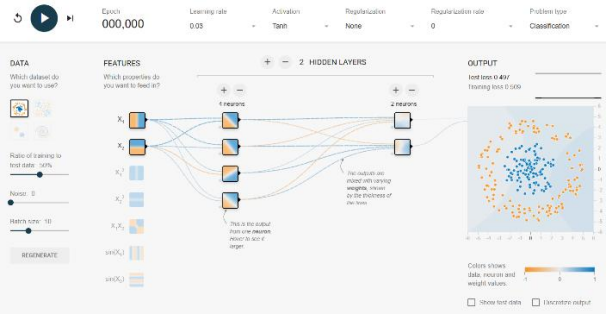
Jessica Navarro
ITAI 1378
A06

TensorFlow Playground Presentation

The goal of this assignment was to help me understand the basics of neural networks and how they function. I used TensorFlow Playground, an interactive web-based tool, to explore configurations and observe their performance.

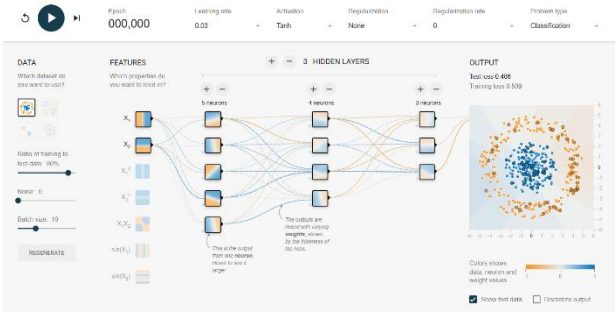
A neural network structure is similar to a human brain's structure and function. A neural network consists of connected neurons, organized in layers. An artificial neuron, also known as perceptron, is the basic unit of a neural network. It's important to not that each neural network must consist of an input layer, at least a hidden layer and an output layer. The neuron/s in the input layer receives the input data a weight is assigned. Each input has at least 1 weight, which determines the importance. In addition, a bias, a parameter that's adjusted during training, is added to the function to help the model make better decisions. The data is then passed an activation function, to the hidden layers, where patterns are identified, and predictions are made. The predicted output is generated in the output layer.

The most common activated functions used are Sigmoid, Tanh, and ReLU. Sigmoid is primarily used for binary classification. Tanh is primarily used for multi-class classification problems. ReLU (Rectified Linear Unit) is the most commonly used. It introduces non-linearity into the model. Non-linearity allows the model to learn complex patterns, rather than simple linear relationships.



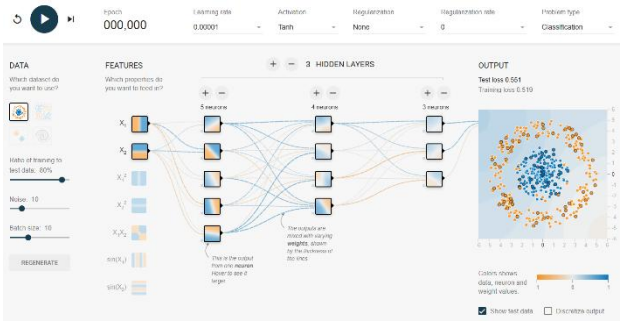
In the TensorFlow model, the color orange shows negative values and blue shows positive values. This includes the data points, hidden layers, and output layers. Throughout the model I chose to use a classification problem. This image shows the default parameters, prior to any experiment.

I increased the training data to 80% and decreased the test data to 20%. I increased the hidden layers to 3. The first layer had 5 neurons, the second layer had 4 neurons, and the third layer had 3 neurons. The rest of the parameters stayed the same. The test loss (0.486) was less than the training loss (0.509). This suggested that the model is not overfitting. It actually performs slightly better on unseen test data. A loss of 0.4–0.5 generally means the model is making decent predictions but still has room for improvement.

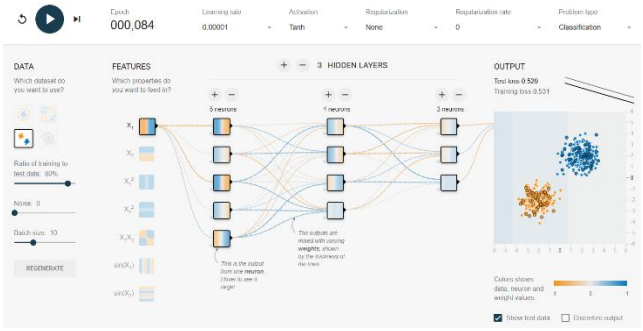


Jessica Navarro
ITAI 1378
A06

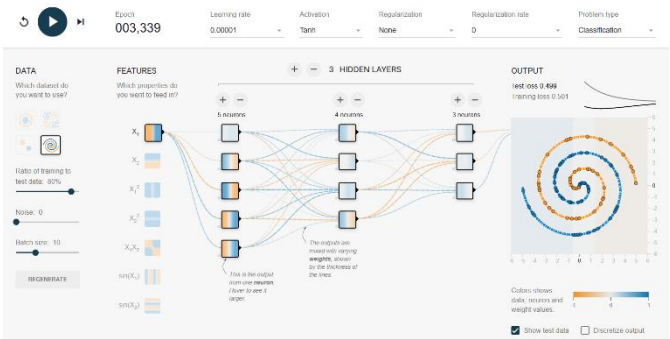
The learning rate is a hyperparameter that controls the size of weight updates during training. It determines how quickly or slowly a neural network adapts to patterns in the data. I decreased the learning rate to 0.001. A low learning rate means more stable training and more precise convergence. Then, I increased the learning rate to 10. A high learning rate means faster initial learning and can escape shallow local minima.



Data noise refers to random fluctuations or inaccuracies in the dataset that do not reflect the actual underlying pattern. Increasing the "Noise" slider in TensorFlow Playground introduces random disruptions to the dataset, making it more challenging for the neural network to identify clear patterns. I increased the noise to 10. This resulted in a test loss of 0.551 and a training loss of 0.519. A higher test loss suggests some level of overfitting, where the model memorizes the training data too well but struggles to generalize to new examples.



I switched to the Gaussian dataset; removed the noise, and kept the other parameters. The Gaussian dataset is relatively simple and usually follows a linear pattern, making it easier for the neural network to learn and generalize. When I ran Epoch, the training loss and test loss lines quickly ran parallel to each other. The losses (training and test) quickly converge because there is less complexity to model, and the neural network doesn't need to spend much time fine-tuning its weights. The Spiral dataset is more complex and involves non-linear patterns. When I switched to the Spiral dataset, the training loss and test loss lines took a while to run parallel. They did not exactly become parallel lines though. They curved next to each other and then ran parallel. The model initially struggles to capture the non-linear patterns in the spiral data, which is why the loss curves are not parallel initially.



This assignment helped me understand and explore neural networks using TensorFlow Playground and other resources.

Jessica Navarro
ITAI 1378
A06

References

TensorFlow Playground. (n.d.). *An interactive tool for exploring neural networks*. Retrieved from https://playground.tensorflow.org

Houston Community College. (n.d.). *[ITAI_1378_2024_Module_06_Intro_to_NN.pptx]*. Retrieved from https://eagleonline.hccs.edu/courses/278598/files/70587251?module_item_id=18927460