**Title:** **Exploring Neural Networks with TensorFlow Playground**

**Author:** Ruben Valenzuela

**1. Introduction**

Neural networks are a fundamental component of deep learning, designed to mimic the functionality of the human brain. They consist of input layers, hidden layers, and output layers, with neurons acting as processing units. Neural networks learn from data by adjusting weights through training iterations. The **TensorFlow Playground** provides an interactive platform for understanding how various parameters influence the network’s learning and performance.

This report documents a series of experiments conducted using TensorFlow Playground, focusing on **activation functions, hidden layers, learning rate, data noise, and dataset variations.**

**2. Tasks and Observations**

**Task 1 - Activation Functions**

**Objective:** To analyze the impact of different activation functions on the network’s performance.

**Experiment:**

* A **single hidden layer** was used.
* Different activation functions were tested, including **ReLU, Sigmoid, Tanh, and Linear**.

**Observations:**

* **ReLU**: Performed well with fast convergence but struggled with negative values (dying ReLU problem).
* **Sigmoid**: Smoothed outputs but suffered from vanishing gradients.
* **Tanh**: Similar to sigmoid but centered around zero, leading to better learning.
* **Linear**: Did not capture complex patterns effectively.

A screenshot of a computer

AI-generated content may be incorrect.

*Conclusion:* ReLU generally performs best for deep networks, while Tanh is preferable when centered values help convergence.

**Task 2 - Hidden Layer Neurons**

**Objective:** To understand how increasing or decreasing the number of neurons affects network performance.

**Experiment:**

* Started with **2 neurons**, then increased to **4, 6, and 8 neurons** in a **single hidden layer**.

**Observations:**

* **Few neurons (2-4)**: Struggled to capture complex patterns, leading to underfitting.
* **Optimal range (6-8 neurons)**: Balanced performance with good generalization.

**A screenshot of a computer

AI-generated content may be incorrect.**

*Conclusion:* The number of neurons should be balanced to ensure proper generalization without overfitting.

**Task 3 - Learning Rate**

**Objective:** To observe how learning rate affects training speed and accuracy.

**Experiment:**

* Used learning rates of **0.001, 0.01, 0.03, and 0.1**.

**Observations:**

* **Too low (0.001)**: Slow convergence, requiring many iterations.
* **Moderate (0.01-0.03)**: Balanced learning speed and accuracy.
* **Too high (0.1)**: Caused erratic behavior, preventing convergence.

A screenshot of a computer

AI-generated content may be incorrect.

*Conclusion:* The ideal learning rate balances training speed and stability. 0.01 - 0.03 performed best.

**Task 4 - Data Noise**

**Objective:** To analyze the impact of noisy data on neural network performance.

**Experiment:**

* Added increasing levels of noise using the **"Noise" slider** in TensorFlow Playground.

**Observations:**

* **No noise (0%)**: The model learned patterns effectively.
* **Low noise (10-20%)**: Minor impact, still able to generalize well.
* **High noise (30%+)**: Training became unstable, accuracy dropped due to difficulty in identifying patterns.
* A screenshot of a computer

  AI-generated content may be incorrect.

*Conclusion:* Low noise improves robustness, but excessive noise disrupts training and reduces accuracy.

**Task 5 - Dataset Exploration**

**Objective:** To test how different datasets influence neural network performance.

**Experiment:**

* Tried **circle, spiral, and plane datasets**.

**Observations:**

* **Circle dataset**: Required more neurons and hidden layers due to complex decision boundaries.
* **Spiral dataset**: Difficult to learn without multiple hidden layers.
* **Plane dataset**: Easy to classify, even with fewer neurons.
* A screenshot of a computer

  AI-generated content may be incorrect.

*Conclusion:* Complex datasets require deeper networks with fine-tuned parameters.

**3. Discussion and Practical Implications**

Understanding neural network behavior through **activation functions, hidden layers, learning rates, noise, and dataset complexity** is crucial for designing efficient models. The findings suggest:

* **ReLU activation** is preferable for deep learning.
* **A balanced number of neurons** prevents underfitting and overfitting.
* **Moderate learning rates (0.01-0.03)** offer the best convergence.
* **Minimizing noise** improves accuracy.
* **Complex datasets** need deeper architectures for feature extraction.

These insights are directly applicable in areas such as image classification, speech recognition, and recommendation systems.

**4. Conclusion**

Through this experiment, I gained hands-on experience with neural network architectures and their optimization. The ability to **tune parameters effectively** is essential in building **robust AI models**. The practical knowledge acquired through TensorFlow Playground demonstrates how different configurations impact a model’s accuracy and generalization ability.

**Future Work:** Exploring **dropout layers and regularization** techniques can further enhance model performance.

**5. References**

* Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
* TensorFlow Playground: [https://playground.tensorflow.org](https://playground.tensorflow.org/)
* Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
* Online AI Research Papers and Class Slides.