**Christian Mpabuka**

A06 TensorFlow Playground Presentation

**ITAI 1378 Computer Vision**

**Professor: Patricia McManus**

**A06 TensorFlow Playground Presentation**

**TensorFlow Playground** is an interactive web-based tool created by Google to help users visually understand how neural networks work. It demonstrates how different aspects of a neural network, such as hidden layers, activation functions, learning rates, and data features, influence model training.

When designing neural networks, one of the most important decisions is determining the number of hidden layers. This decision significantly affects the model’s performance and its ability to learn and generalize from data. In this blog post, we will explore the concept of hidden layers, their importance, and how to determine the optimal number for your neural network.

An Activation Function decides whether a neuron should be activated or not, determining the neuron’s input to the network's importance in the process of prediction using simple mathematical operations. The role of the Activation Function is to derive output from a set of input values fed to a node (or a layer).

There are different types of activation functions:

- ReLU (Rectified Linear Unit): This operates very fast because it doesn’t saturate for large input values. It allows gradients to remain large enough to update the weights effectively. The ReLU function is non-linear, which allows the network to learn complex data patterns. Its output is not bounded, avoiding the gradient saturation problem found in other functions like sigmoid or tanh.

- Sigmoid: The network learns slowly compared to ReLU, and the gradient values tend to become small.

Neurons are the basic computational units in a neural network. Each neuron receives input, processes it using a weight and bias, applies an activation function, and outputs a signal tocomplex relationships**.**

Data Noise: In this task, I introduce noise into the dataset using the "Noise" slider in TensorFlow Playground and analyze how it affects the network's performance, particularly its ability to generalize. Understanding the influence of noise on neural networks is crucial, as real-world data often contains various types of noise that can challenge the learning process.A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**Key Learnings**

**Activation Functions**: Experimenting with different activation functions revealed their importance in determining the network’s ability to learn complex patterns. The choice of function directly influenced convergence speed and the ability to capture non-linear relationships.

- ReLU was highly effective for faster learning but introduced some instability with noisy data due to dead neurons.

- Sigmoid and tanh were slower to converge and had issues with vanishing gradients, particularly in deeper networks.

**Hidden Layers and Neurons**: Modifying the number of neurons and hidden layers demonstrated how increasing network complexity can improve performance on complex tasks but also increases the risk of overfitting.

Finding the right balance between model complexity and dataset characteristics was crucial to avoid overfitting, particularly with noisy data.

**Learning Rate:** Adjusting the learning rate was eye-opening in terms of how it can impact both convergence speed and stability. A learning rate that was too high caused the network to diverge, while a low rate resulted in slow learning.

The importance of learning rate schedules or adaptive learning rates became clear as a solution to help optimize performance.

**Data Noise:** Introducing noise into the dataset highlighted the network's vulnerability to overfitting when faced with imperfect data. It also demonstrated how noise affects decision boundaries, generalization, and ultimately performance.

**Conclusion**

This hands-on exploration of neural networks using TensorFlow Playground has deepened my understanding of how key factors—such as activation functions, the number of hidden layers and neurons, learning rates, and data noise—affect the behavior and performance of a neural network. The experiments provided invaluable insights into both the theoretical and practical aspects of deep learning.

**Key Learnings**

**Activation Functions**: Experimenting with different activation functions revealed their importance in determining the network’s ability to learn complex patterns. The choice of function directly influenced convergence speed and the ability to capture non-linear relationships.

* + **ReLU** was highly effective for faster learning but introduced some instability with noisy data due to dead neurons.
  + **Sigmoid** and **tanh** were slower to converge and had issues with vanishing gradients, particularly in deeper networks.

**Hidden Layers and Neurons**: Modifying the number of neurons and hidden layers demonstrated how increasing network complexity can improve performance on complex tasks but also increases the risk of overfitting.

* + Finding the right balance between model complexity and dataset characteristics was crucial to avoid overfitting, particularly with noisy data.

**Learning Rate**: Adjusting the learning rate was eye-opening in terms of how it can impact both convergence speed and stability. A learning rate that was too high caused the network to diverge, while a low rate resulted in slow learning.

* + The importance of **learning rate schedules** or **adaptive learning rates** became clear as a solution to help optimize performance.

**Data Noise**: Introducing noise into the dataset highlighted the network's vulnerability to overfitting when faced with imperfect data. It also demonstrated how noise affects decision boundaries, generalization, and ultimately performance.

* + Regularization techniques like **dropout** and **L2 regularization** emerged as essential tools to prevent overfitting, especially in noisy environments.

**Challenges Faced and Overcoming Them**

Dealing with Overfitting in Complex Networks: One of the main obstacles I faced was finding the right balance in the model's complexity to prevent overfitting, especially when dealing with noisy data. To tackle this, I implemented regularization techniques such as dropout and simplified the architecture to enhance generalization.

Selecting the Learning Rate: Another challenge I encountered was determining the optimal learning rate. Initially, I struggled to set the right rate, which resulted in slow training or caused the model to diverge. Through experimentation with different rates and gradually increasing the learning rate, I was able to find a stable configuration for the network.

Visualizing Decision Boundaries with Noise: The introduction of noise caused the decision boundaries to become erratic, posing a challenge in understanding the underlying reasons. This prompted me to delve deeper into the concepts of overfitting and the role of regularization, which ultimately enabled me to interpret these visualizations more effectively.

**Conclusion**:

Throughout this lab, I gained a practical understanding of how neural networks learn, how they handle noisy data, and how different hyperparameters affect their performance. It reinforced the importance of balancing model complexity, regularization, and data quality to achieve optimal results. Additionally, I learned how sensitive neural networks can be to hyperparameters such as the learning rate, and how regularization is essential when working with noisy datasets.

Overall, this project not only solidified theoretical concepts but also provided practical skills in tuning neural networks to improve performance in real-world applications.

.

**Citations**:

. Sanjay Dutta Jun 5, 2024, [Understanding the Number of Hidden Layers in Neural Networks: A Comprehensive Guide | by Sanjay Dutta | Medium](https://medium.com/@sanjay_dutta/understanding-the-number-of-hidden-layers-in-neural-networks-a-comprehensive-guide-0c3bc8a5dc5d)

.Pragati Baheti [Activation Functions in Neural Networks [12 Types & Use Cases] (v7labs.com)](https://www.v7labs.com/blog/neural-networks-activation-functions)