Hayden Wood

Jade Sanchez

Joshua Herold

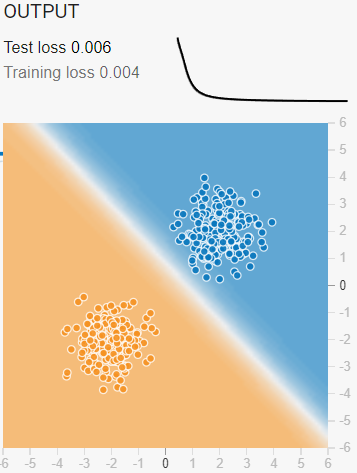
Jacob Ehling

A06 TensorFlow Playground

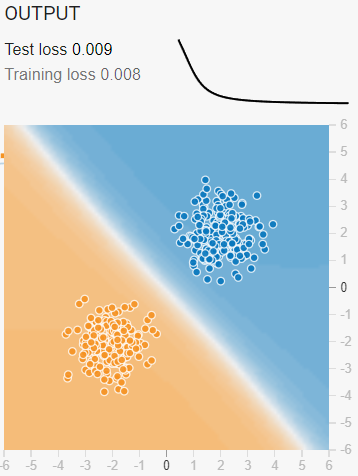
**Activation Functions**

Activation functions in neural networks are mathematical functions applied to the output of each neuron, deciding whether that neuron should be activated (i.e., fired) or not. They introduce non-linearity into the network, a crucial factor that enables the network to learn from complex data. Without activation functions, a neural network would essentially act like a simple linear regression model and couldn't model more complicated patterns.

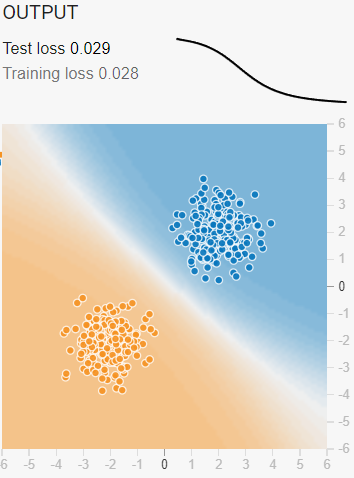
* ReLU outputs the input if it's positive; otherwise, it outputs zero. Its simplicity and capacity to handle the vanishing gradient issue better than others help to explain its general popularity. However, it can suffer from "dying ReLU" when neurons get stuck and only output zero.



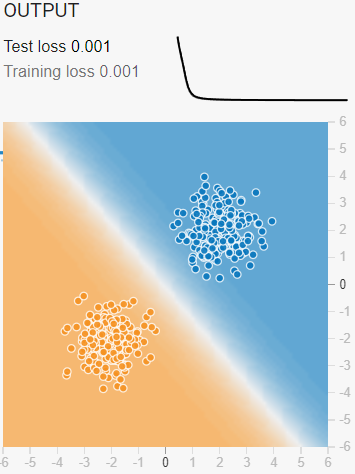
* Tanh squashes values between -1 and 1, which can help center the data around zero. This often leads to faster learning than Sigmoid but still suffers from vanishing gradients for significant inputs.



* Sigmoid squashes values between 0 and 1, making it useful for binary classification problems. However, it can cause vanishing gradients because its output tends to be small for large positive or negative inputs.



* The linear activation function returns the input, without introducing non-linearity. It's often used in the output layer of regression tasks where you need continuous output.

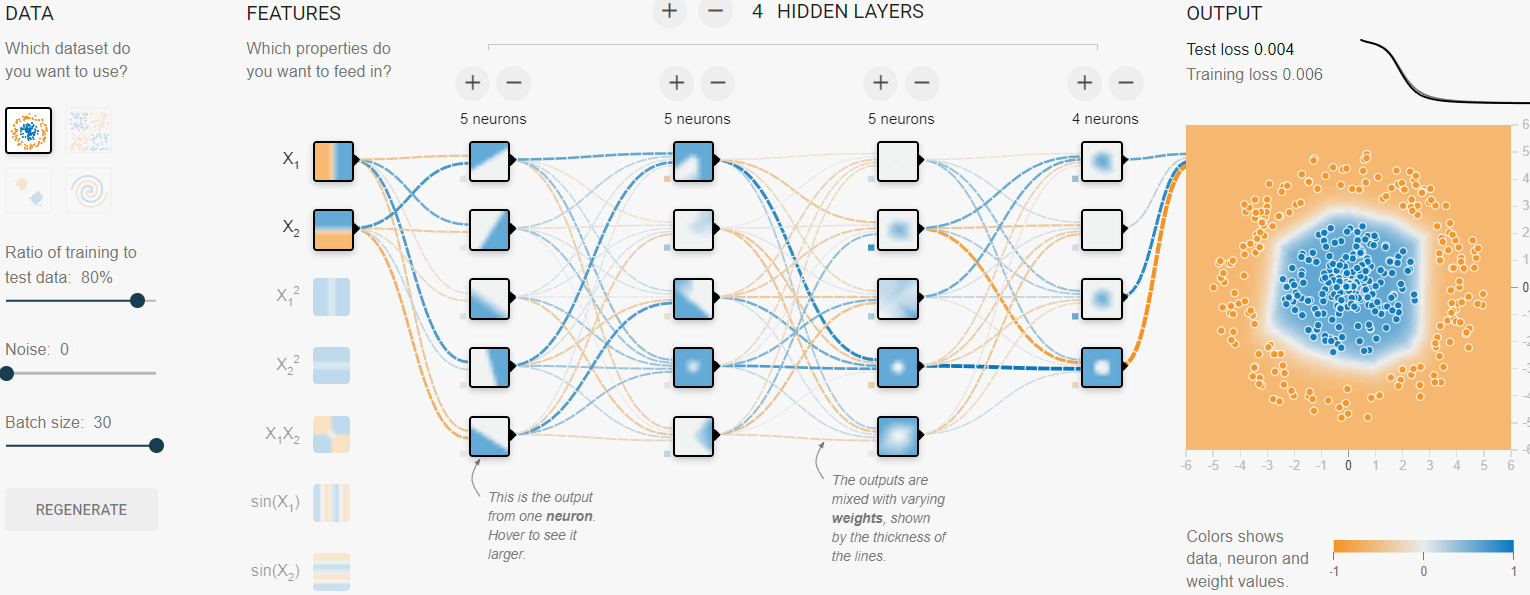


**Hidden Layer Neurons**

A neuron is a fundamental unit of a neural network responsible for receiving inputs, applying a weighted sum, and applying an activation function to those inputs to generate an output. Neurons are organized into layers:

* Input layer: The point at which the network receives its raw data packets.
* Hidden layers: Process the data by performing weighted calculations and activation. More hidden layers can allow the network to learn more complex patterns. Each neuron in the hidden layer is linked to those in the preceding and following layers.
* Output layer: The final layer provides the network's predictions or classifications.

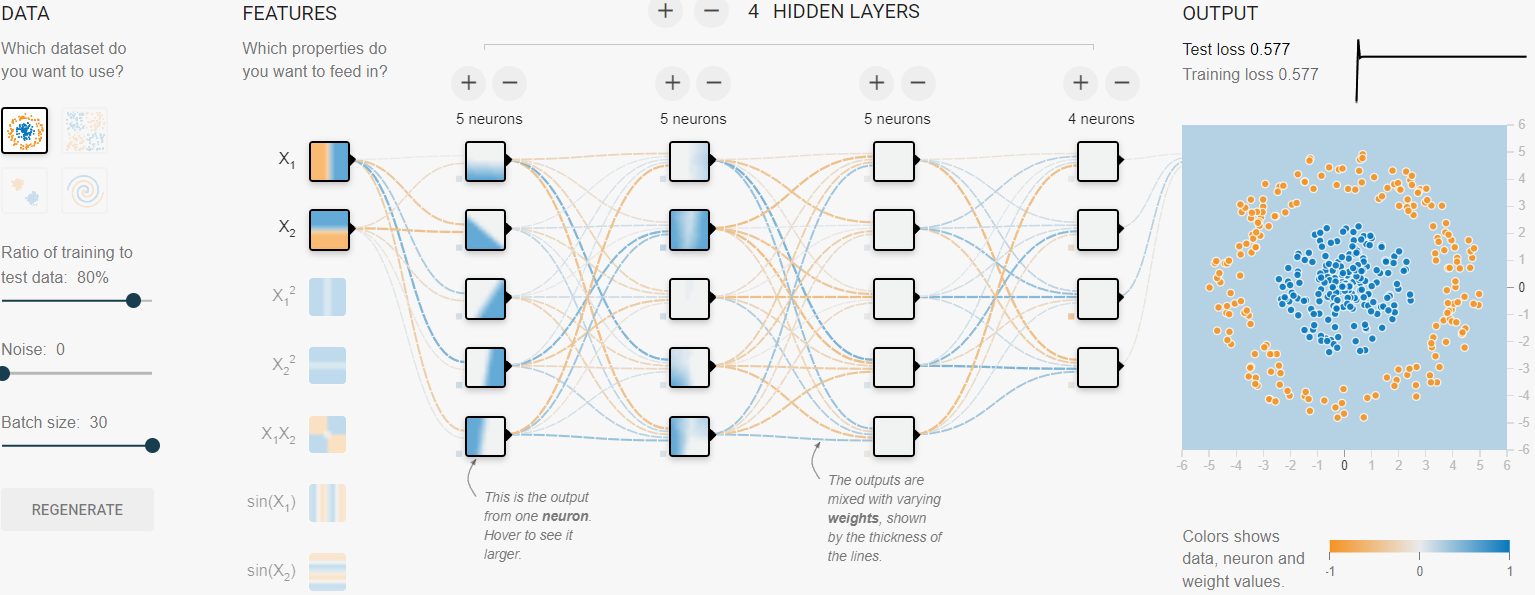
The hidden layers and activation functions help the neural network learn intricate and abstract patterns from the data.



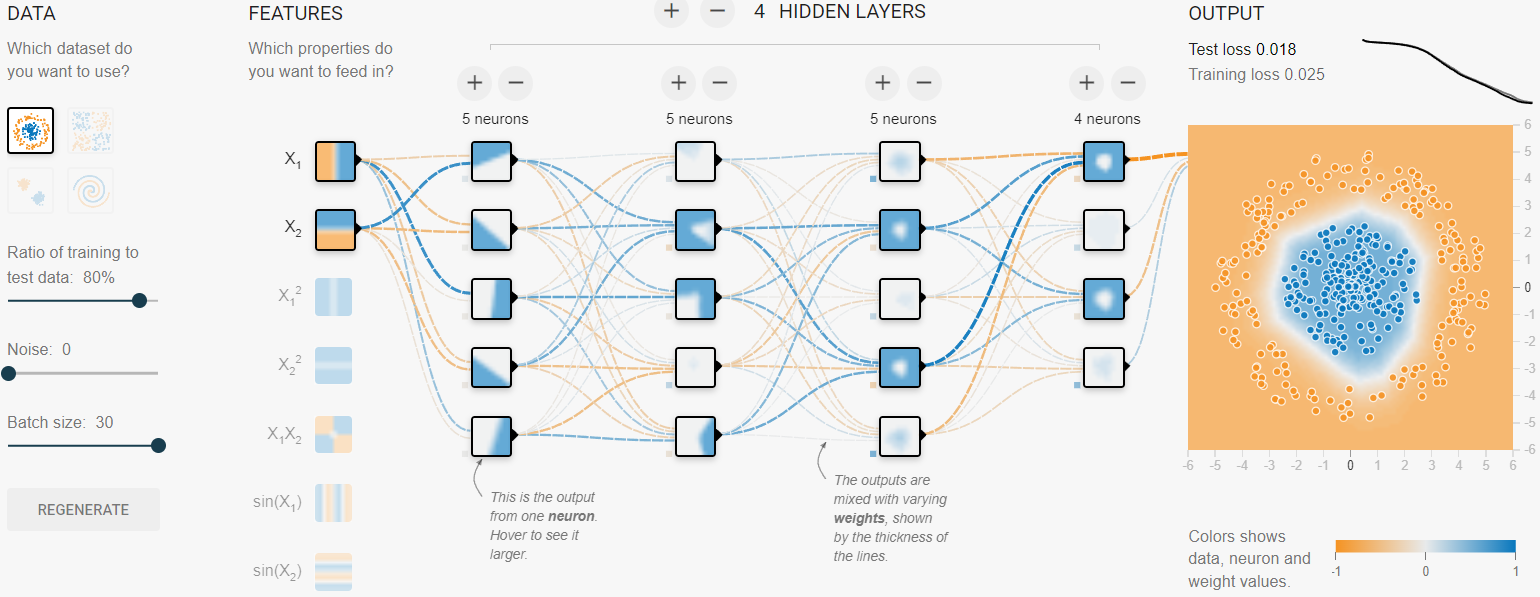
**Learning Rate**

The learning rate is a hyperparameter that determines how much the model's weights need to be adjusted concerning the loss gradient during each training iteration. It's crucial for the training process because:

* A high learning rate can cause the model to converge too quickly, potentially missing the optimal solution. (Example learning rate was 3)



* A low learning rate allows for more precise adjustments, but it can slow the training and risk getting stuck in a local minimum. (Example learning rate was 0.001)

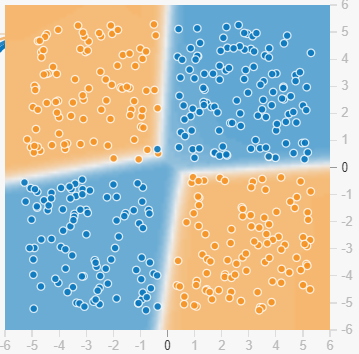
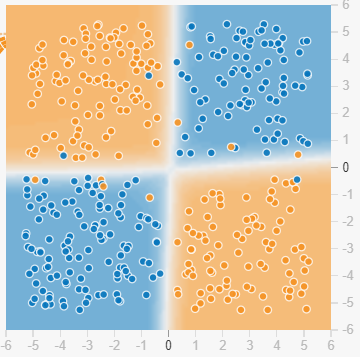


**Data Noise**

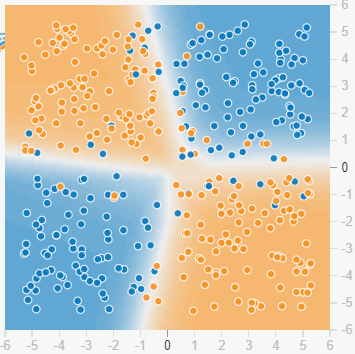
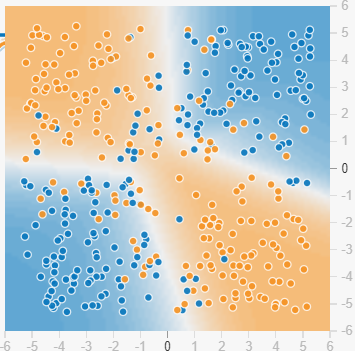
Data noise refers to irrelevant or random variations in the dataset that don't represent the underlying patterns but can still affect the model's training. For instance, measurement errors could be a sensor malfunctioning and recording incorrect data, missing values could be due to a malfunctioning data collection system, and outliers could be extreme values that are significantly different from the rest of the data.

* Overfitting: If a model learns to fit the noise in the training data, it may perform poorly on new, unseen data because it memorizes the noise rather than the actual patterns.
* Reduced accuracy: Noise can degrade the model's ability to generalize and make accurate predictions.
* Longer training time: The model might take longer to converge when learning from noisy data.

Noise: 10 Noise: 20

Noise: 35 Noise: 50

**Dataset Exploration**

**Dataset Options:**

* Circle: In this dataset, points are arranged in a circular pattern. This dataset poses a significant challenge as it is non-linear, meaning it can't be easily separated by a straight line (linear boundary). It necessitates using a more complex neural network to learn the intricate relationships and classify the points accurately.
* Gaussian: This dataset consists of two clusters of points generated from Gaussian distributions. The points can be separated by a linear boundary, making this dataset more manageable for a neural network to learn. Simpler models with fewer hidden layers or linear classifiers can easily handle this dataset, showcasing their adaptability.
* Exclusive OR: This dataset is a classic example in machine learning, where the points are arranged similarly to a checkerboard, so no single straight line can separate them. It remains a significant challenge in the field, requiring a more complex network because of its non-linearity.
* Spiral: This dataset consists of points arranged in a spiral shape. Like the circle dataset, it is highly non-linear, and simple models struggle to learn it. It typically requires more hidden layers and non-linear activation functions to classify the points accurately.

**Importance of Dataset Selection:**

Choosing the suitable dataset is critical in neural network training for several reasons:

* Complexity Matching: Your dataset's complexity should match your model's capacity. Simple datasets (like Gaussian clusters) can be solved with simpler models, while more complex datasets (like the spiral or exclusive OR) need more sophisticated architectures. If the model is too simple for a complex dataset, it won't be able to learn the underlying patterns, leading to underfitting. Conversely, if the model is too complex for a simple dataset, it may overfit, learning noise instead of the actual pattern.
* Model Generalization: The choice of dataset affects how well the model generalizes to new, unseen data. Datasets with more complex patterns push the network to learn deeper representations, which can improve generalization if done correctly.
* Dataset Distribution: If your dataset doesn't represent the real-world data the model will encounter, it can lead to poor performance. For example, a network trained on the Gaussian Clusters might perform well there but fail on the Exclusive OR dataset because it has yet to learn non-linear patterns.