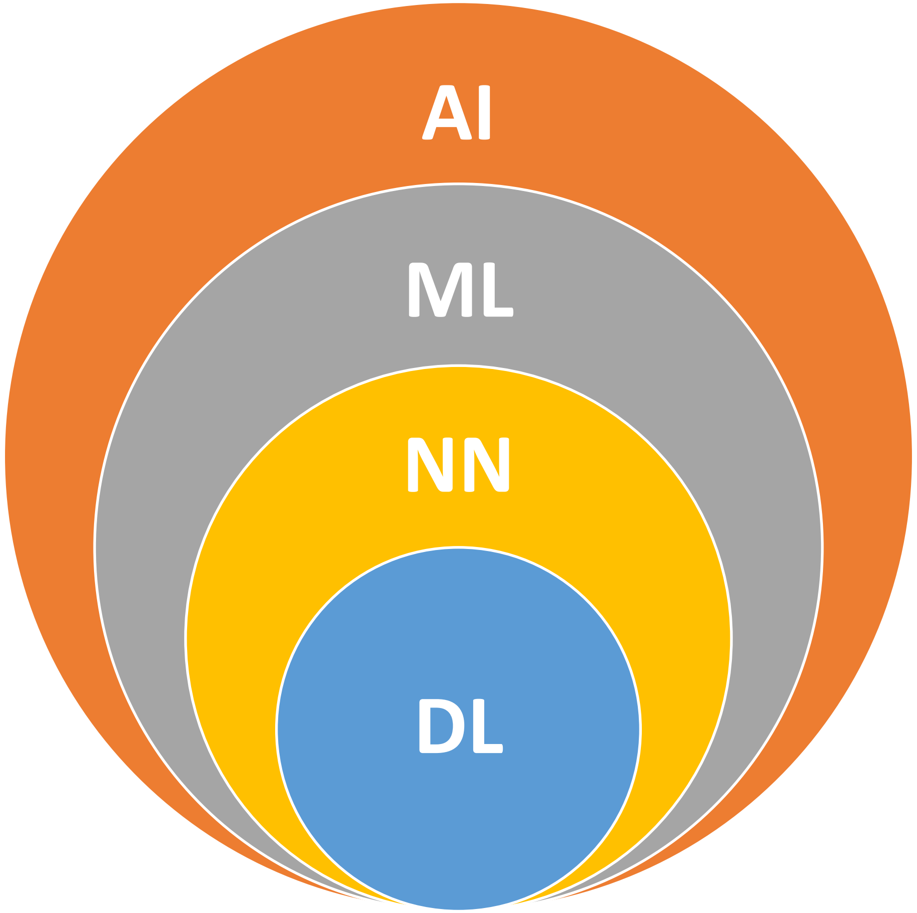
NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Foundational Concepts of Artificial Intelligence-Based Modeling

Peter Bajcsy, Ph.D.
Information Technology Laboratory
National Institute of Standards and Technology

Terminology



- AI: Artificial intelligence is the overarching system.
- ML: Machine learning is a subset of AI.
- NN: Neural networks are the backbone of deep learning algorithms.
- DL: Neural network of more than three layers, including the inputs and the output.

URL

1. Why are we interested in Neural Networks?

1. Why are Scientists interested in Neural Networks?

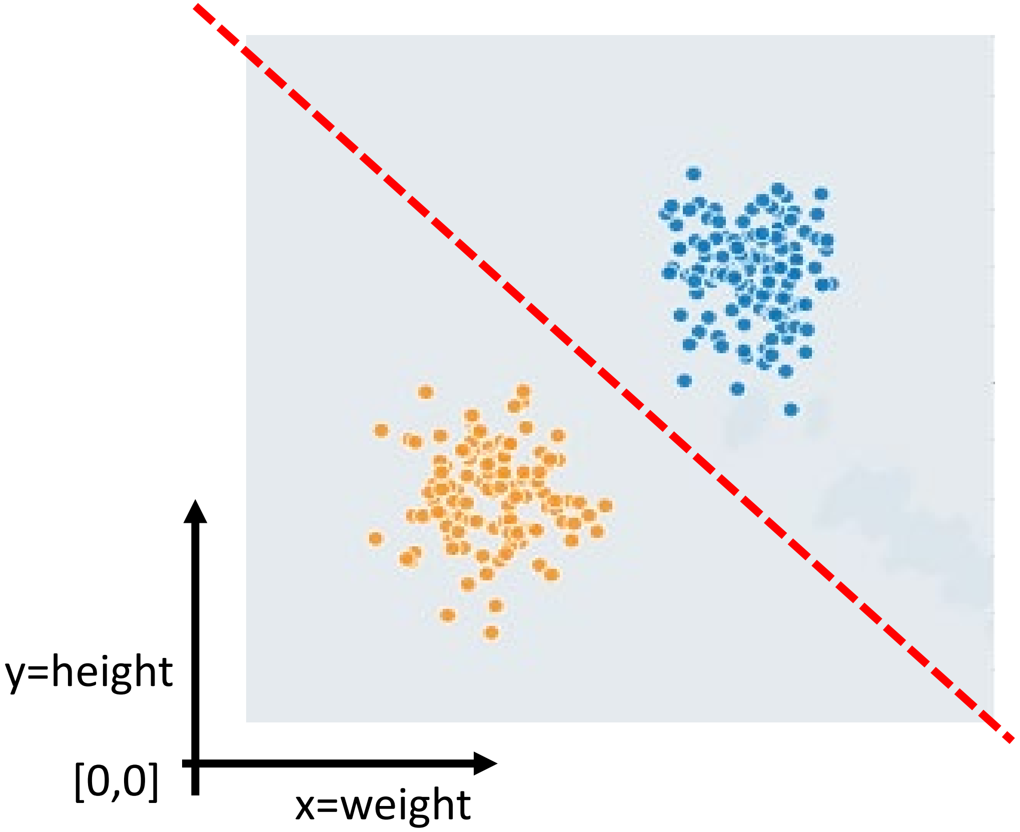
Relevance of AI to Metrology:

- Neural Networks can classify pictures into classes →
  - Tasks: classification, object detection, segmentation, tracking, recognition
- Neural Networks can assist humans →
  - Learn from training data: labeled and partially unlabeled.
  - Learn from past models: a priori knowledge and simulations.
- Neural Networks can outperform humans →
  - Task metrics: overall accuracy (NN do not get tired) beyond visual detection limits and speed.
- Neural Networks can save us a lot of time →
  - Automation of tasks: processing TB of images, detecting outliers, and leading to discovery.

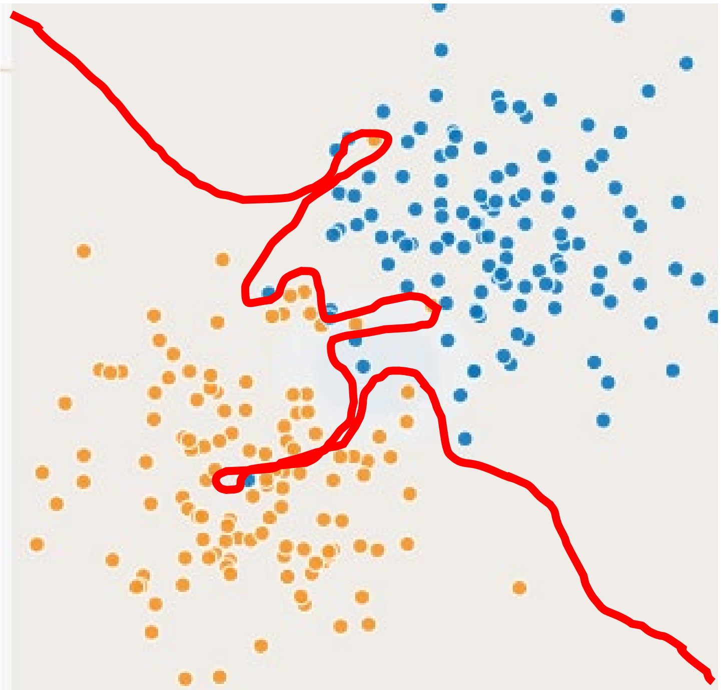
2. How to create a complex boundary separating two classes?

Types of Boundaries

Linear boundary

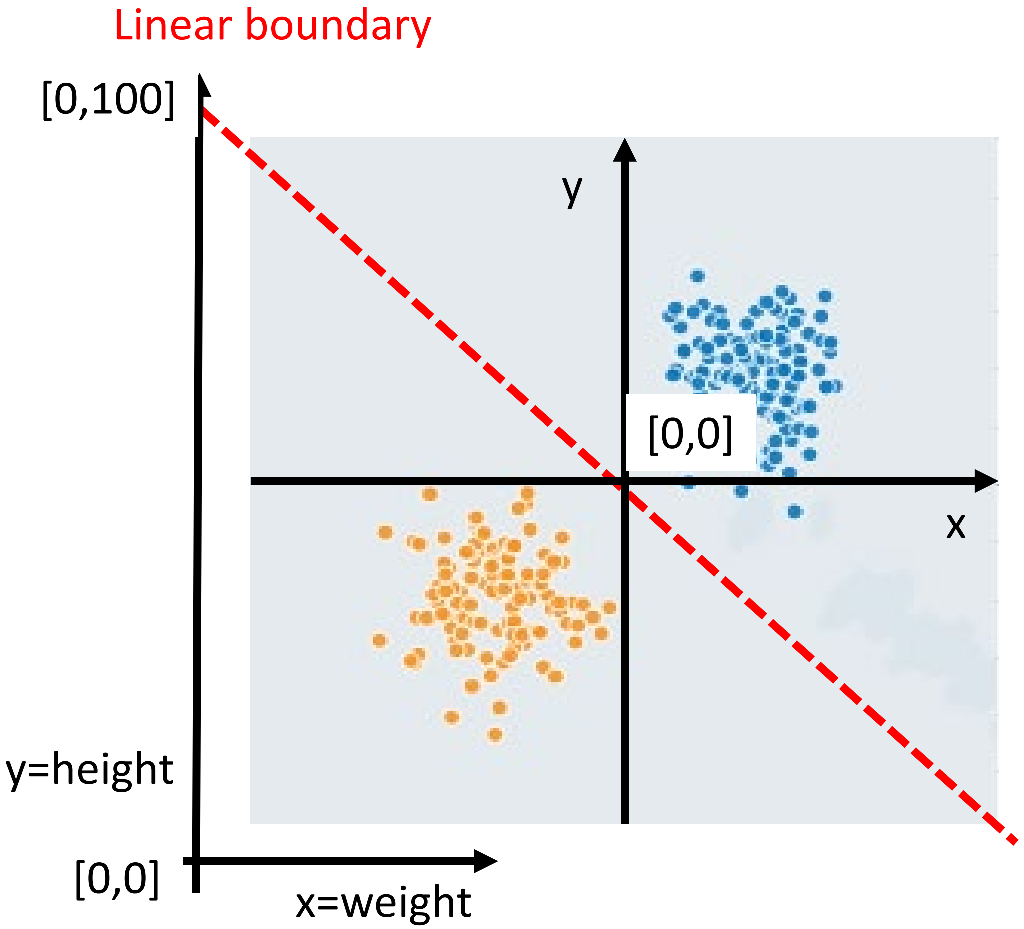


Non-linear boundary



How to describe and derive these boundaries automatically?

Linear Boundaries



$Boundary:\ x + y = 100$

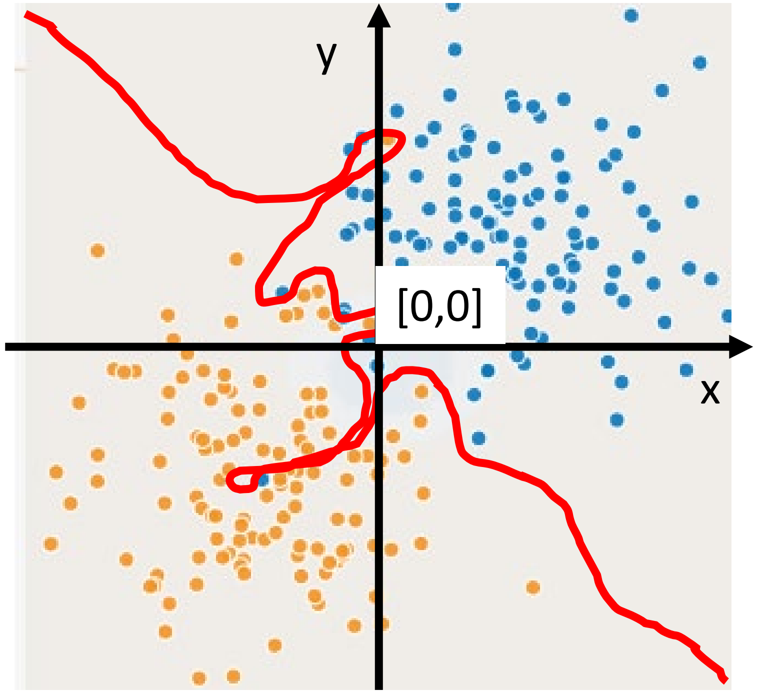
$Boundary:\ x + y = 0$

$Math\,model:\ if\ x + y\ < 0$
$then\ orange\ else\ blue$

How to describe and derive these boundaries automatically?

Nonlinear Boundaries

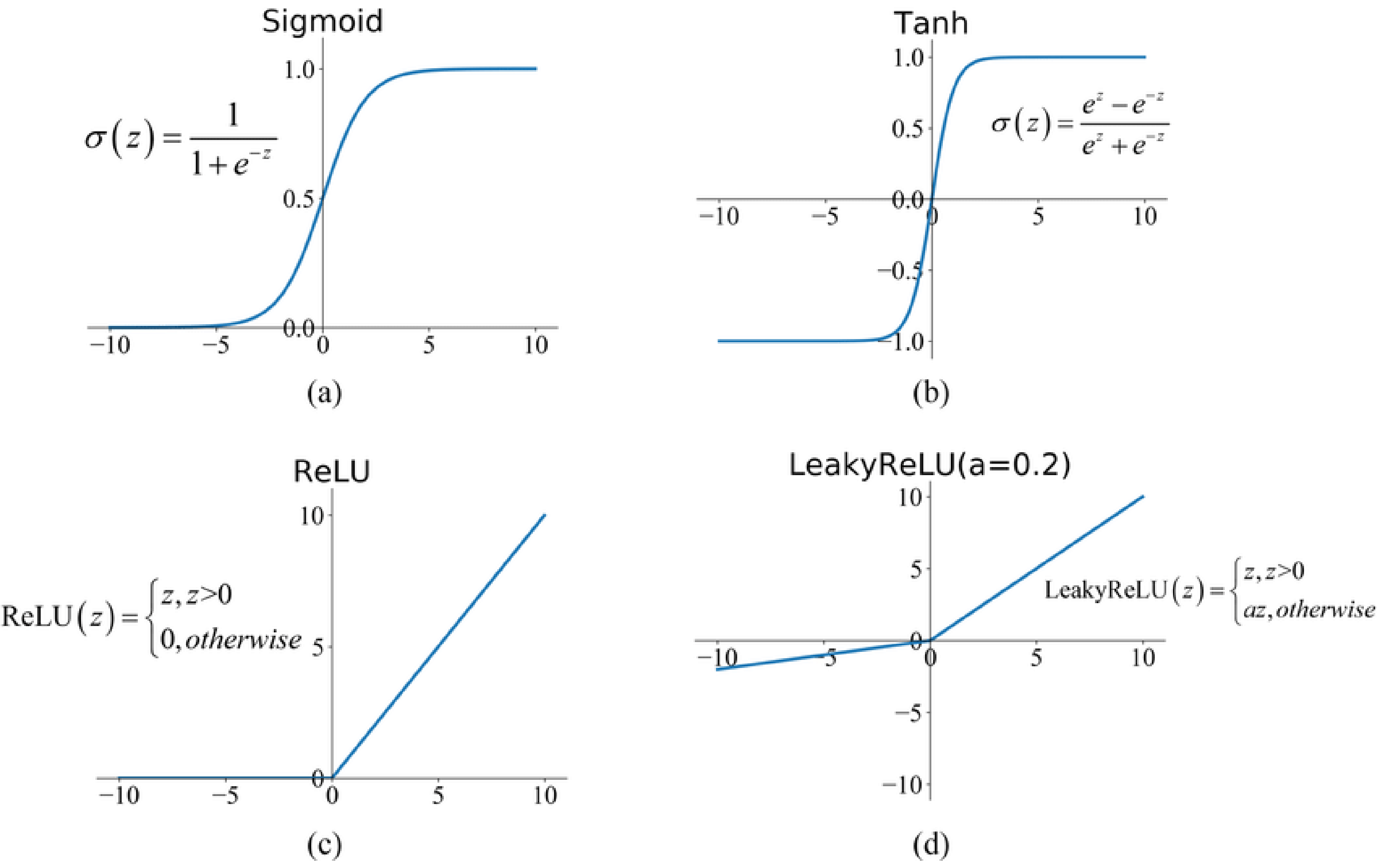
Non-linear boundary



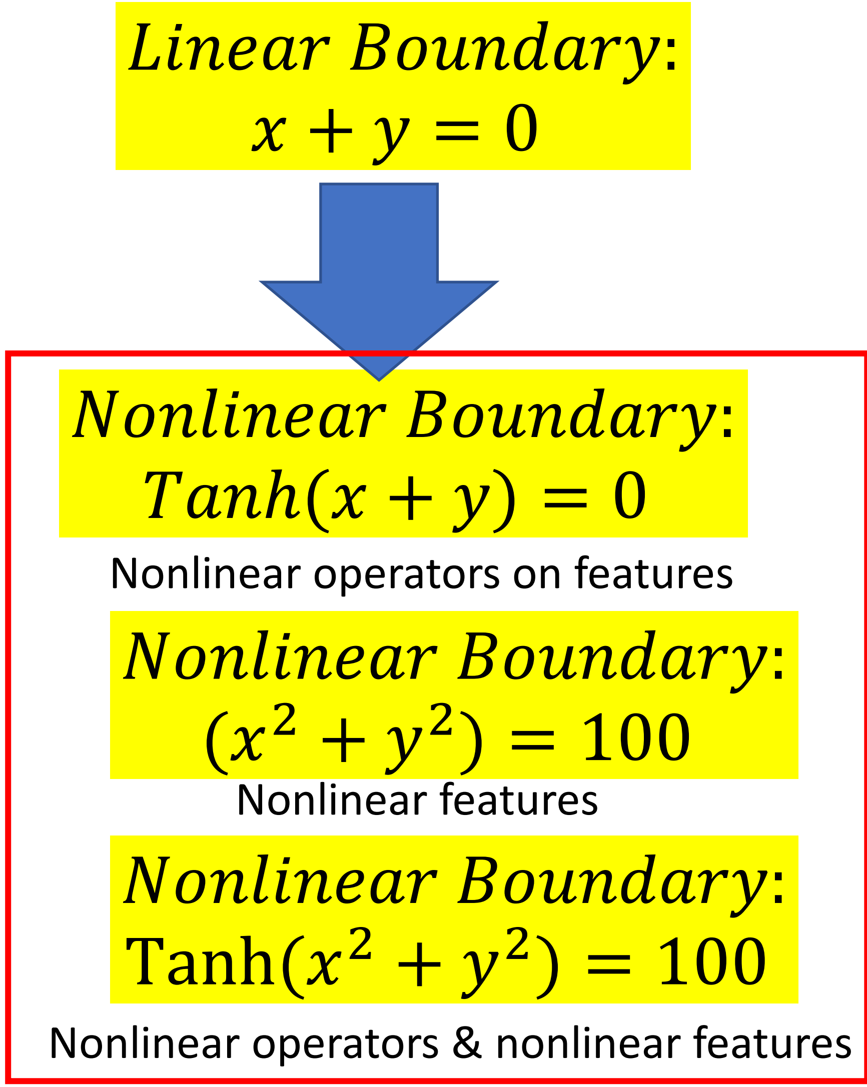
How to describe a nonlinear boundary using x and y?

How to derive the boundary description automatically?

Describe nonlinear boundary using x and y



Nonlinear operators: (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LReLU.



Linear Boundary:
$x + y = 0$

Nonlinear Boundary:
$Tanh(x + y) = 0$

Nonlinear operators on features

Nonlinear Boundary:
$(x^2 + y^2) = 100$

Nonlinear features

Nonlinear Boundary:
$\mathrm{Tanh}(x^2 + y^2) = 100$

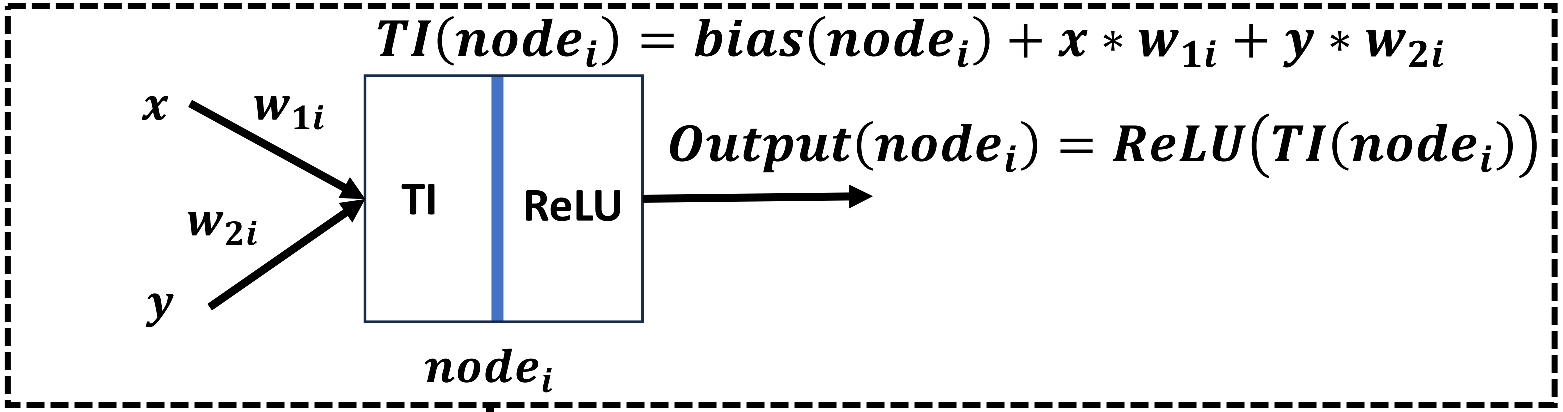
Nonlinear operators & nonlinear features

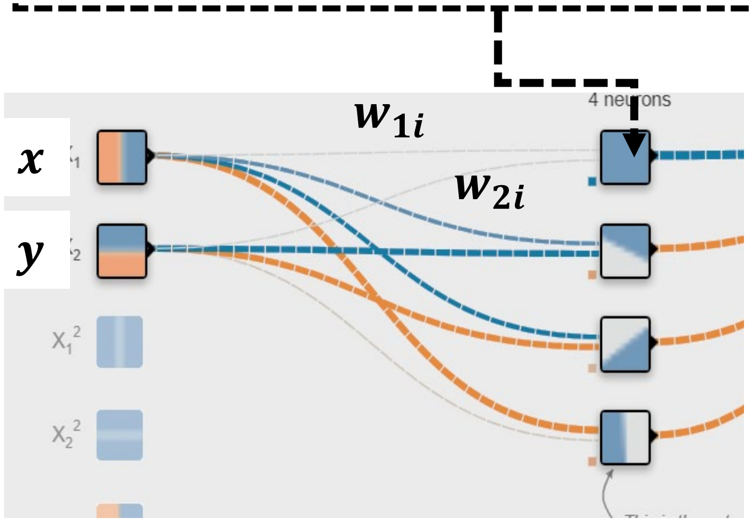
Prepare NN model for non-linear boundary

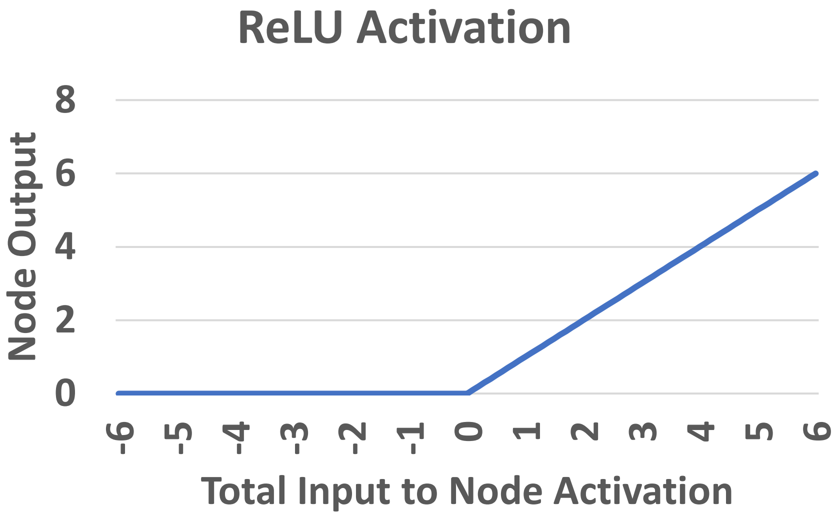
Mathematical Model

$$TI(node_i) = bias(node_i) + x * w_{1i} + y * w_{2i}$$

$$Output(node_i) = ReLU\big(TI(node_i)\big)$$



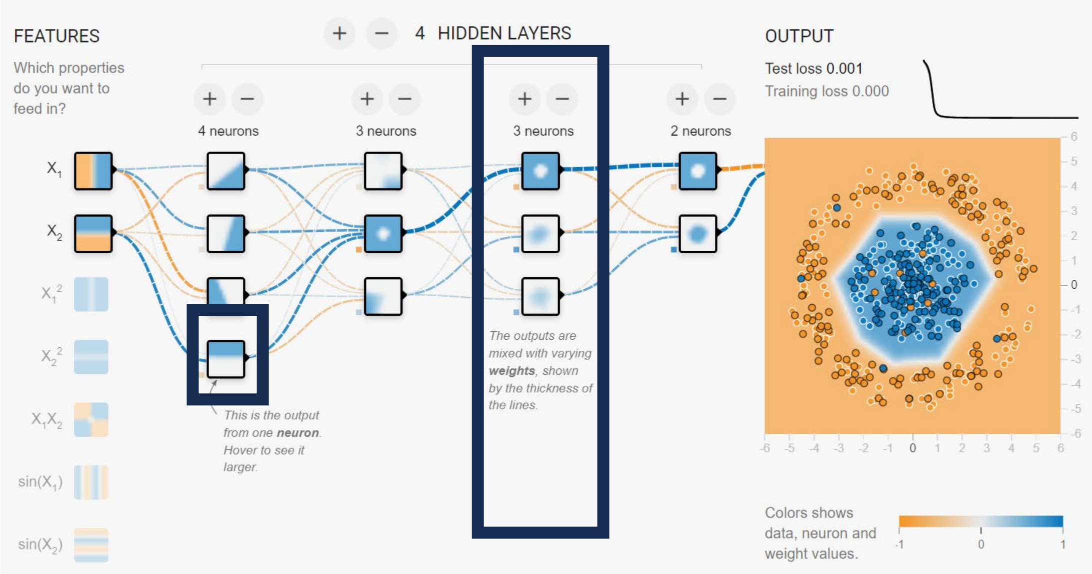




Derive boundary description automatically

1. Step - Configure Neural Network architecture (layers, nodes, connectivity)
2. Step – Prepare (input, output) pairs (images and labels)
3. Step – Initialize all weights in NN with random numbers
4. Step – Train weights so that the multiplications, additions, and nonlinear operators convert input to output (images of cats/dogs to labels of cats/dogs)
5. Step – Evaluate the accuracy of trained NN on test (input, output) pairs

Input:
(x, y) coordinates



Which properties do you want to feed in?

Test loss 0.001
Training loss 0.000

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

Output:
Label of (x, y) coordinates = {Orange, Blue}

2. How Are Scientists Creating Complex Boundary?

Relevance of AI to Metrology:

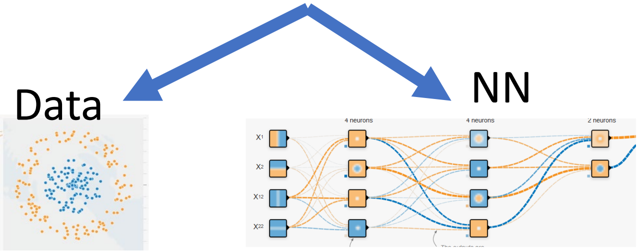
- Describe boundaries mathematically
  - → Boundaries are unknown in discoveries!
- Linear boundaries can be described directly with x and y
  - → Linear regression is rarely accurate for complex biomedical phenomena!
- Nonlinear boundaries must be described with nonlinear transformations (nonlinear features and operators/activations)
  - → ReLU is the most frequently used in practice. The choice might depend on the convergence of the model training.
- Nonlinear boundary description can be derived automatically by training Neural Networks (NN) with training data.
  - → TBD next

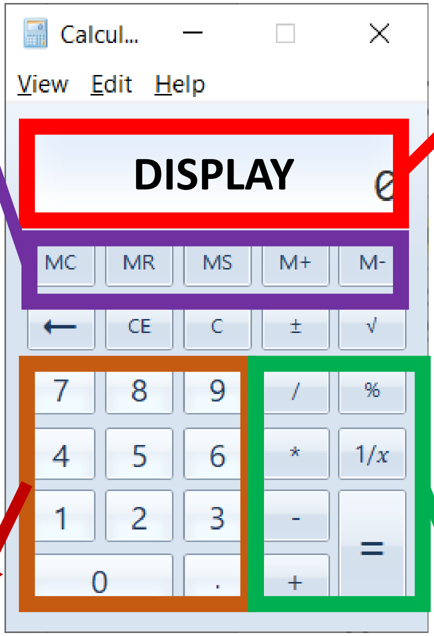
3. How do you learn primitive Neural network concepts interactively?

- https://pages.nist.gov/nn-calculator/
- Four simple exercises
    1. Simple to complex NN architecture
    2. Linear to non-linear features
    3. Linear to non-linear activation functions
    4. Well-separated to inseparable (interleaved) classes

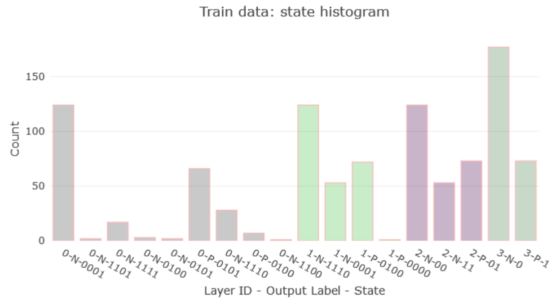
Learning about NNs Using a Web-Based “NN Calculator”

Memory Operations: MC, MR, M+, M-, MS, AVG

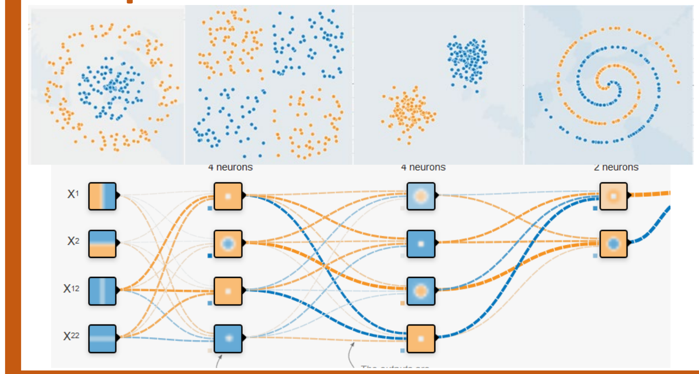




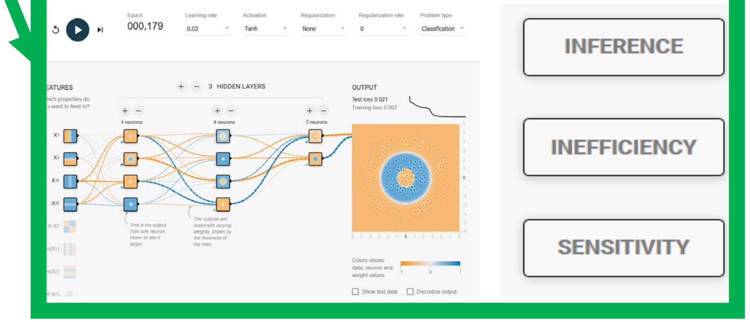
Display: NN, Data, Results



Operands: 2D data and NN



NN Operators: design, parametrize, train, infer, inefficiency, sensitivity



Epoch 000,179

Learning rate 0.03

Activation Tanh

Regularization None

Regularization rate 0

Problem type Classification

3 HIDDEN LAYERS

OUTPUT

Test loss 0.021

Training loss 0.007

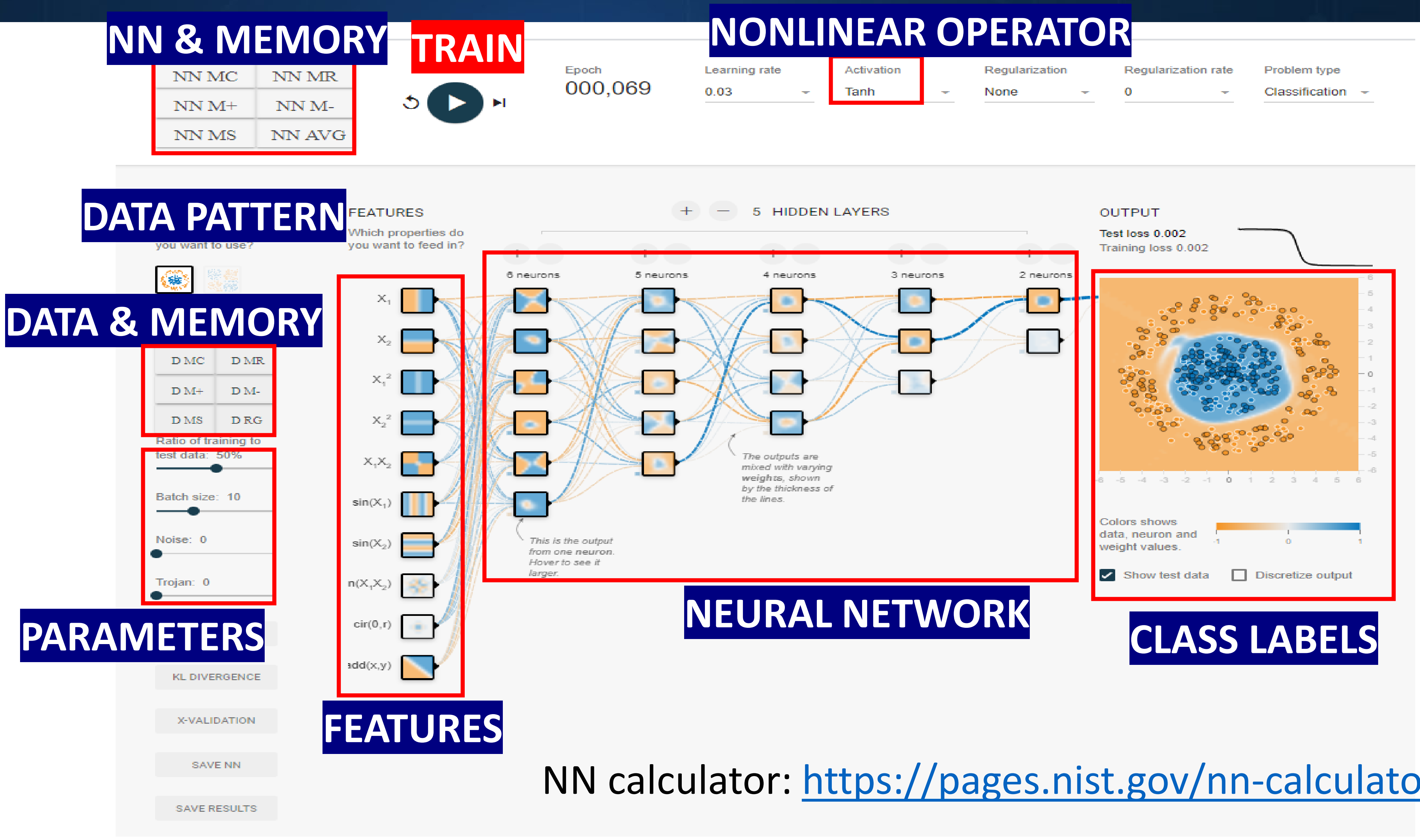
This is the output from one neuron. Hover to see it larger.

The outputs are mixed with varying weights, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

NN Calculator Layout



Epoch 000,069

Learning rate 0.03

Activation Tanh

Regularization None

Regularization rate 0

Problem type Classification

you want to use?

FEATURES

Which properties do you want to feed in?

5 HIDDEN LAYERS

The outputs are mixed with varying weights, shown by the thickness of the lines.

This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.002

Training loss 0.002

Colors shows data, neuron and weight values.

☑ Show test data □ Discretize output

Ratio of training to test data: 50%

Batch size: 10

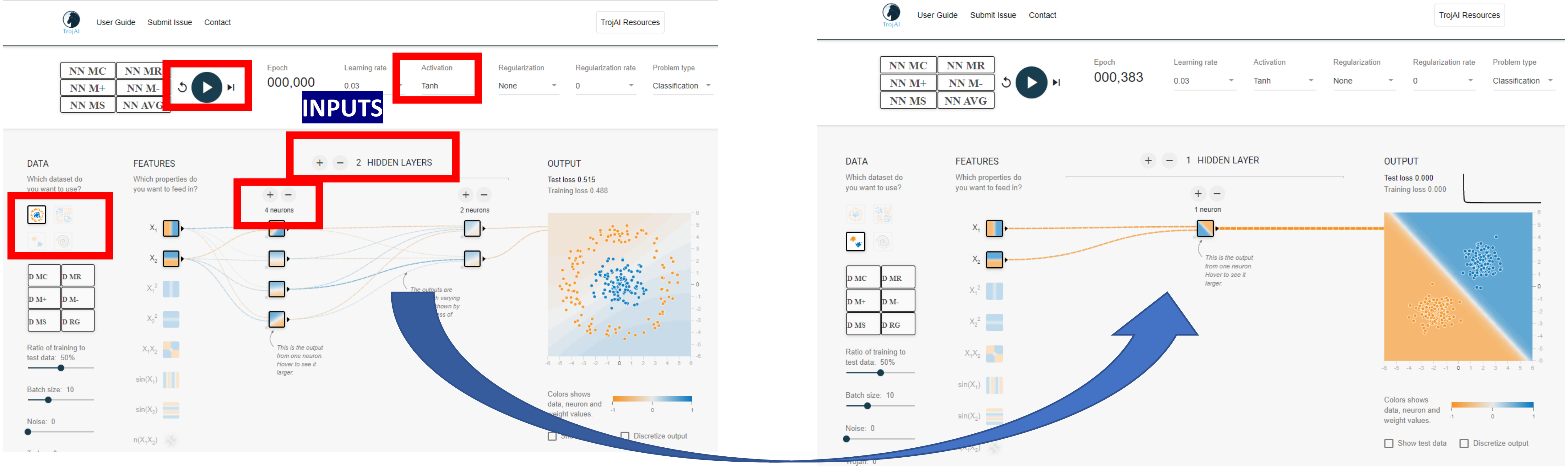
Noise: 0

Trojan: 0

NN calculator: https://pages.nist.gov/nn-calculator/

EX #1: Modify the NN to a single node

1. Remove layers
2. Remove nodes
3. Select data pattern with cluster pattern
4. Select Linear activation function



User Guide Submit Issue Contact TrojAI Resources

Epoch 000,000 · Learning rate 0.03 · Activation Tanh · Regularization None · Regularization rate 0 · Problem type Classification

2 HIDDEN LAYERS · 4 neurons · 2 neurons

OUTPUT: Test loss 0.515, Training loss 0.488

Ratio of training to test data: 50% · Batch size: 10 · Noise: 0

User Guide Submit Issue Contact TrojAI Resources

Epoch 000,383 · Learning rate 0.03 · Activation Tanh · Regularization None · Regularization rate 0 · Problem type Classification

1 HIDDEN LAYER · 1 neuron

OUTPUT: Test loss 0.000, Training loss 0.000

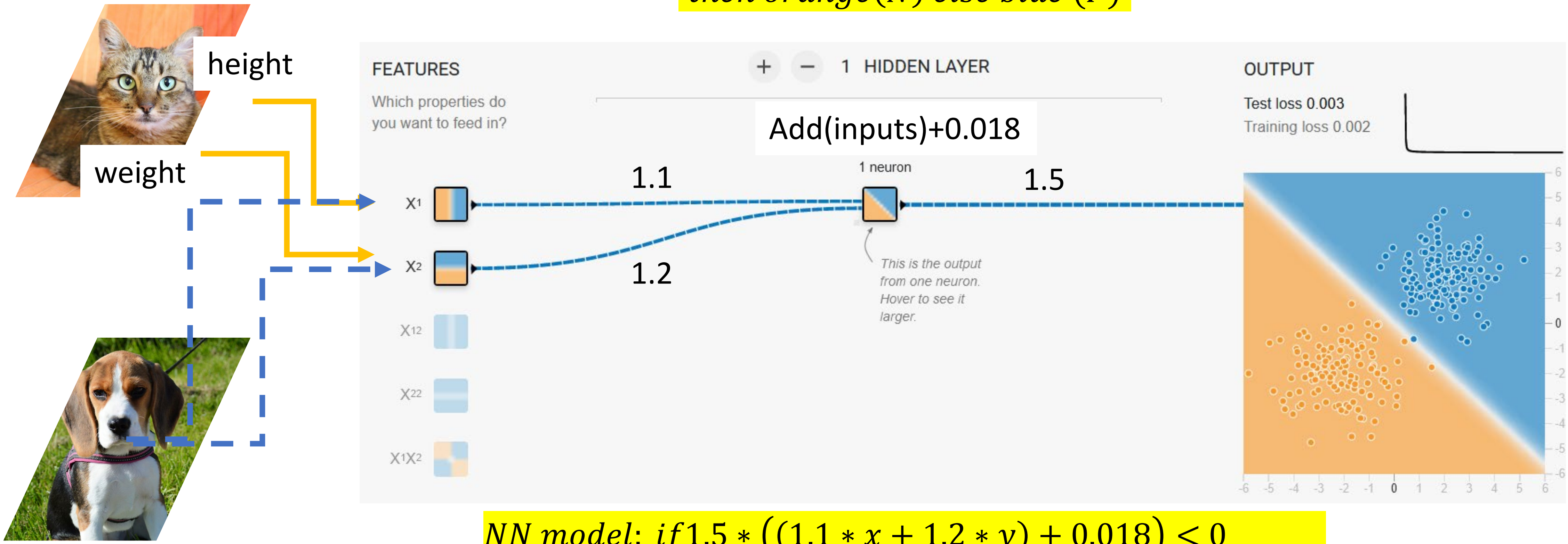
Ratio of training to test data: 50% · Batch size: 10 · Noise: 0

□ Show test data □ Discretize output

Single node NN with Linear activation function

$$\mathit{Math\ model:\ if}\ x + y < 0$$
$$\mathit{then\ orange}(N)\ \mathit{else\ blue}\ (P)$$

Linear Activation



Which properties do you want to feed in?

Test loss 0.003
Training loss 0.002

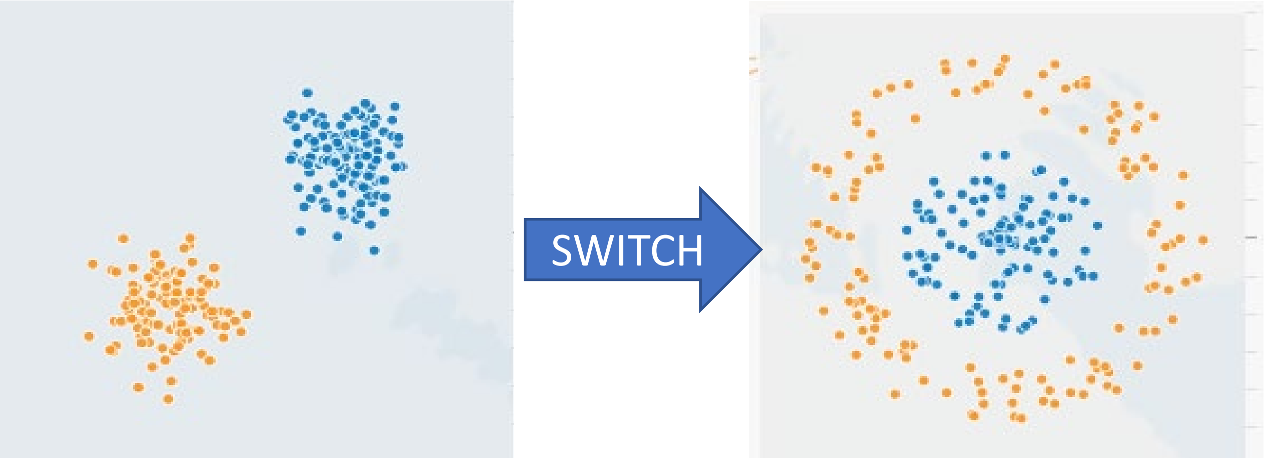
This is the output from one neuron. Hover to see it larger.

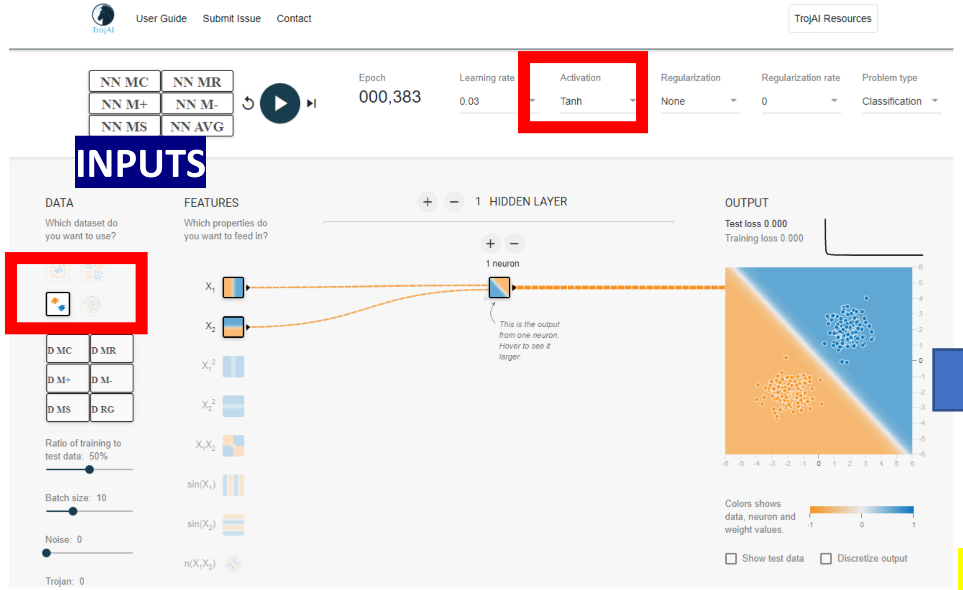
$$\mathit{NN\ model:\ if}\, 1.5 * \big((1.1 * x + 1.2 * y) + 0.018\big) < 0$$
$$\mathit{then\ orange}\ (N)\ \mathit{else\ blue}\ (P)$$

If 1.65 * x + 1.8 * y + 0.027 < 0 then orange (N) else blue P)

Complexity of Data Patterns

- Switch data patterns
- Train





Epoch 000,383

Learning rate 0.03

Activation Tanh

Regularization None

Regularization rate 0

Problem type Classification

Which dataset do you want to use?

Which properties do you want to feed in?

Ratio of training to test data: 50%

Batch size: 10

Noise: 0

Trojan: 0

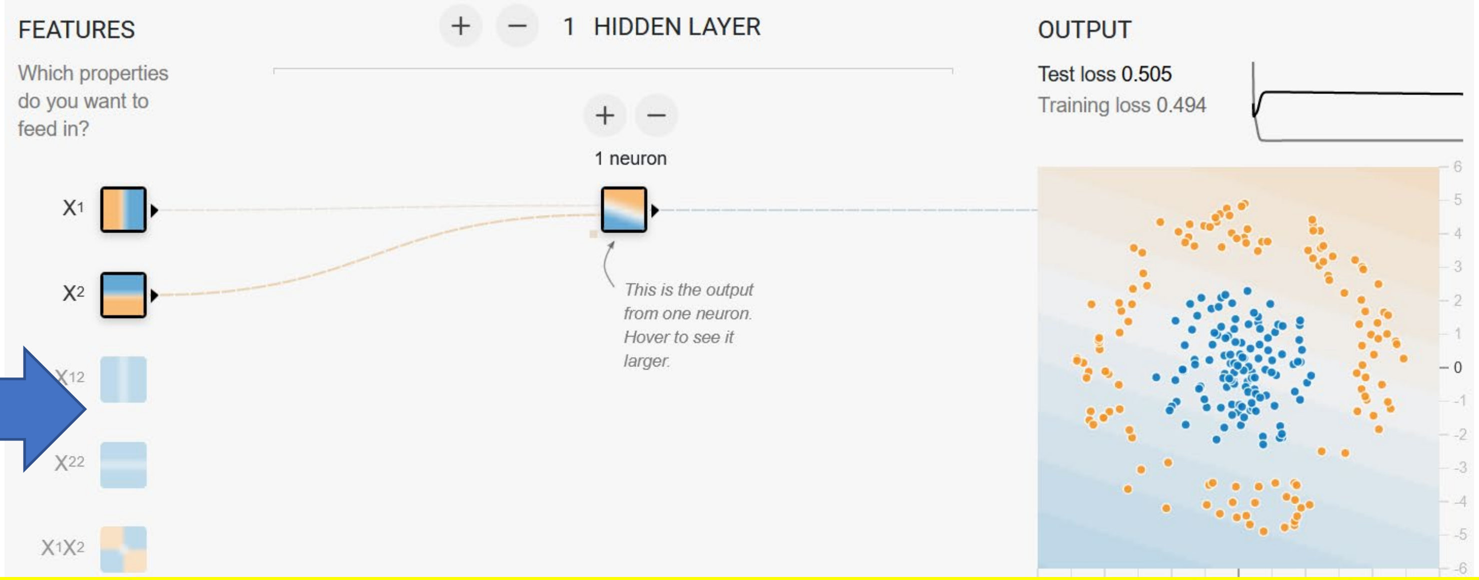
This is the output from one neuron. Hover to see it larger.

Test loss 0.000

Training loss 0.000

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output



Which properties do you want to feed in?

This is the output from one neuron. Hover to see it larger.

Test loss 0.505

Training loss 0.494

NN model: FAILED

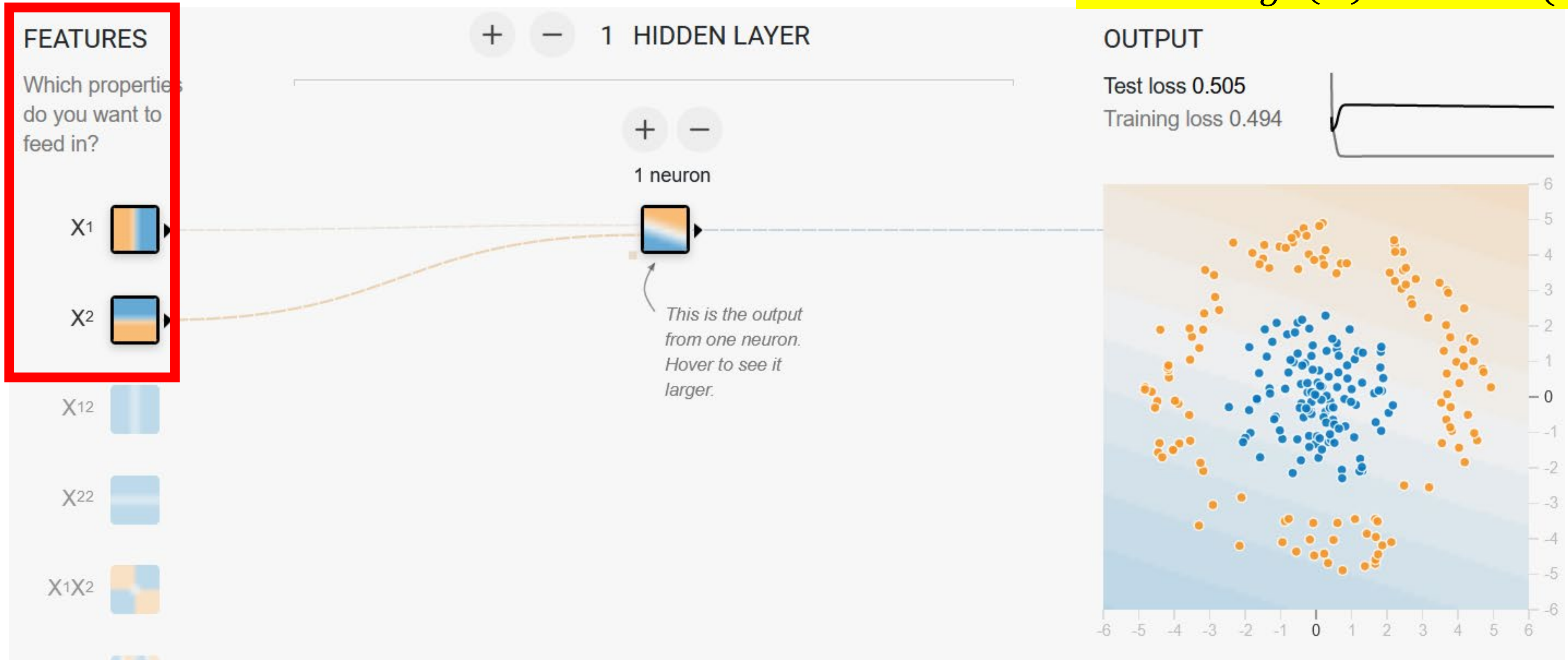
— could not come up with an accurate labeling formula based on x & y

Why did the model fail?

Linear Activation

$Math\colon if\ x^2 + y^2 > r^2$
$then\ orange(N)\ else\ blue\ (P)$

INPUTS



FEATURES

Which properties do you want to feed in?

+ − 1 HIDDEN LAYER

OUTPUT

Test loss 0.505
Training loss 0.494

This is the output from one neuron. Hover to see it larger.

$NN\ model\colon FAILED$
$-\ could\ not\ come\ up\ with\ an\ accurate\ labeling\ formula\ based\ on\ x\ \&\ y$

Solution 1:

- Create non-linear features

Solution 2:

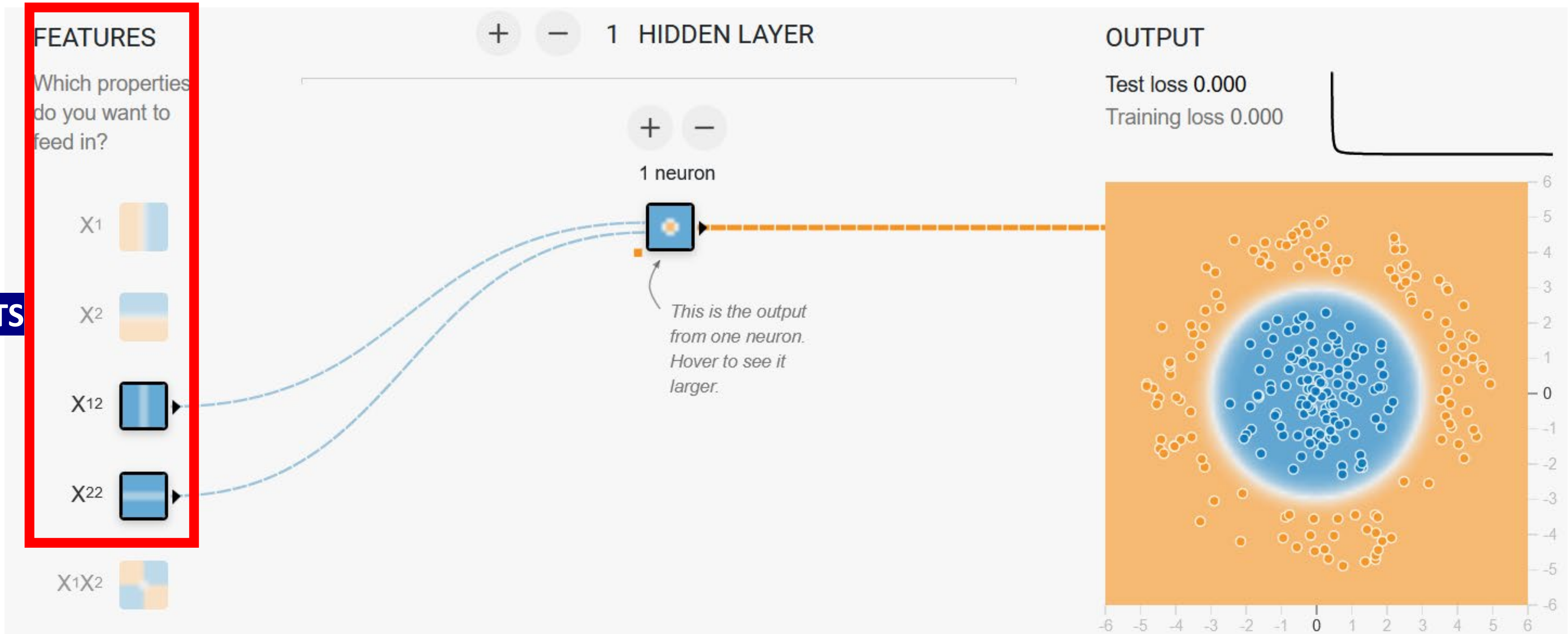
- Add 3 more nodes
- Switch to Tanh activation function

EX #2: Non-Linear Features (x^2 and y^2)

$$Math:\ if\ x^2 + y^2 > r^2$$
$$then\ orange(N)\ else\ blue\ (P)$$

Linear Activation

INPUTS



Which properties do you want to feed in?

This is the output from one neuron. Hover to see it larger.

Test loss 0.000

Training loss 0.000

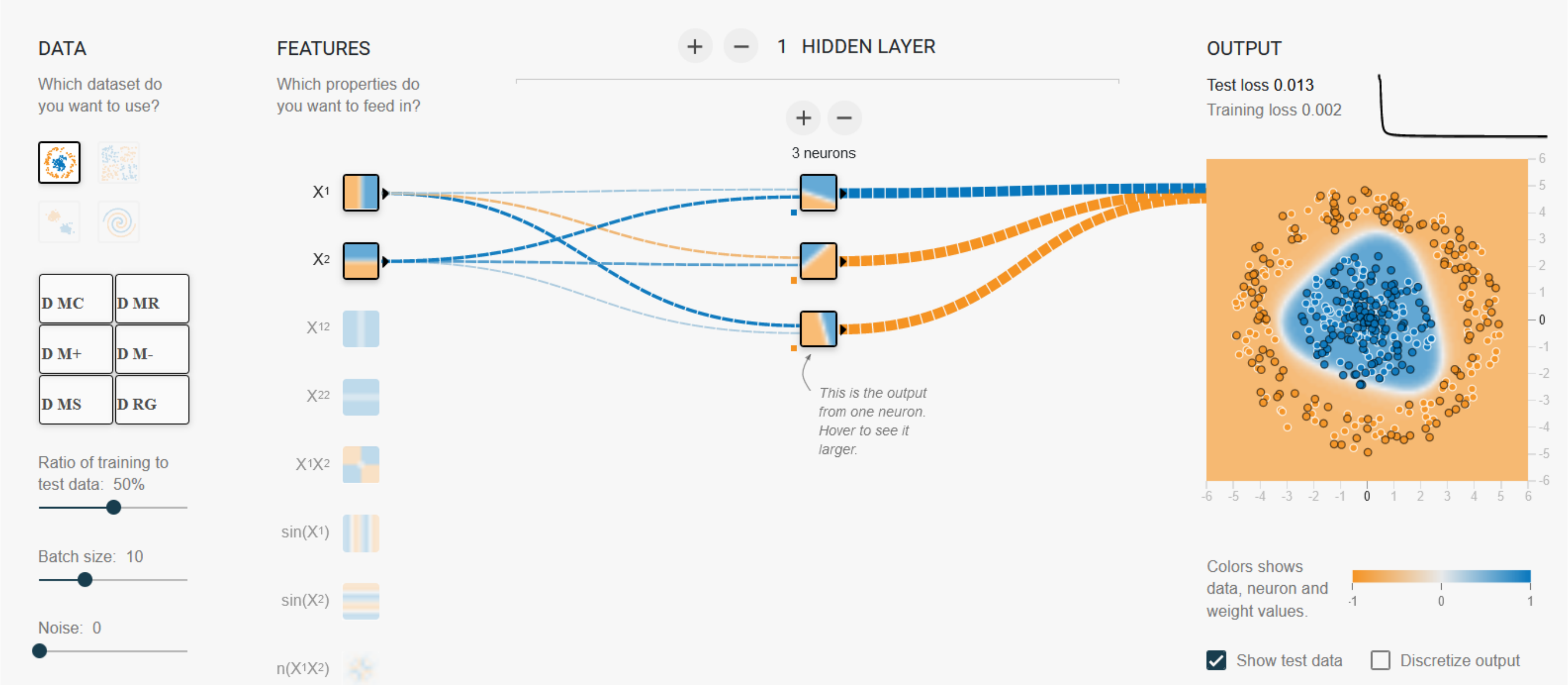
$$NN\ model:\ if - 1.3 * \left((0.33 * x^2 + 0.34 * y^2) - 1.3\right) < 0$$
$$then\ orange(N)\ else\ blue\ (P)$$

If - 0.429 * x^2 - 0.442 * y^2 + 1.69 < 0 then orange (N) else blue (P) → radius ~ Sqrt(1.69/0.433) = 1.97

EX #3: Non-Linear Tanh Activation

$Math: if\ x^2 + y^2 > r^2$
$then\ orange(N)\ else\ blue\ (P)$

Tanh Activation



DATA

Which dataset do you want to use?

D MC | D MR | D M+ | D M- | D MS | D RG

Ratio of training to test data: 50%

Batch size: 10

Noise: 0

FEATURES

Which properties do you want to feed in?

1 HIDDEN LAYER

3 neurons

This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.013
Training loss 0.002

Colors shows data, neuron and weight values.

☑ Show test data □ Discretize output

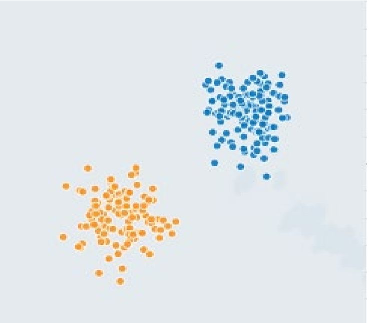
Test loss 0.013
Training loss 0.002

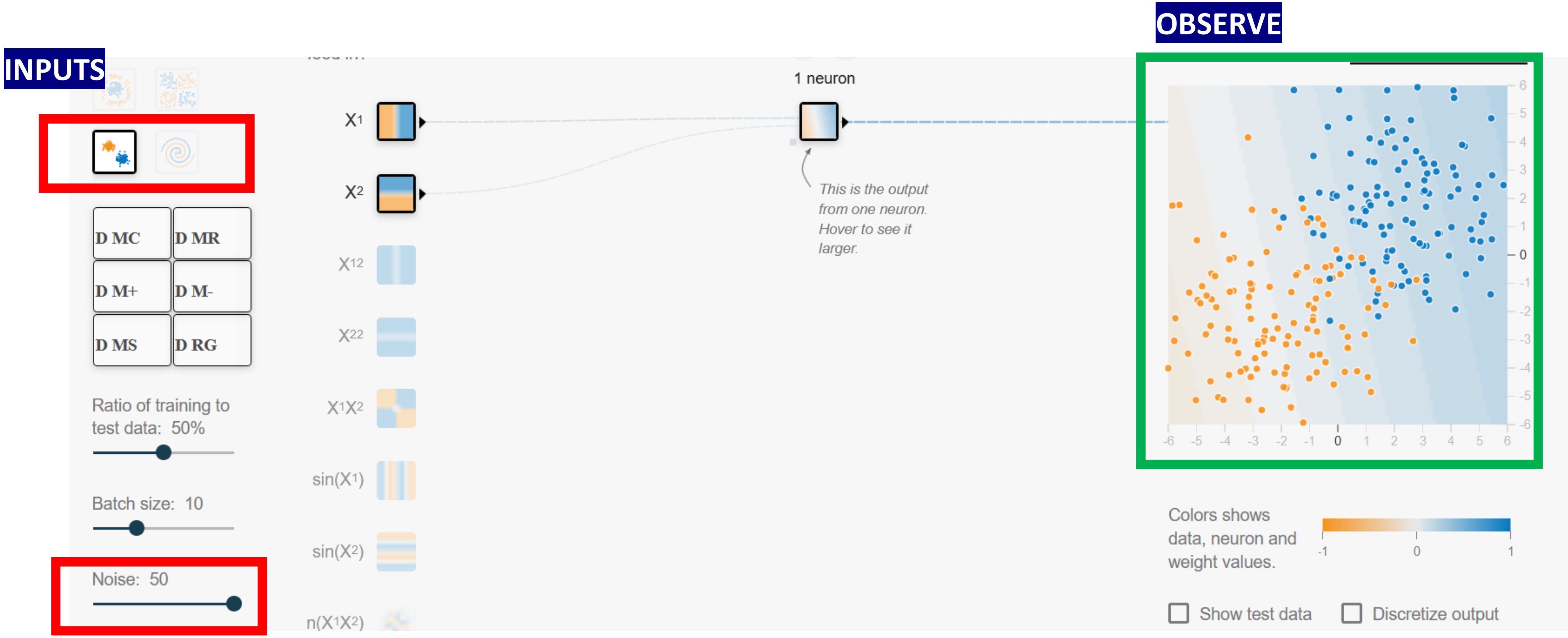
$NN\ model:\ if$ 4 * tanh(1.7 + 0.32 * x + 0.94 * y) – 4 * tanh(-1.9 - 0.59 *x + 0.66 * y) – 4 * tanh(-2.0 + 0.96 * x + 0.27 * y) < 0
$then\ orange\ (N)\ else\ blue\ (P)$

EX #4: Test well-separated and interleaved patterns

Select a dataset with cluster pattern

Use the “Noise” slider bar





This is the output from one neuron. Hover to see it larger.

Ratio of training to test data: 50%

Batch size: 10

Noise: 50

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output