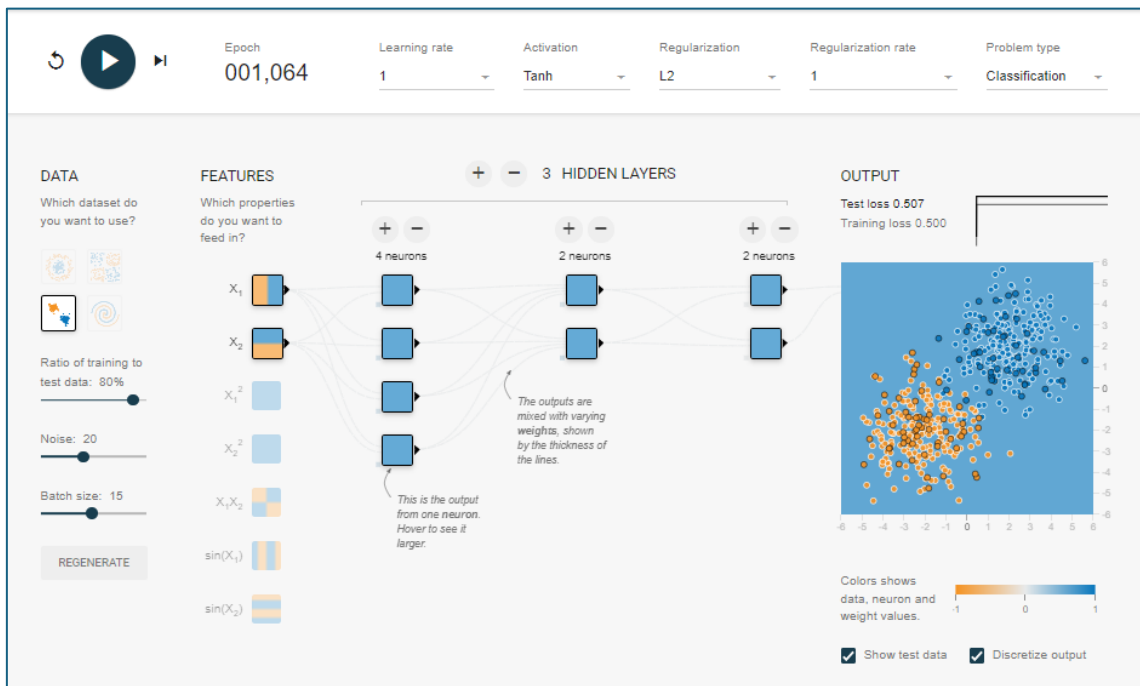
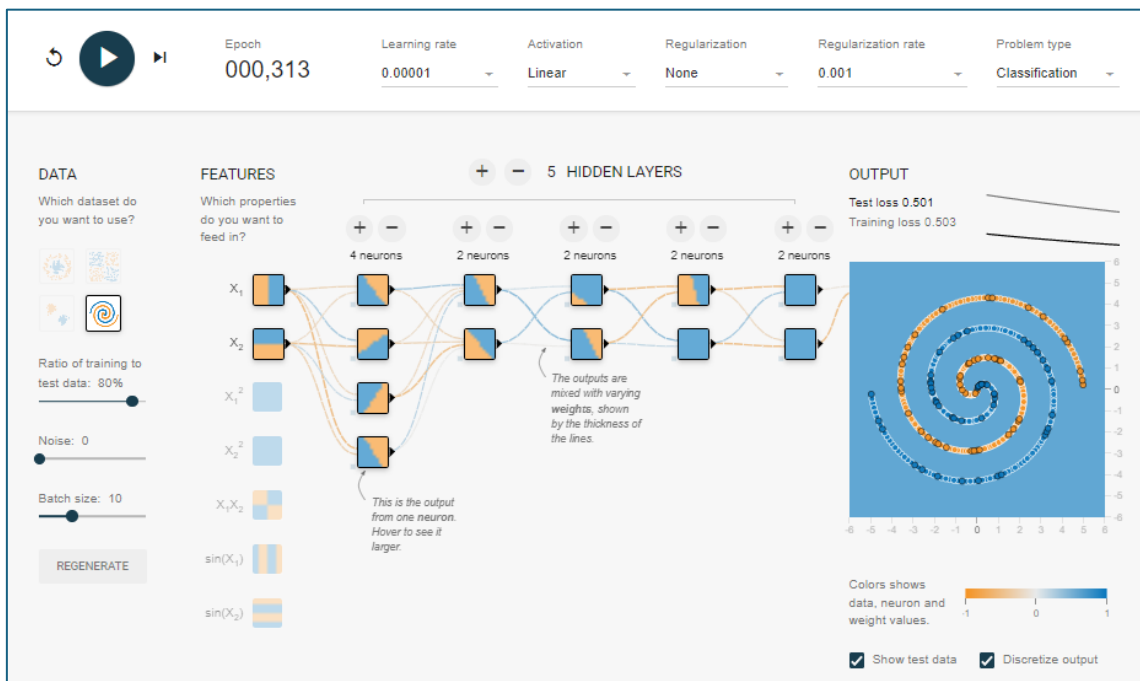


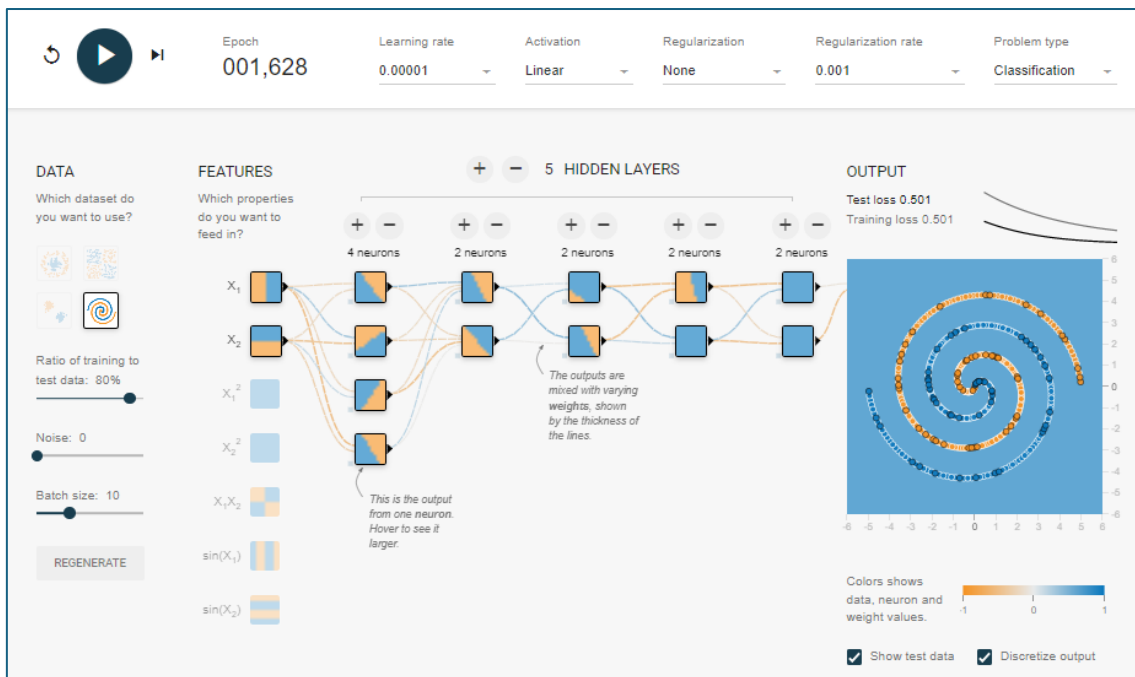
- Procedemos ajustando nuevos parámetros y modificando la cantidad de neuronas:



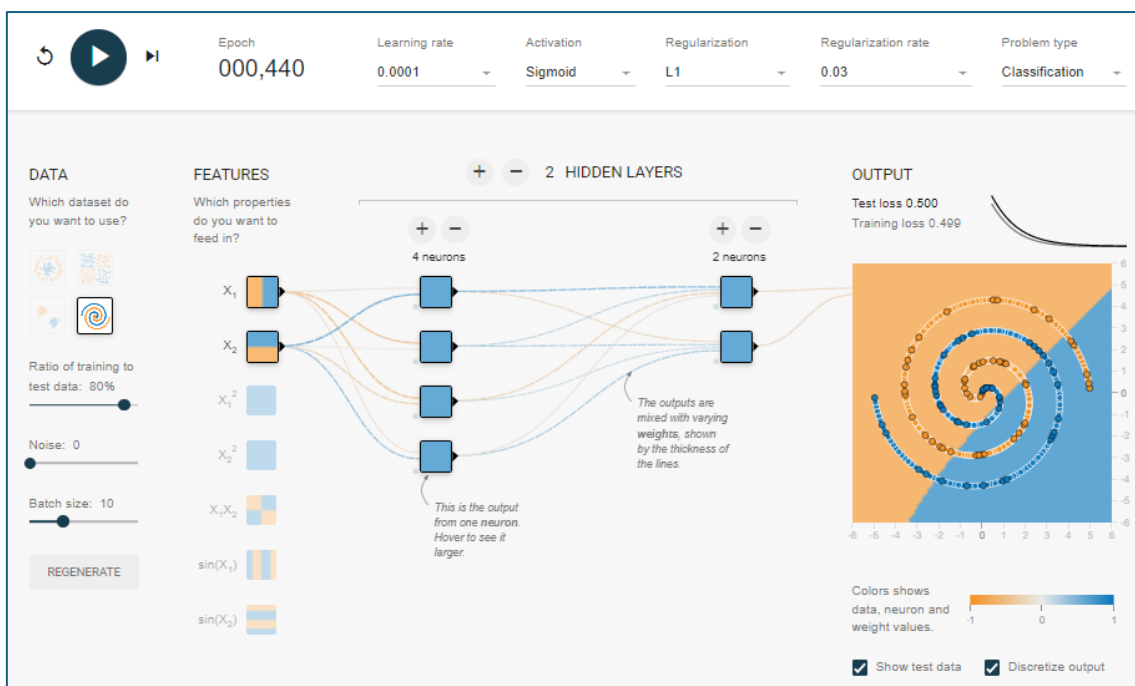
- Luego, mantenemos los parámetros, pero modificamos la cantidad de neuronas de nuestra red neuronal.



Observamos que a medida que la red se va entrenando, en el output las líneas se van acercando cada vez más.



- Para ver qué más ocurre, cambiamos los parámetros, pero dejamos la cantidad de neuronas:

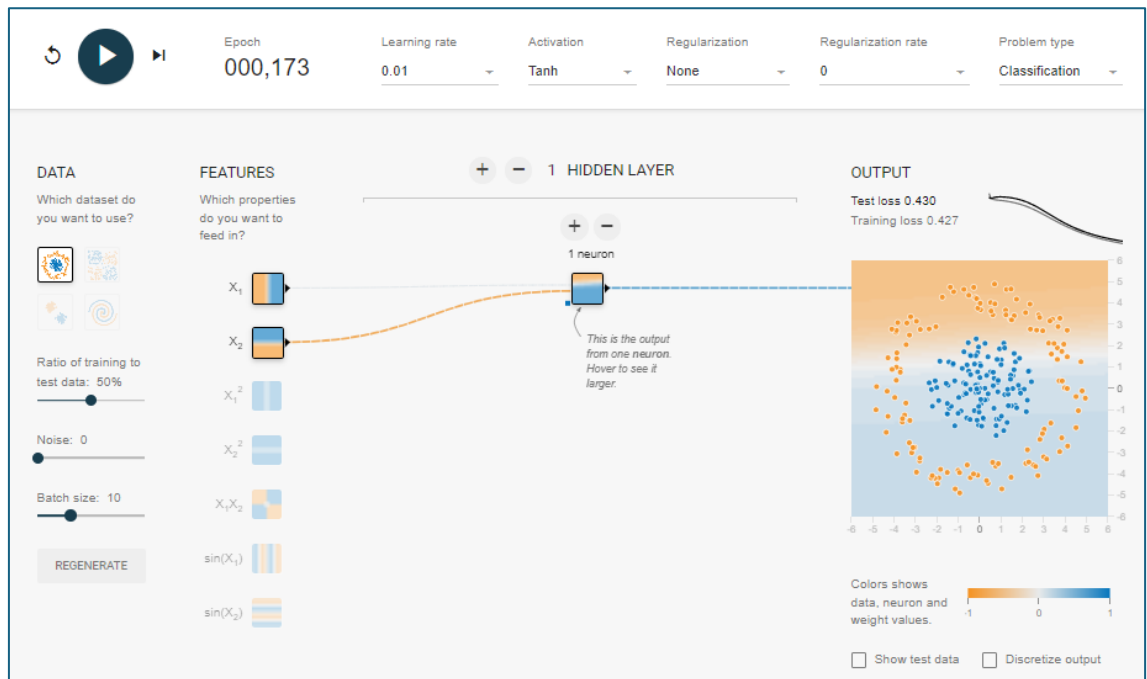


IMPLEMENTACIÓN Y EXPERIMENTACIÓN PERCEPTRÓN Y FUNCIONES DE ACTIVACIÓN

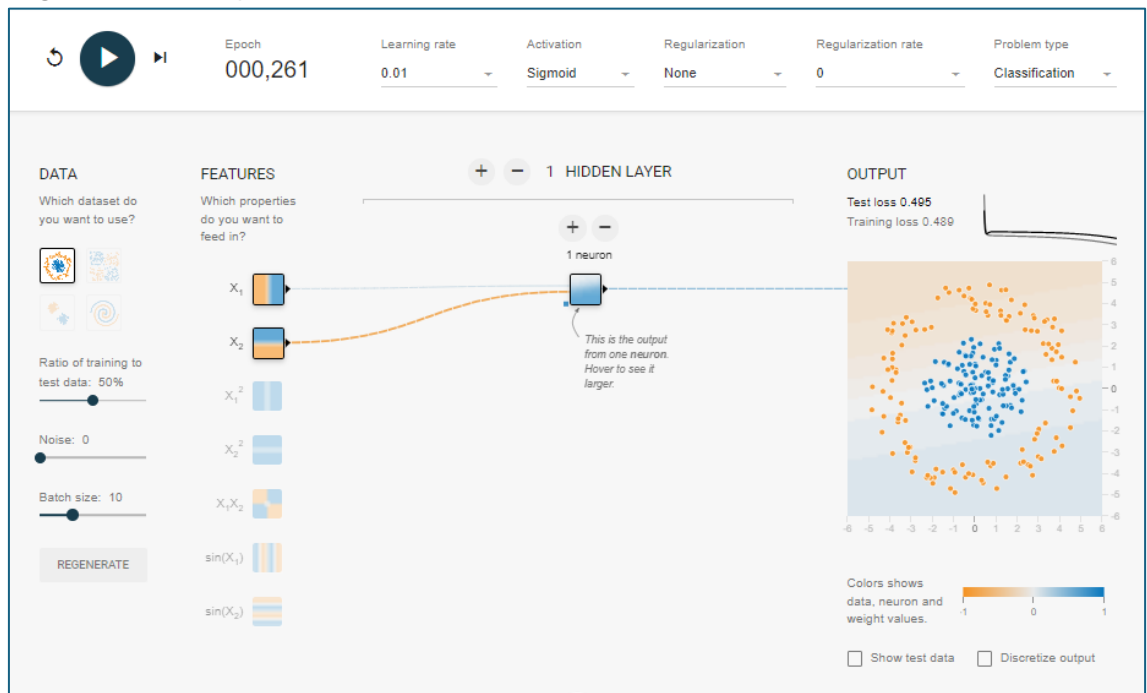
- Sabemos que un perceptrón es un algoritmo de clasificación binaria inspirado en el funcionamiento de las neuronas del cerebro humano. Este algoritmo toma varias entradas binarias y les aplica pesos, suma dichos productos de las

entradas y pesos para finalmente aplicar una función de activación que produce una salida binaria.

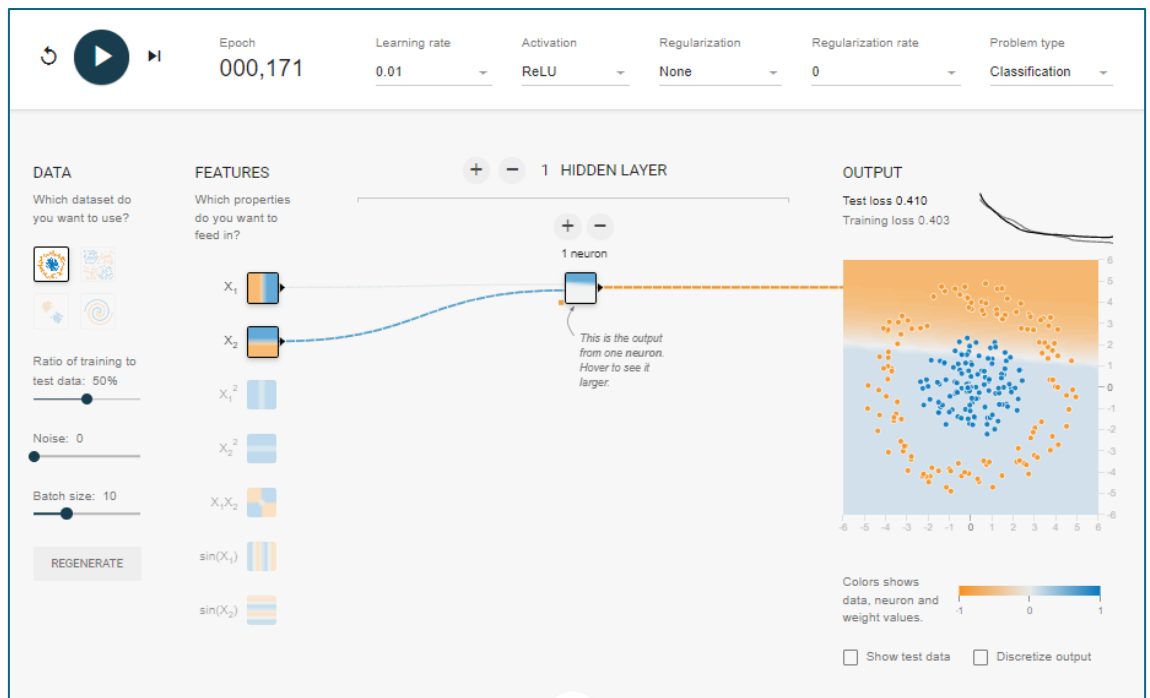
- Vemos que gráficamente vendría siendo un modelo simple de una red neuronal de una sola capa, como se observa a continuación:



Luego, al entrenarlo, podemos cambiarle la función de activación para ver cómo se comporta. En este caso, pasamos de *Tanh* a *Sigmoid*, por lo que vemos en la sección de output la diferencia del modelo... Una parece casi exponencial negativa mientras que la otra casi constante.



Hacemos lo mismo pero ahora con ReLU:



Lo anterior nos muestra la implementación de un perceptrón en un problema sencillo de clasificación (circle) y su experimentación con funciones de activación y observación como Tanh, Sigmoid y ReLU, evidenciando diferencias del impacto en la salida de la red.