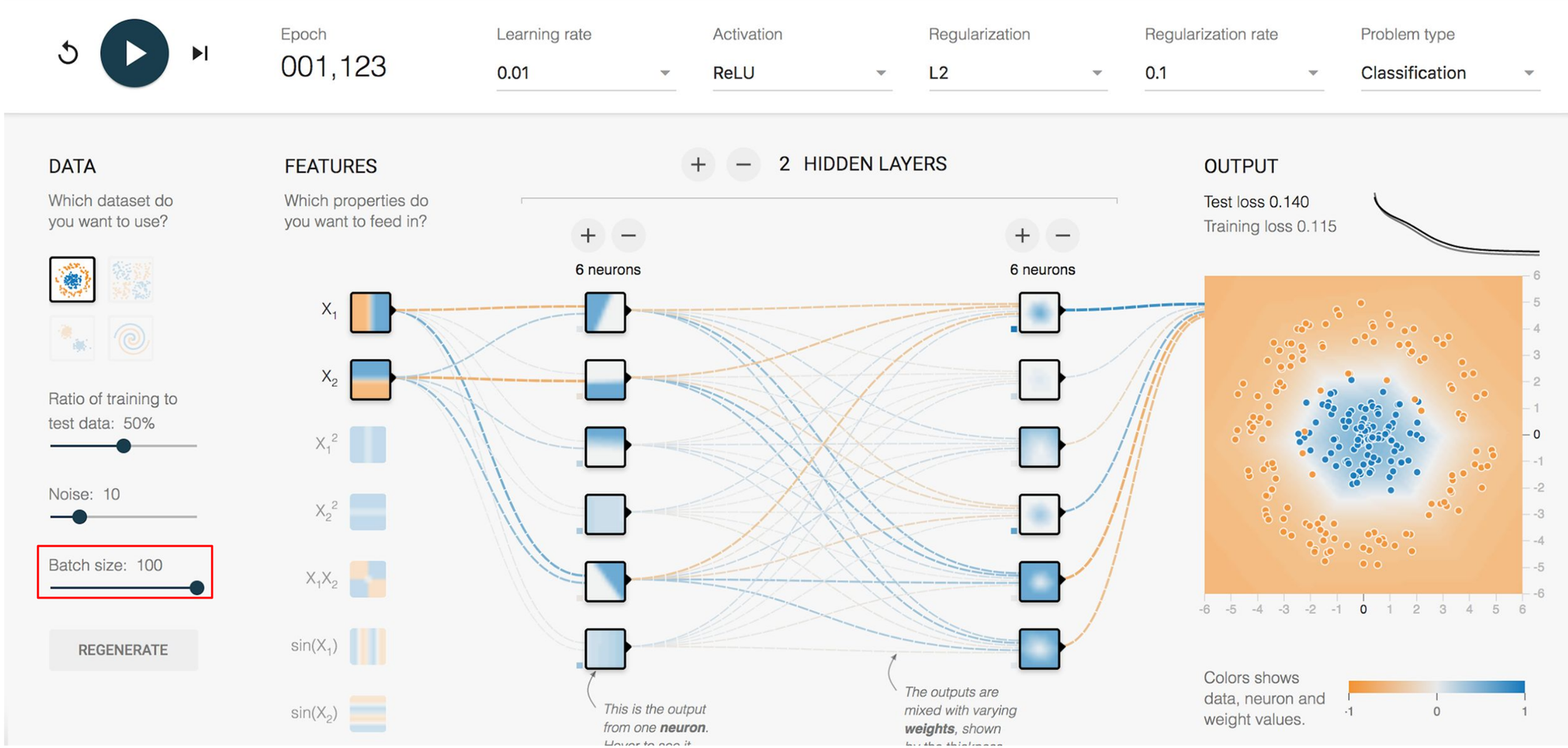
How can you define model complexity?



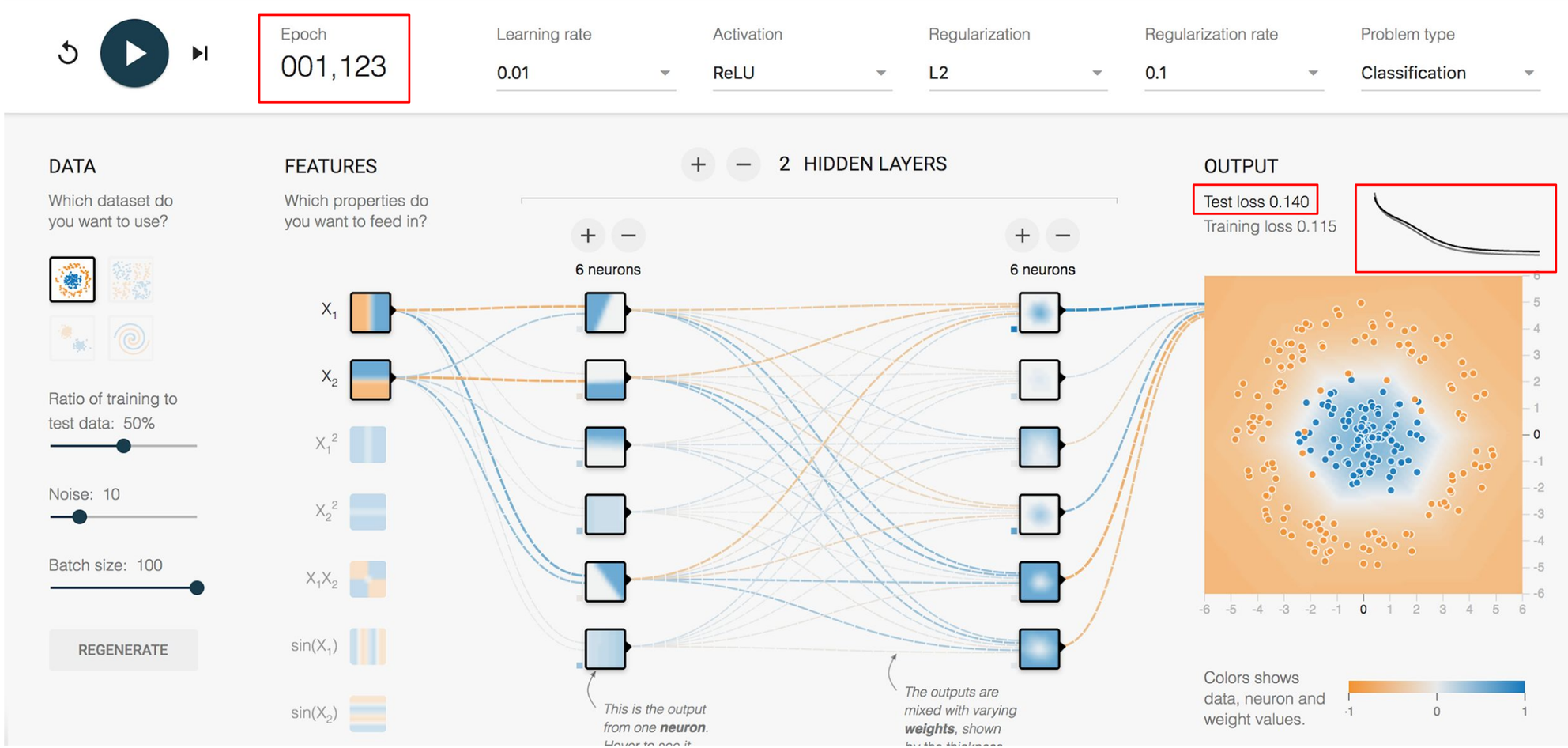
How can you define model complexity?



How can you define model complexity?



How can you define model complexity?



How can you define model complexity?

We have several knobs
that are dataset-dependent

Learning rate controls the size of the step in weight space

If too small, training will take a long time

If too large, training will bounce around

Default learning rate in Estimator's LinearRegressor is smaller of 0.2 or 1/sqrt(num_features) -- this assumes that your feature and label values are small numbers

The batch size controls the number of samples that gradient is calculated on.

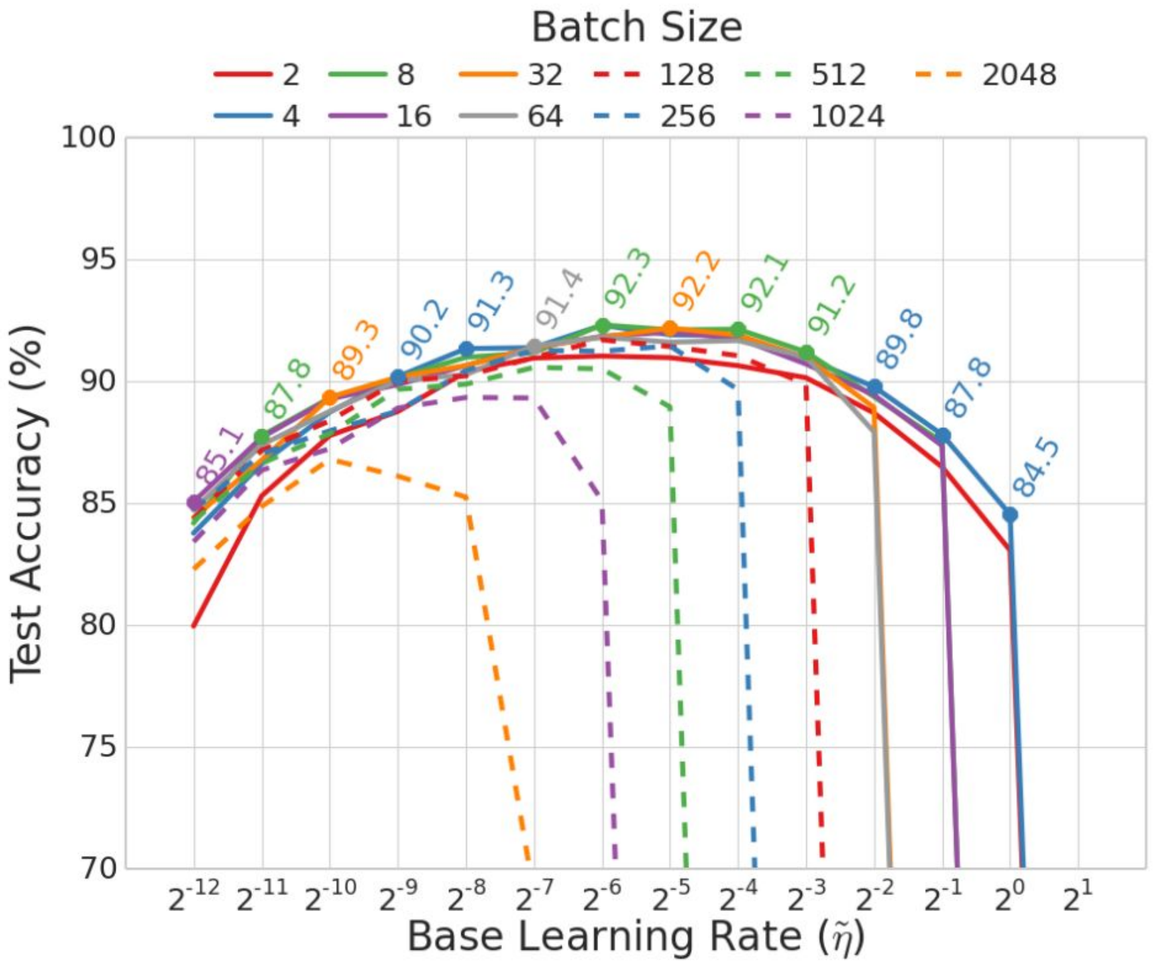
If too small, training will bounce around

If too large, training will take a very long time

40-100 tends to be a good range for batch size

Can go up to as high as 500

Larger batch sizes require smaller learning rates



Revisiting Small Batch Training for Deep Neural Networks, Masters and Luschi, 2018

Regularization provides a way to define model complexity based on the values of the weights

horse

horse racing

horseback riding

horseradish

horse trading

The examples are ordered alphabetically; this ordering results in high correlation between successive examples.

Google Cloud

Optimization

Fereshteh Mahvar

Optimization is a major field of ML research

GradientDescent

Momentum

AdaGrad

AdaDelta

Adam

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum

AdaGrad

AdaDelta

Adam

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad

AdaDelta

Adam

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad -- Give frequently occurring features low learning rates

AdaDelta

Adam

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad -- Give frequently occurring features low learning rates

AdaDelta -- Improves AdaGrad by avoiding reducing LR to zero

Adam

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad -- Give frequently occurring features low learning rates

AdaDelta -- Improves AdaGrad by avoiding reducing LR to zero

Adam -- AdaGrad with a bunch of fixes

Ftrl

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad -- Give frequently occurring features low learning rates

AdaDelta -- Improves AdaGrad by avoiding reducing LR to zero

Adam -- AdaGrad with a bunch of fixes

Ftrl -- “Follow the regularized leader”, works well on wide models

. . .

Optimization is a major field of ML research

GradientDescent -- The traditional approach, typically implemented stochastically i.e. with batches

Momentum -- Reduces learning rate when gradient values are small

AdaGrad -- Give frequently occurring features low learning rates

AdaDelta -- Improves AdaGrad by avoiding reducing LR to zero

Adam -- AdaGrad with a bunch of fixes

Ftrl -- “Follow the regularized leader”, works well on wide models

. . .

Good defaults for DNN and Linear models

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

How to change optimizer, learning rate, batchsize

```
train_fn = tf.estimator.inputs.pandas_input_fn(..., batch_size=10)
myopt = tf.train.FtrlOptimizer(learning_rate=0.01,
                               l2_regularization_strength=0.1)
model = tf.estimator.LinearRegressor(..., optimizer=myopt)
model.train(input_fn=train_fn, steps=10000)
```

1. Control batch size via the input function
2. Control learning rate via the optimizer passed into model
3. Set up regularization in the optimizer
4. Adjust number of steps based on batch_size, learning_rate
5. Set number of steps, not number of epochs because distributed training doesn't play nicely with epochs.

Lab

Improve model performance by hand tuning parameters

Fereshteh Mahvar