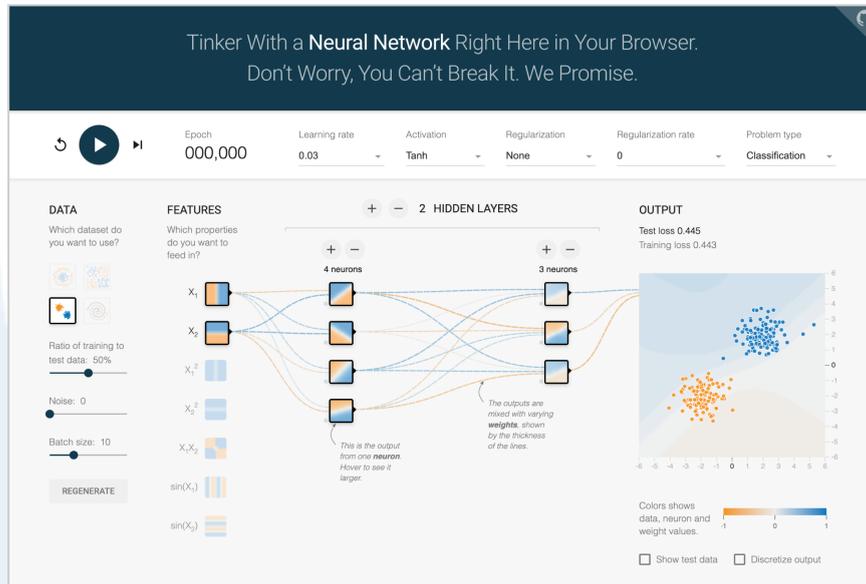




# Neuronale Netze: Wie der Computer lernen kann, Pong zu spielen

Heute habt ihr erst Neuronale Netze kennengelernt und trainiert, um orange von blauen Punkten zu trennen.



<http://playground.tensorflow.org/>

Danach habt ihr in einem Jupyter Notebook mit Python selbst ein Neuronales Netzwerk definiert und trainiert, das gelernt hat, Pong zu spielen.

```
Defining the structure of the neural network

Let's not rely on the code to automatically create a neural network for us. What do you think, from what we did earlier, influences how the neural network must be structured?

Remember, we use a list notation to define the structure of the neural network. E.g. E.g. [4, 2, 3, 1] created the following neural network:

In [ ]: from P11 import Image
        im = Image.open("nn_structure.png")
        display(im)

Implement your own network structure!

In [ ]: def get_network_structure() -> List[int]:
        # a neural network with 1 input, two hidden layers of size 2 and 2 output
        return [1, 2, 2, 2]
        strategy_patterns.strategies[strategy.NETWORK_STRUCTURE] = get_network_structure

Let's see if it works!

In [ ]: train(training_time=60)

Defining a reward function

Now we will define a reward function i.e. tell the neural network what it did right and what it did wrong. But first, let's define a few helper functions:

In [ ]: def enemy_scored(observation: Pong, next_observation: Pong) -> bool:
        """
        Did the enemy score a point?
        operan observations how the game state looks like shortly after
        operan next_observation: how the game state looks like shortly after
        (returns) True if the enemy scored a point, False otherwise
        notes: The enemy is the right side player!
        """
        return True # replace with your own logic
```

Alle Links zum Projekt findet ihr auf der Webseite des InfoLab: [infolab.cs.uni-saarland.de](http://infolab.cs.uni-saarland.de)

Wir freuen uns, dass ihr das InfoLab besucht habt!

