**EJERCICIO DE RECONOCIMIENTO – SEBASTIAN BELALCAZAR MOSQUERA**

**Introducción:**

Este documento presenta una introducción práctica al campo de las Redes Neuronales (RN) utilizando la plataforma TensorFlow Playground. La plataforma permite visualizar de manera interactiva los componentes de una RN y su impacto en su funcionamiento.

**Componentes de una Red Neuronal:**

* **Conjunto de datos:** Se selecciona el conjunto de datos a utilizar para el entrenamiento y la prueba de la RN.
* **División de datos:** Se establece la proporción del conjunto de datos que se utilizará para el entrenamiento (train) y la evaluación (test).
* **Ruido:** Se introduce ruido artificial a los datos para simular condiciones del mundo real.
* **Tasa de aprendizaje:** Se define la velocidad a la que la RN ajusta sus parámetros durante el entrenamiento.
* **Función de activación:** Se selecciona la función matemática que determina la salida de cada neurona.
* **Regularización:** Se aplica una técnica para evitar el sobreajuste del modelo.
* **Tipo de problema:** Se especifica si la RN se utiliza para clasificación (agrupar datos) o regresión (predecir valores continuos).
* **Arquitectura de la red:** Se define el número de capas ocultas y el número de neuronas en cada capa.

**Resultados del entrenamiento:**

* **Épocas:** Número de iteraciones realizadas durante el entrenamiento.
* **Error:** Gráfica que muestra el error de entrenamiento (train) y evaluación (test) a lo largo de las épocas.
* **Contribución de las neuronas:** Visualización del aporte de cada neurona al modelo.
* **Predicciones:** Representación gráfica de las predicciones de la RN sobre el conjunto de datos (grupos para clasificación o tendencia para regresión).

**A screenshot of a computer

Description automatically generated**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Análisis de la regularización:**

El documento presenta ejercicios prácticos que varían el tipo de problema (regresión vs. clasificación) y el tipo de regularización (sin regularización, L1 y L2) para observar el impacto de estas configuraciones en el rendimiento del modelo.

**Conclusión:**

El documento ofrece una introducción práctica y visual al aprendizaje de Redes Neuronales utilizando TensorFlow Playground, permitiendo comprender los componentes clave de una RN y su funcionamiento a través de la experimentación con diferentes parámetros.

**CLASIFICACIÓN**

Podemos observar como, en este tipo de problema tanto no realizar Regularización, como aplicar la regularización L2 impacta de manera positiva el modelo ya que se consigue llegar a los grupos dentro del dataset de una manera más fácil y en un rango de tiempo moderado (épocas).

Por otra parte al usar L1, se observa como en principio parece dar buen resultado y sin embargo, al pasar las épocas el modelo ya no puede disminuir la tasa de error y se estanca y no logra terminar de definir bien los grupos.

Lo anterior se puede deber a que la regularización L1 es un método más fuerte que anula los pesos de las neuronas que considera no están aportando y por tanto, la red no tiene la información suficiente para encontrar el resultado esperado.

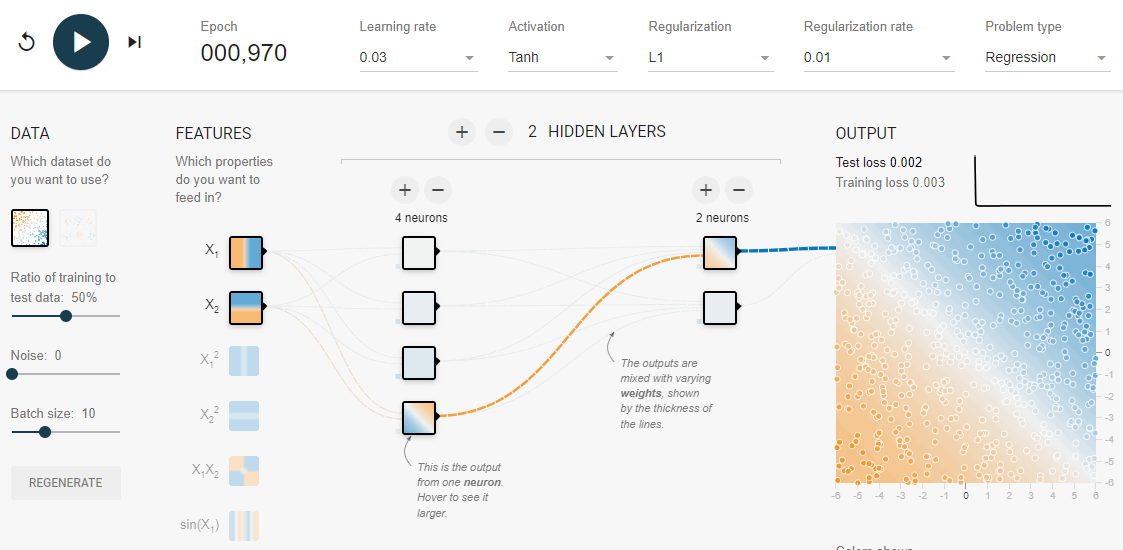
**REGRESIÓN**

En este tipo de problema, podemos observar como en los tres casos, tanto no realizar regularización, como realizar L1 y L2 dan muy buenos resultados, ya que llegan a la regresión esperada con un error bastante pequeño tanto en test como en train y además lo hacen en muy pocas épocas.

Es importante resaltar aquí que la red funciona de mejor manera sin ningún tipo de regularización, como se observa en la primera imagen, la red alcanzó un error de test y train más bajo que los alcanzados con L1 y L2. Esto podría deberse al tipo de problema, ya que al ser “sencillo” las regularizaciones solo quitan información y como se mencionó anteriormente esto entorpece su funcionamiento.

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated