

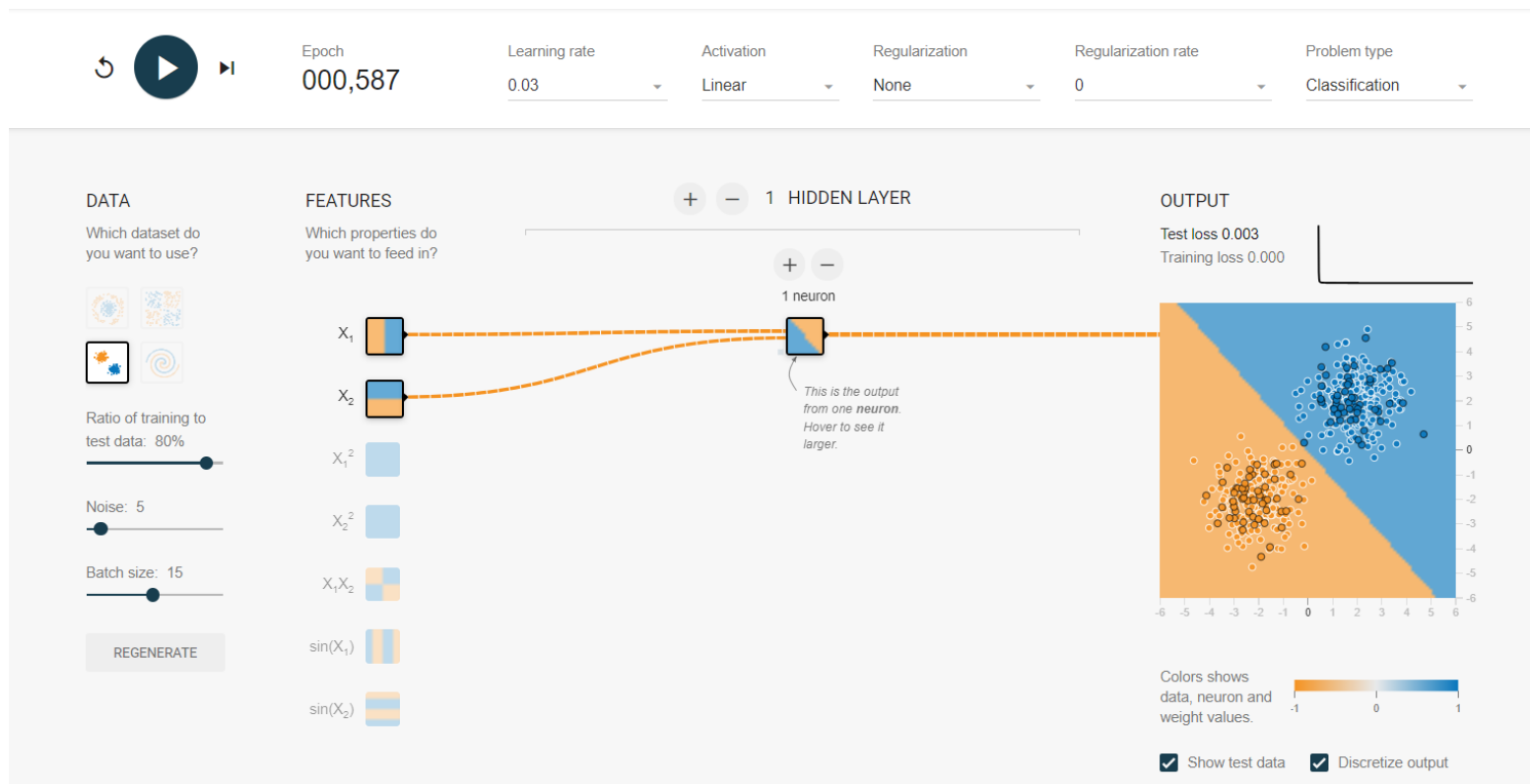
TensorFlow Playground Es una aplicación web desarrollada por Google que permite experimentar con redes neuronales de manera interactiva. Algunas características destacadas son:

1. **Interactividad:** Permite ajustar parámetros de una red neuronal (como la cantidad de capas y neuronas, la función de activación, etc.) y ver los efectos en tiempo real en la clasificación de datos.
2. **Visualización:** Proporciona visualizaciones gráficas dinámicas que ayudan a entender cómo una red neuronal aprende y cómo se separan los datos en diferentes clases.
3. **Datasets Predefinidos:** Viene con varios conjuntos de datos predefinidos (como círculos, espirales, XOR) que se pueden usar para experimentar con diferentes configuraciones de red.
4. **Entrenamiento Rápido:** Utiliza TensorFlow.js para ejecutar el entrenamiento directamente en el navegador, lo que permite ajustar parámetros y ver los resultados de manera instantánea.

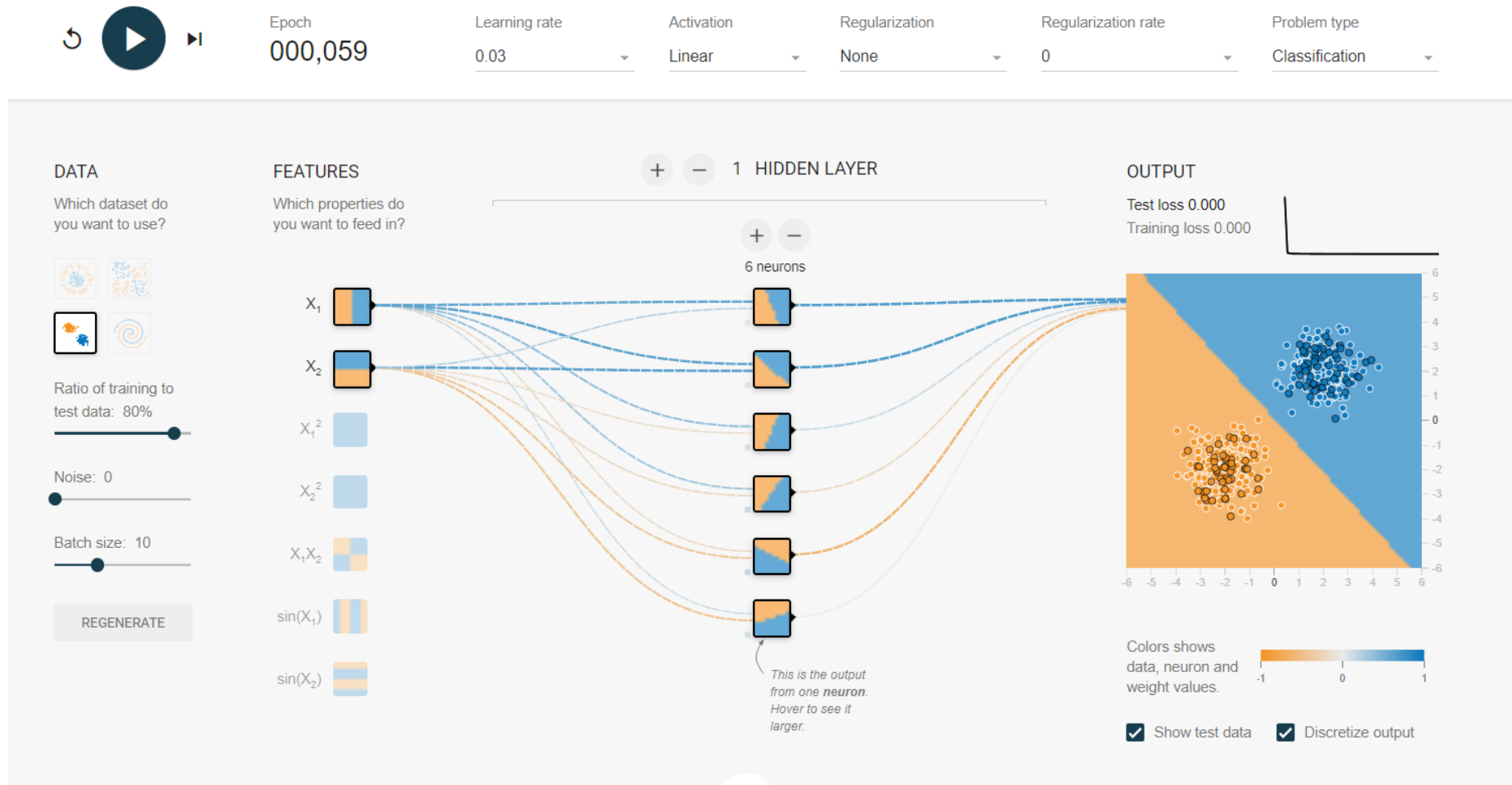
CLASIFICACION DE DATOS POR TENSORFLOW.

En la parte izquierda, selecciona el conjunto de datos desde la lista de conjuntos de datos predefinidos. Esto cargará automáticamente los puntos que forman una figura con dos conjuntos de datos.

En este caso ajustamos los factores a una sola capa de red neuronal con una sola neuronal y vemos que esto es suficiente para clasificar los datos por medio de regresión lineal en tan solo 587 Epoch y con un entrenamiento de 80% y un ruido de 5. La tasa de error es de 0.003

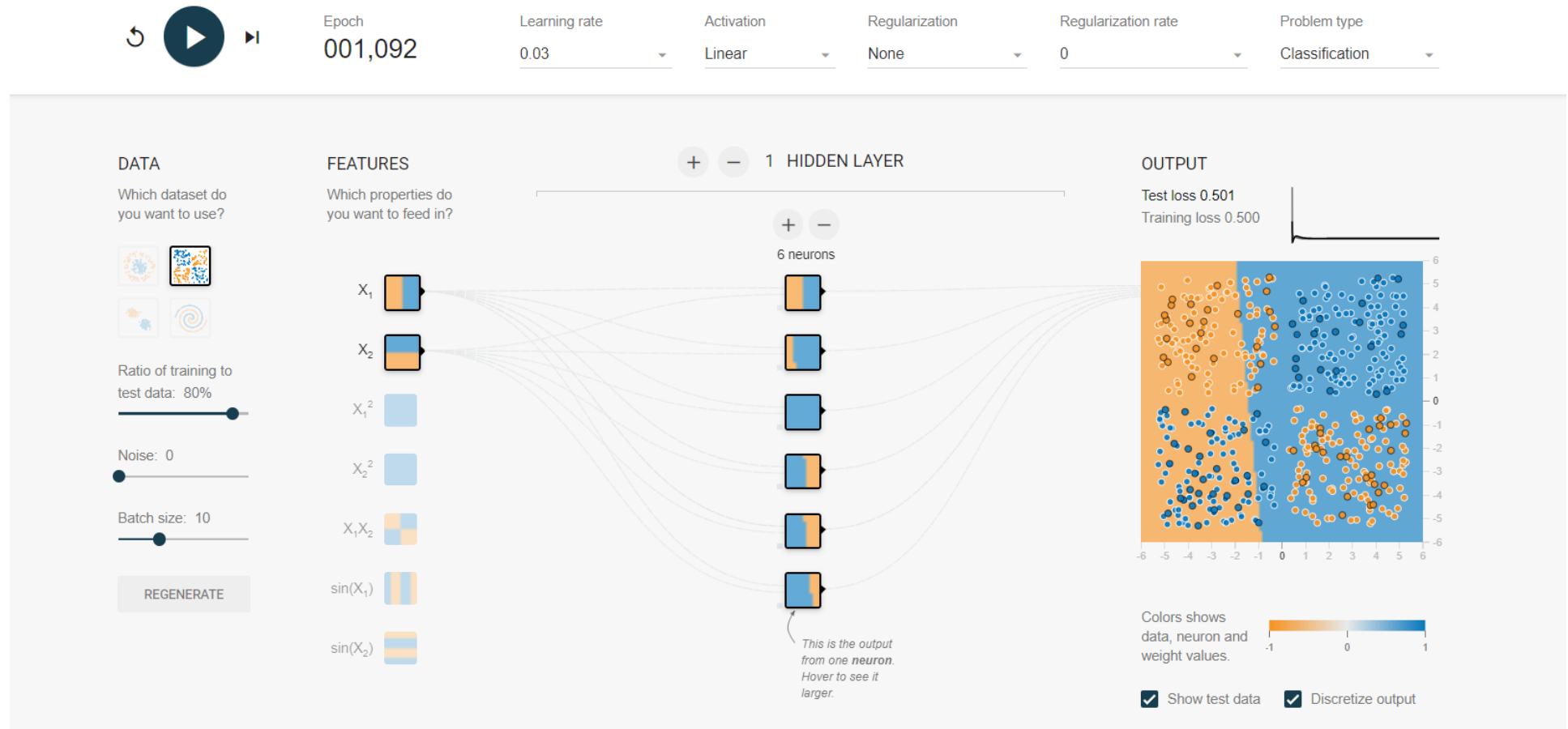


Al aumentar a 6 el numero de neuronas se puede ver que la tasa de error es de 0.00 y la cantidad de tiempo en epoch también disminuye a un 0.59 pero esto puede demostrar también un sobreentrenamiento en el modelo.

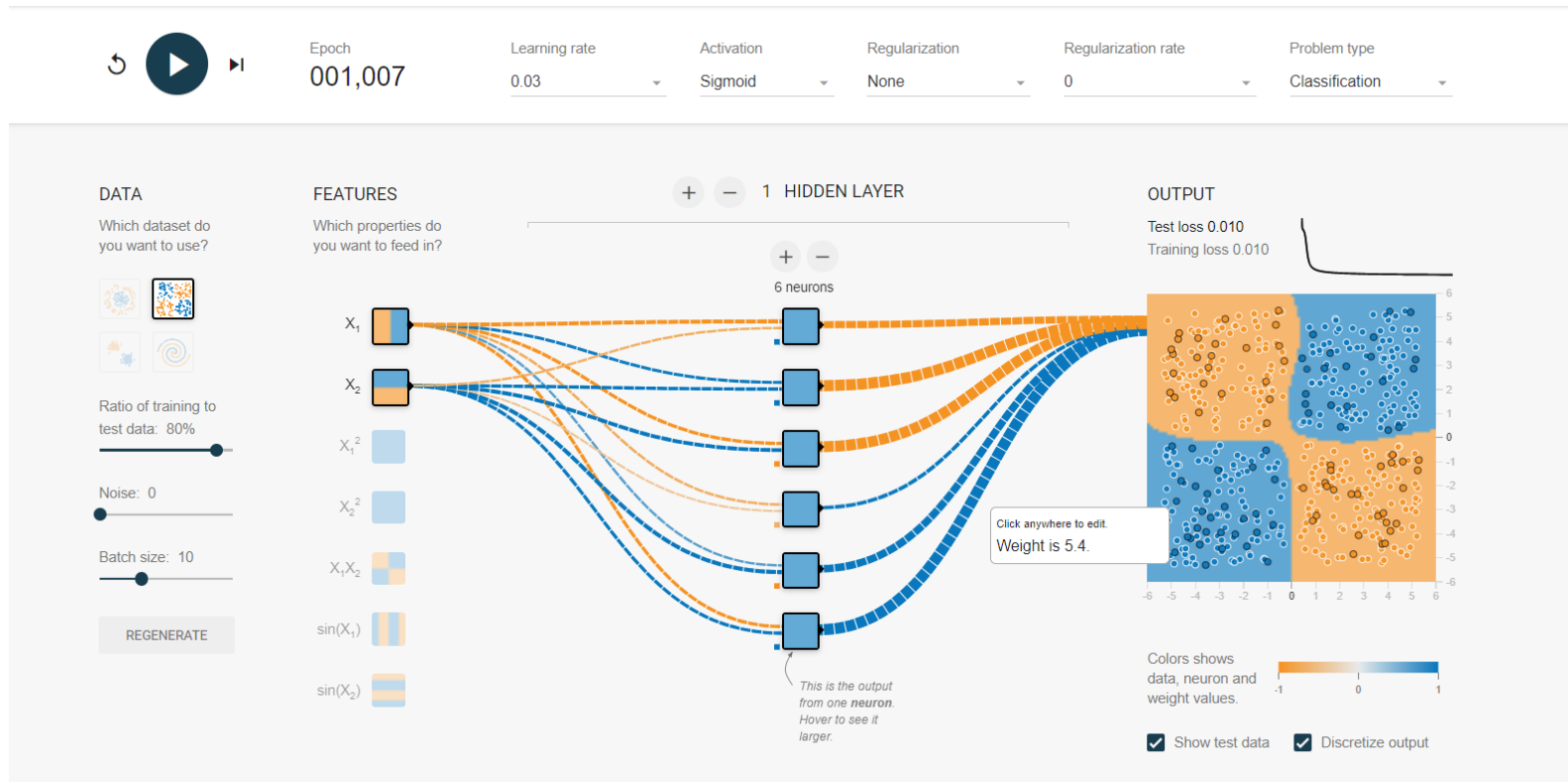


Al cambiar el modelo de datos a "XOR" desde la lista de conjuntos de datos predefinidos. Se carga una figura en forma de X en el espacio.

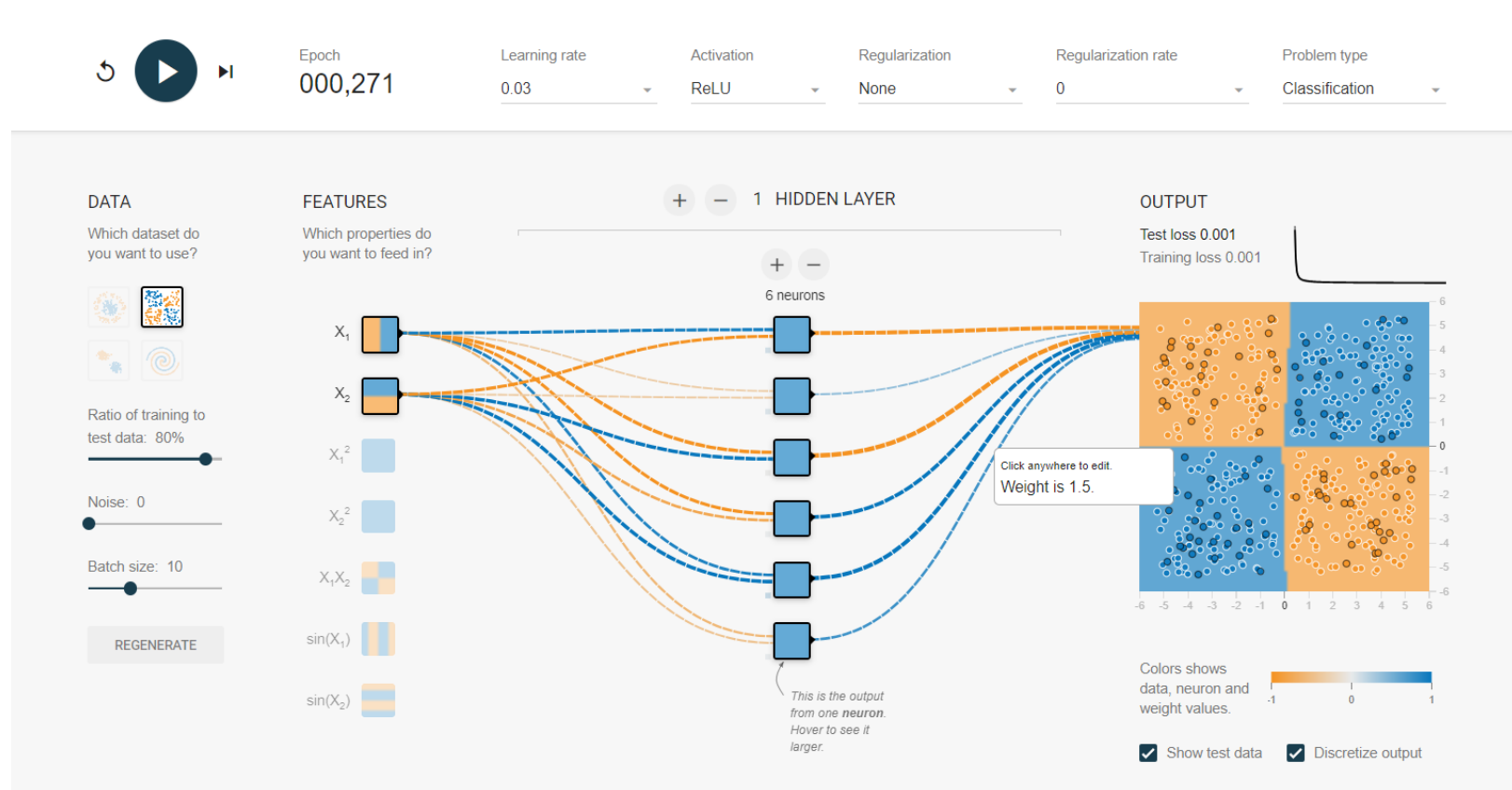
Sin variar los parámetros del **modelo lineal** y manteniendo el modelo con un entrenamiento del 80%, y una red de capa de neuronas a 6 neuronas y una activación lineal, se obtiene una **taza de error del 0.501** en un tiempo de epoch de 1.092. en la imagen se ve como claramente este modelo de red neuronal no se ajusta lo suficiente para clasificar este tipo de datos.



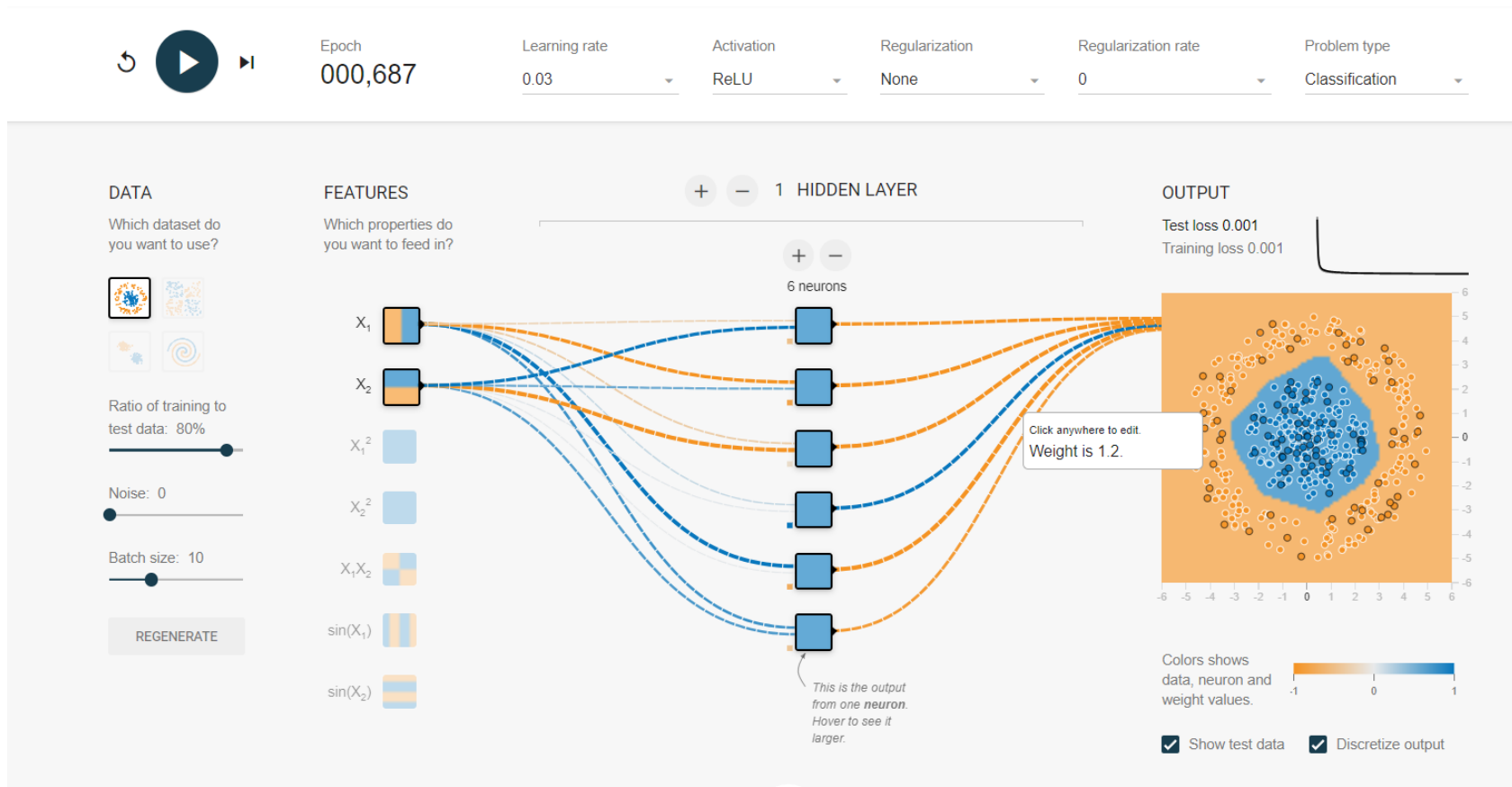
Al modificar el **modelo de activación a sigmoid** y mantener los otros parámetros sin variar se puede ver una mejora en el ajuste del modelo disminuyendo **la tasa de error al 0.010** y **el training los en un 0.010**, en un tiempo aproximado de 1.000 Epoch, adicional se puede ver un trafico pesado con un peso aproximado de 5.4.



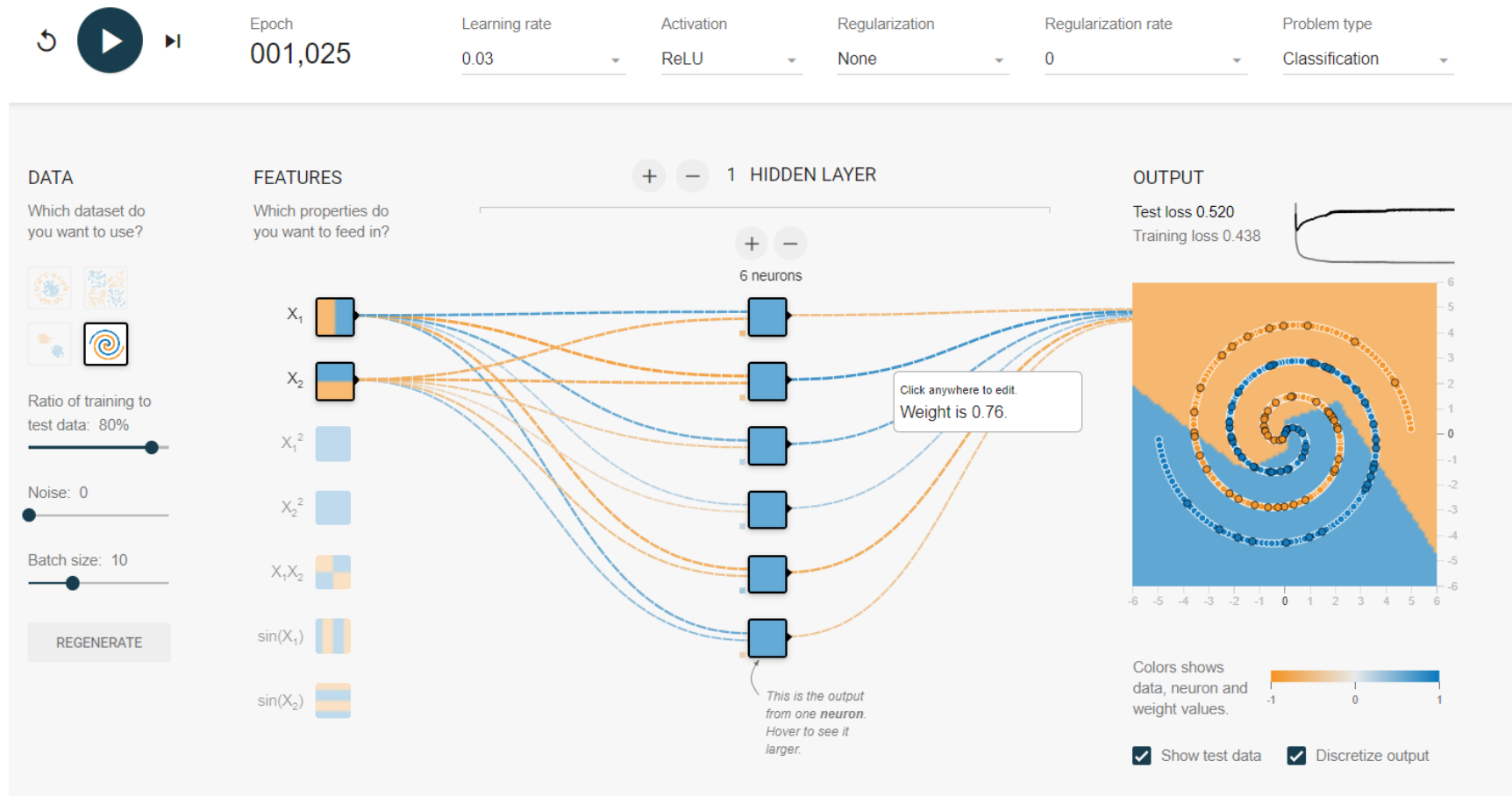
Aunque esta tasa de error y de training no es mala, se puede observar una clara mejoría en el análisis de este tipo de datos variando el **modelo de activación a ReLU**. Ya que la **tasa de error y de training** se disminuye al **0.001** en un tiempo equivalente a **271 Epoch** y se ve un peso mas ligero en las salidas de aproximadamente 1.5.



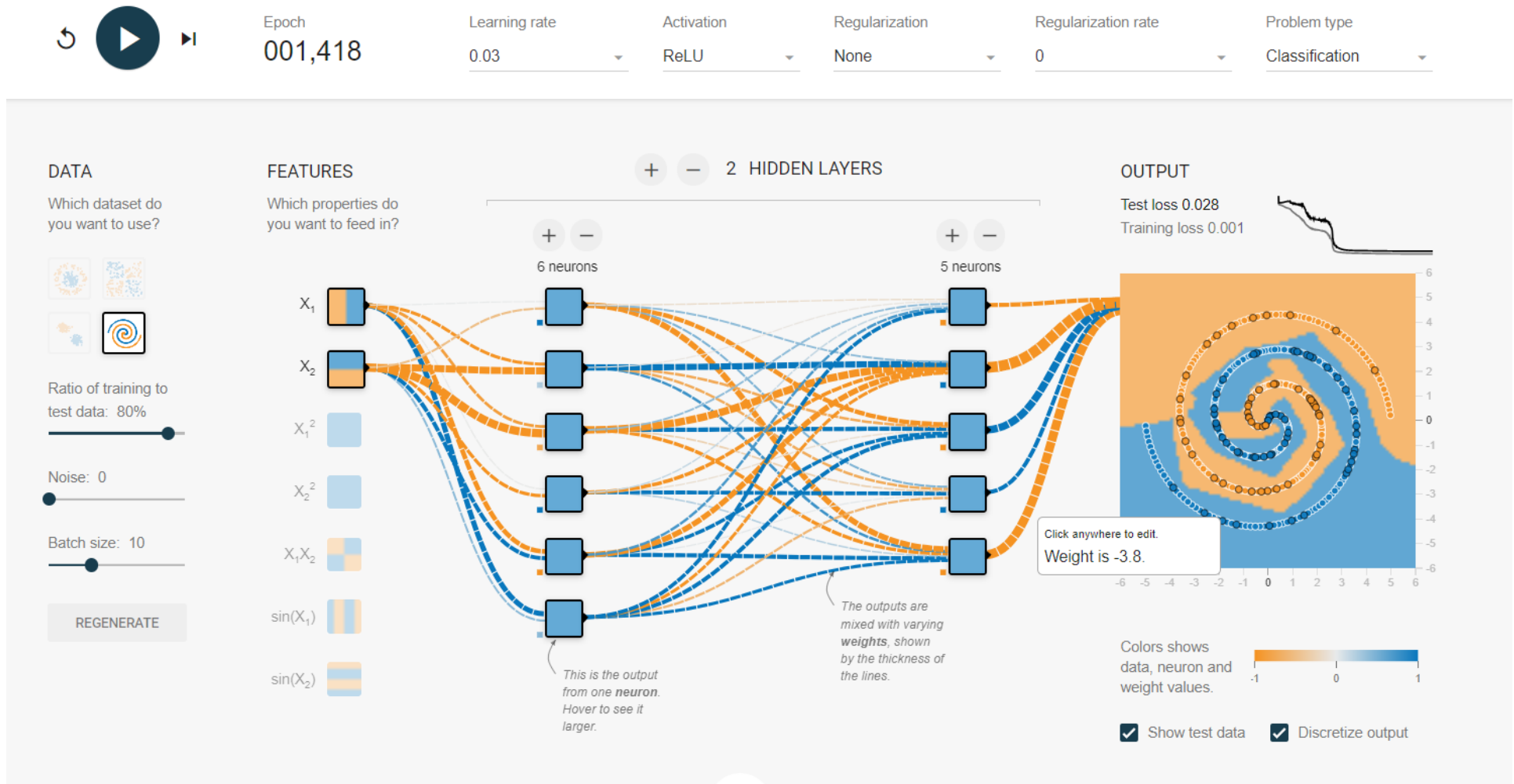
Al cambiar el modelo de dos círculos concéntricos desde la lista de conjuntos de datos predefinidos. Se carga una figura en forma de dos círculos concéntricos. Sin variar los parámetros del **ReLU** y manteniendo el modelo con un entrenamiento del 80%, y una red de capa de neuronas a 6 neuronas y una activación ReLU, se obtiene una **taza de error del 0.01** en un tiempo de Epoch de 687. en la imagen se ve como claramente este modelo de red neuronal se ajusta bastante bien para clasificar este tipo de datos, aunque aumente un poco el tiempo de épocas.



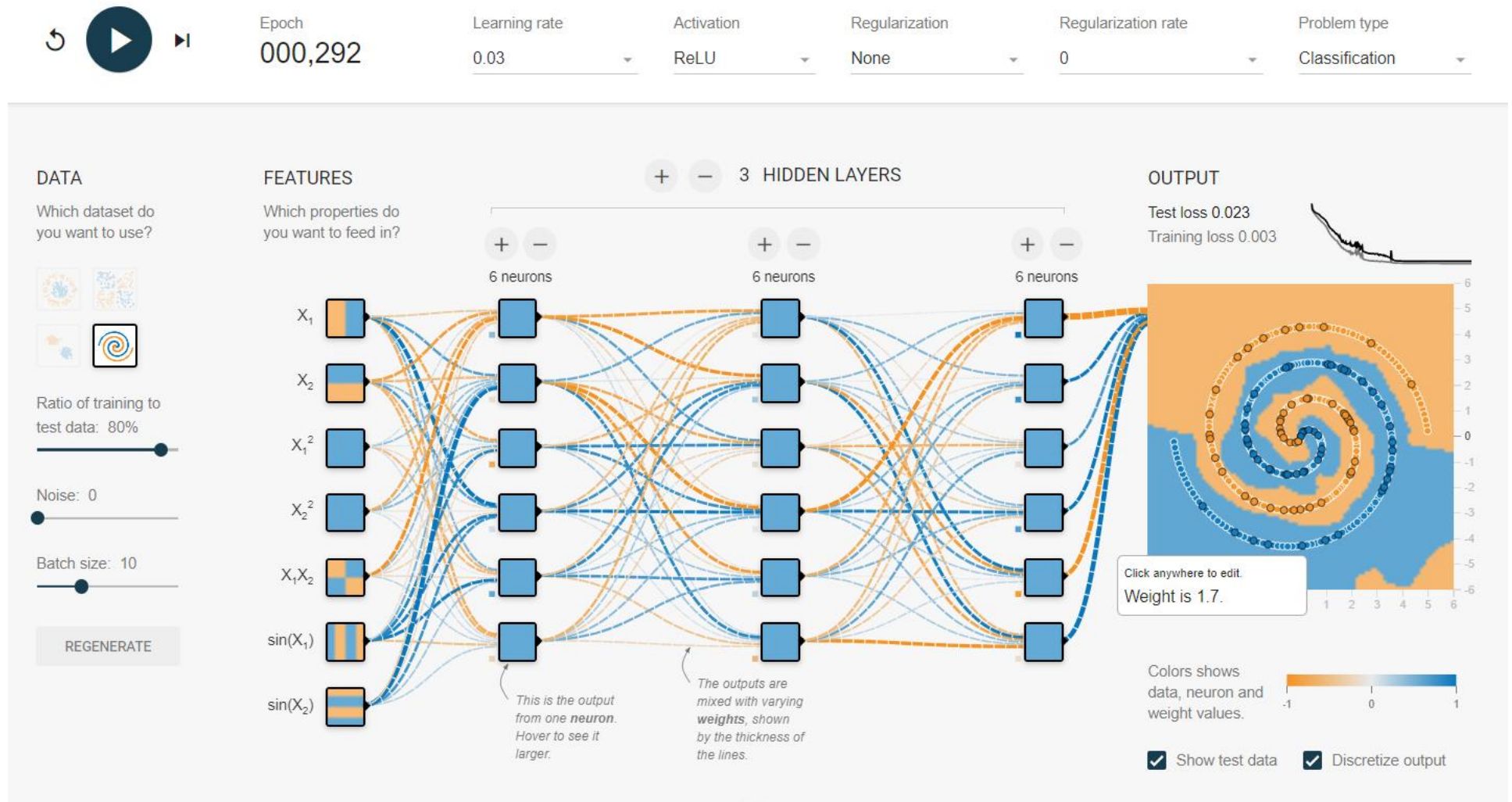
Al cambiar el modelo a datos en Espiral desde la lista de conjuntos de datos predefinidos. Se carga una figura en forma de espirales entrelazadas. Sin variar los parámetros del **ReLU** y manteniendo el modelo con un entrenamiento del 80%, y una red de capa de neuronas a 6 neuronas y una activación ReLU, se obtiene una **taza de error del 0.520 y un training loss de 0.438** en un tiempo de Epoch de 1025. en la imagen se ve como claramente este modelo de red neuronal no se ajusta bastante bien para clasificar este tipo de datos.



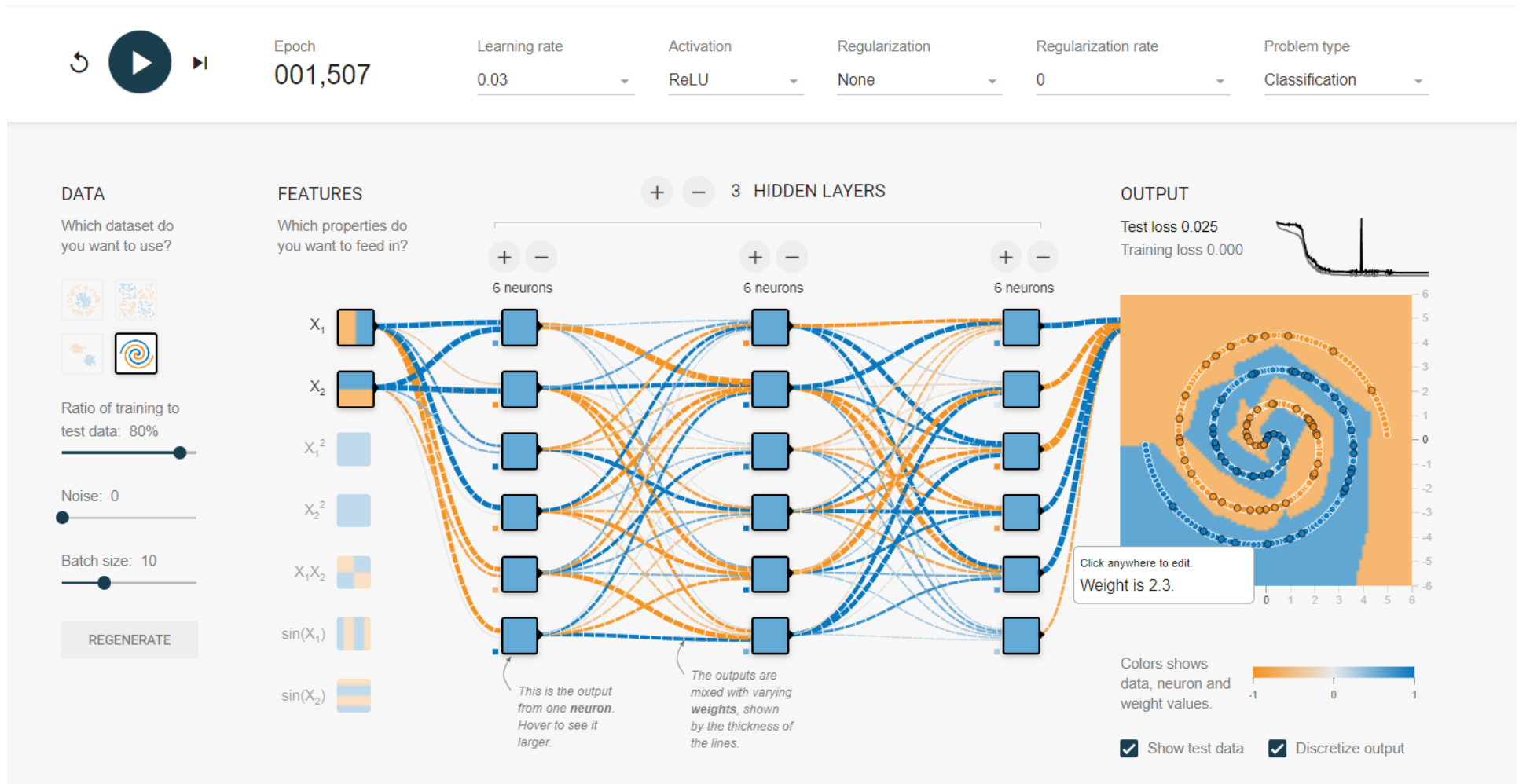
Al modificar otros parámetros como el aumento de capas se puede ver una mejora en el ajuste del modelo disminuyendo la **taza de error al 0.028 y el training loss en un 0.001**, en un tiempo aproximado de 1.418 Epoch, adicional se puede ver un trafico pesado con un peso aproximado de 3.8.



Aunque se varia la cantidad de entradas y se disminuye la cantidad de Epoch a 292 la taza de error no disminuye mucho ya que esta queda en 0.023.



Dado que la tasa de error es alta se decide colocar otra capa de neuronas pero aun así la tasa de error no disminuye mucho dado que esta sigue estando en 0.025, se ve una leve mejoría en el training loss ya que disminuye a 0.00 pero el tiempo transcurrido es de 1.507 Epoch. Lo cual es elevado y demuestra que el modelo no se ajusta de forma adecuada para resolver te tipo de datos.



para lograr un mejor ajuste en el modelo con este tipo de datos en espiral se aumentaron a 7 capas de 7 neuronas cada uno y se aumentaron la cantidad de input que recibió el modelo en sus inicio con este logramos disminuir el test loss a 0.000 y el training loss a 0.001, utilizando el modelo de activación de ReLU y en un tiempo de 238 Epoch.



