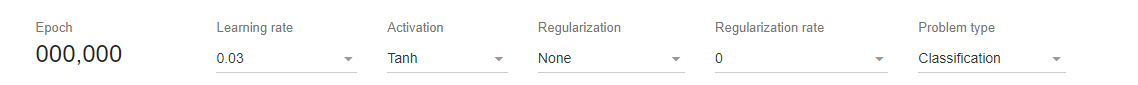
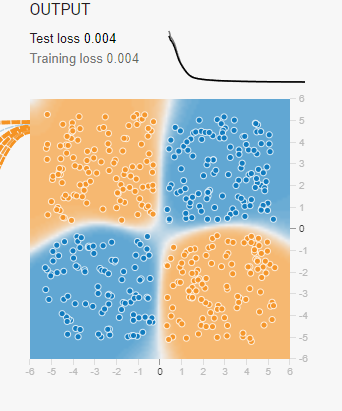
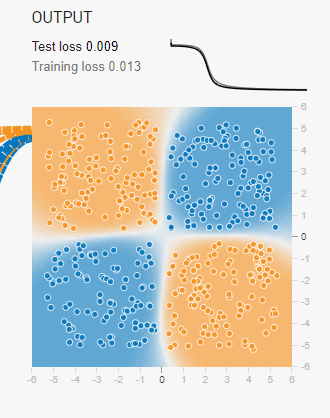
Rayyaan Haamid  
Patricia McManus  
Computer Vision  
February 26, 2024

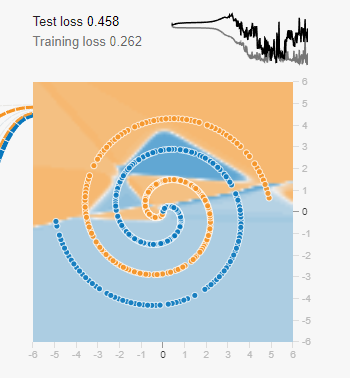
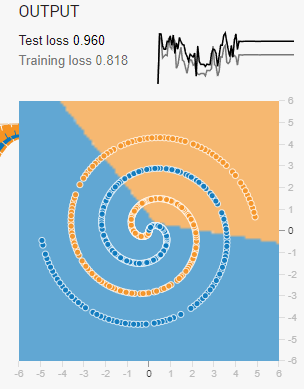
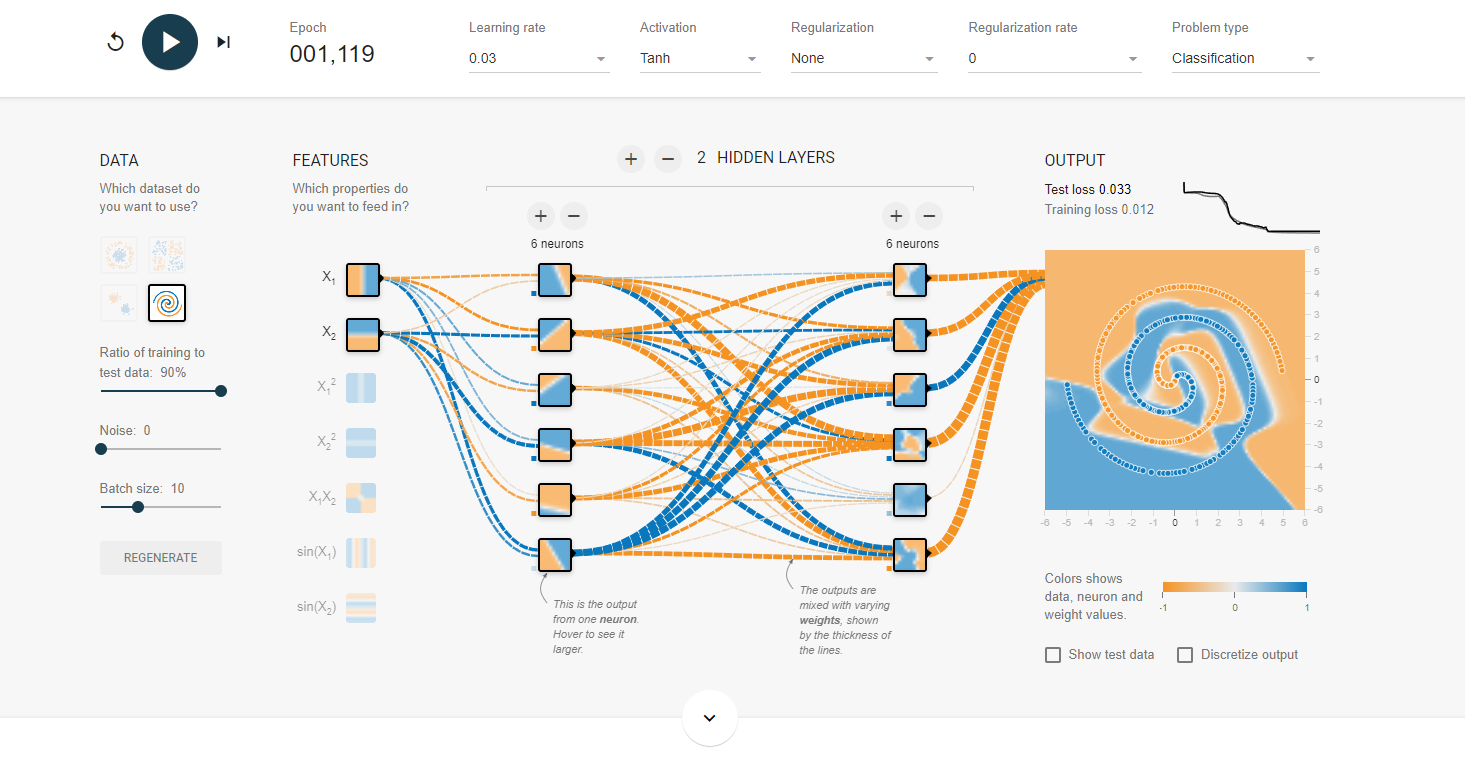
L05 – Tenserflow playground

Neural network is a machine learning program or model that operates in a form of a brain or at least mimicking a human brain. It utilizes itself with the processes of biological neurons that relate itself to identifying patterns, weigh options, and arrive at conclusions. As for its components, the input layer; which receives the initial data. Hidden layer; the middle layer that process the information given. Output layer; Produces the final result. And this connects together with weight and threshold. Weights determine the importance of input variables. Biases (Thresholds); adjust the overall output of a node. Activation function; after multiplying input data by their respective weights and summing them up, the output passes through an activation function. Common activation functions include sigmoid, ReLU, and tanh. (in this case). Training data and learning; neural networks rely on training data to learn and improve their accuracy. Once fine-tuned. Neural networks excel at tasks like speech recognition, image classification, and more.  
  
(some of the components that affects the model)



The activation functions changes how the calculation is done in the neural network, each of them has their own way to make the result in their own form. For example, linear answer for the result is straight but if introduced curving, it loses its accuracy. Tanh shows the curviness for the training data which can be situational. RELU allows the use of more straight points and makes it able to be flexible compared to linear. Sigmoid is similar to tanh, it has training data curved but more linear when training it compared to tanh.

  
this is tanh activation function  
  
  
this is sigmoid activation function

The increase of hidden layer which increases the number of neurons complexifies the training data and makes it faster but there is a catch to it, it is prone to overfit it with information and destroys the accuracy.   
  
this is similar to the learning rate too.  
  
(the high learning rate problem)  
  
The learning rate is similar to the result you get from the hidden layers, if it is increased it destroys the accuracy but shows incredibly fast pace to the learning pace of it, same thing with a lower learning rate, it is much slower but has the possibility to make accurate results. Noise makes the dataset dispersed and randomizes the data to the non-accurate part of the dataset. It allows the model to make a distinction to data that is different from the normal data that is already fed to it, it makes a more robust version of the model that allows unseen examples to be recognized, making it noticeable to the model. In the TensorFlow playground, there are four types of datasets with its own components to change how the training will happen. Firstly, the circle with the outer layer of orange. This dataset mainly has the curvature and mainly makes curved results, activation functions like linear doesn’t work in it and rather breaks itself with a large testing and training loss. Secondly, the dataset that is divided with quadrants. It is a linear based dataset that allows all of the activation functions to work with it but is required to place the feature X1X2 to make sure you get a result from it. Thirdly, it is a definite two sides that allow to make a linear format from all of the activation functions, it really depends on the features, you can change it to get different results like a swirly line, straight line, curved line, etc. Lastly, the spiral dataset, this is the most complicated version of all the datasets and frankly, I still have been able to crack it with an hour just to try to make the testing and training loss minimized, the only time I got close was using tanh and two hidden layer with 0.150 testing and training loss.  
  
(My best attempt to the spiral)  


The practicality of TensorFlow playground is an amazing tool to understand neural networks. It provides intuitive learning, problem solving by the computer and allow it to learn data and provide results from it. I was able to experiment with various hyperparameters like hidden layers, learning rates, epochs, activation functions, and regularization.

In conclusion, Neural networks mimic the brain’s processes to identify patterns and make decisions. Components: Input layer (receives data), hidden layer (processes information), output layer (produces results). Weights and biases adjust node behavior. Activation functions (e.g., sigmoid, ReLU, tanh) shape the output. Training data fine-tunes the network. Be cautious with hidden layers—too many can lead to overfitting. Neural networks excel at tasks like speech recognition and image classification. High learning rate: Fast learning but may overshoot optimal values. Low learning rate: Slower but more accurate convergence. Balance is crucial. Noise: Adds randomness to the dataset. Helps the model generalize better. Robustness to unseen examples. TensorFlow Playground Datasets: Circle: Curved dataset, linear activation fails. Quadrants: Linear-based, works with all activations. Two Sides: Linear format, depends on features. Spiral: Complex, challenging, requires careful tuning.