Jesus Gudino

02/27/25

ITAI 1378

Professor: Anna Devarakonda

**Neural Network Experiment Report**

1. **Introduction**

Neural networks are like mini brains for computers. They help with things like recognizing faces, predicting weather, and even recommending what to watch on Netflix. The goal of this experiment was to see how different settings- like the number of neurons, activation functions, and learning rates- affect how well a neural network learns. I used TensorFlow Playground to test these changes and recorded what happened.

1. **What I Did in Each Task**

Task 1: Activation Functions

I started by testing different activation functions, which decide how much information moves forward in the network.

* ReLU (Rectified Linear Unit) worked the fastest but didn’t do well when I only had one neuron.
* Sigmoid and Tanh took longer to learn but were smoother.
* Linear activation didn’t work well at all, it couldn’t recognize the shapes in the data.

A screenshot of a computer

AI-generated content may be incorrect.Observations: When I used only one Neuron with ReLU, the decision boundary was just a straight line. It couldn’t capture the circular shape of the data at all. I had to add more neurons to fix this.

Task 2: Hidden Layer Neurons

Next, I changed the number of neurons in the hidden layer to see how it affected learning.

* 1 neuron: Couldn’t figure out the shape at all.
* 4 neurons: Finally learned the circular pattern.
* 8 neurons: Worked well but might have been too much- could memorize instead of learning real patterns.

A screenshot of a computer

AI-generated content may be incorrect.Observations: With only one neuron, the decision boundary was almost useless. But when I added more, the network started shaping itself to fit the data.

Task 3: Learning Rate

The learning rate controls how fast the network adjusts when it makes mistakes.

* A low learning rate (0.001) was too slow. The network took forever to learn anything.
* A good learning rate (0.03) worked best. It was fast and accurate.
* A high learning rate (3.0) was a disaster. The model kept jumping around and never actually learned.

A screenshot of a computer

AI-generated content may be incorrect.Observations: When I set the learning rate to 3.0, the loss graph was all over the place. The decision boundary kept changing and never settled.

Task 4: Data Noise

I added noise to the data, which simulates messy real-world data.

* No noise (0): The network learned perfectly.
* Some noise (10): It still worked but had to try harder.
* Too much noise (25): The network got confused and couldn’t learn properly.

A screenshot of a computer

AI-generated content may be incorrect.Observations: With noise at 25, the decision boundary was all over the place. The test loss was high, meaning the network wasn’t learning well.

Task 5:

I switched datasets to see how the network handled different patterns.

* Circle dataset: Easy to learn with enough neurons.
* Spiral dataset: Super hard. Even with four layers, it struggled.
* High noise with spiral dataset: Impossible. The network was completely lost.

A screenshot of a computer

AI-generated content may be incorrect.Observations: The test loss was high when I used the spiral dataset with noise. The decision boundary barely followed the shape of the spirals, showing that the network had trouble learning.

1. **What I Learned**

| **Change I Made** | **What Happened** | **Good or Bad?** |
| --- | --- | --- |
| **Used different activation functions** | ReLU worked best, Linear was useless | ✅ Good |
| **Changed neuron count** | More neurons helped, but too many might overfit | ✅ Good |
| **Adjusted learning rate** | Too low = slow, too high = unstable | ✅ Best at 0.03 |
| **Added noise** | A little noise helped, too much ruined learning | ✅ Balance is key |
| **Tried different datasets** | Simple ones were easy, spirals were tough | ✅ Some problems need more fine-tuning |

1. **Why This Matters**

This whole experiment taught me that small changes make a big difference when training a neural network. If I set up wrong, the network wouldn’t learn at all. If I balanced everything, it learned well.

* Picking the right activation function matters. ReLU is fast, but only if you have enough neurons.
* More neurons help, but too many can be a problem.
* The learning rate must be just right. Too slow and it takes forever, too fast and it never settles.
* Noise makes things harder. A little is fine, but too many ruins everything.
* Some datasets are way harder to learn than others. The spiral dataset needed way more tuning than the circle.

1. **Final Thoughts**

This was a cool experiment because I could see how each change affected the network. The hardest part was dealing with high noise and bad learning rates, but after some trial and error, I got better results. This experiment showed me how real world AI models need a lot of tuning to work well.

Work Cited:

1. TensorFlow Playground. "Playground: An Interactive Visualization of Neural Networks." playground.tensorflow.org, https://playground.tensorflow.org/.
2. Brownlee, Jason. "A Gentle Introduction to Neural Networks." Machine Learning Mastery, 2019, https://machinelearningmastery.com/neural-networks-crash-course/.
3. Chollet, François. "Neural Networks and Deep Learning." Keras Documentation, 2021, https://keras.io/guides/sequential\_model/.