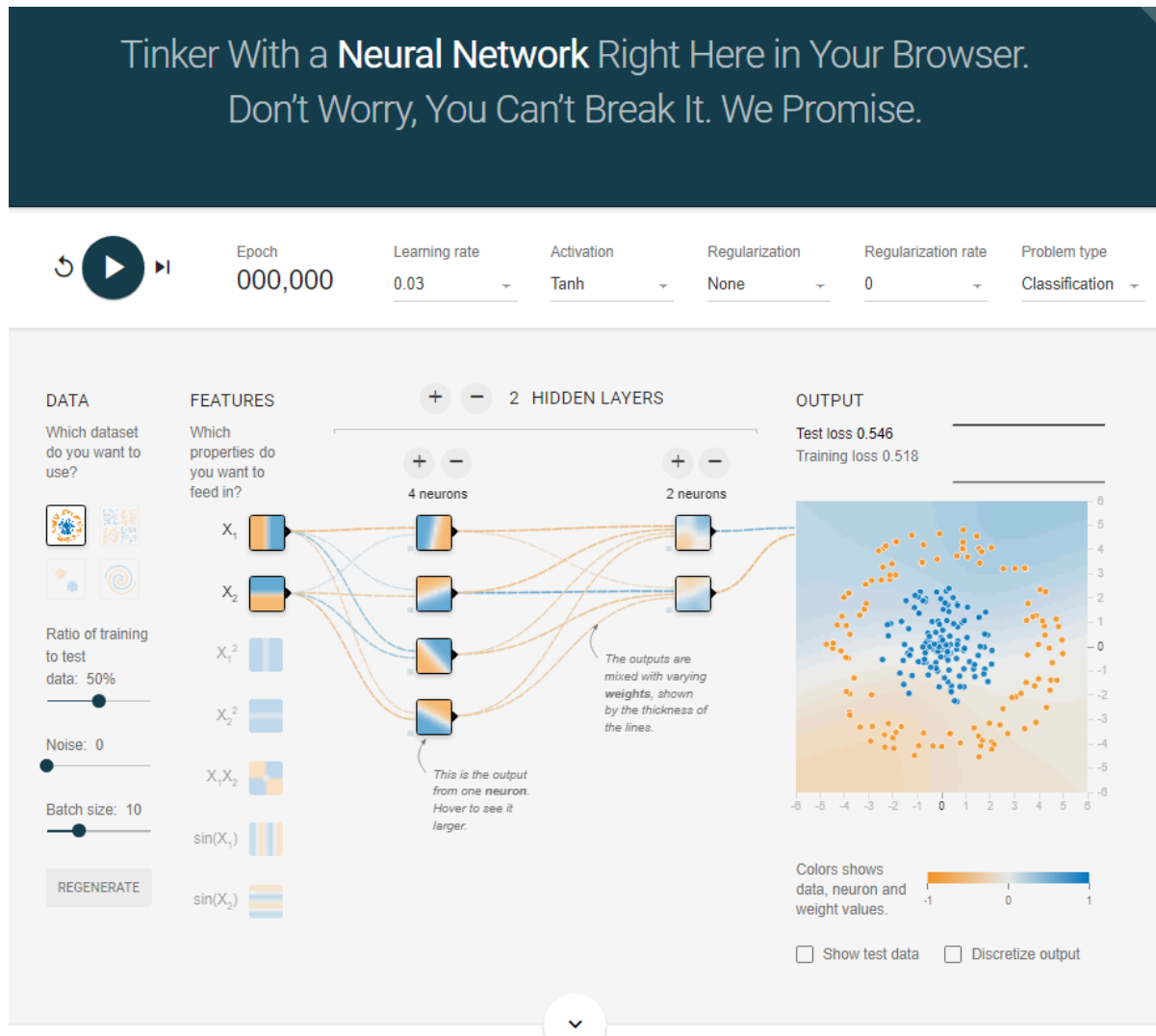
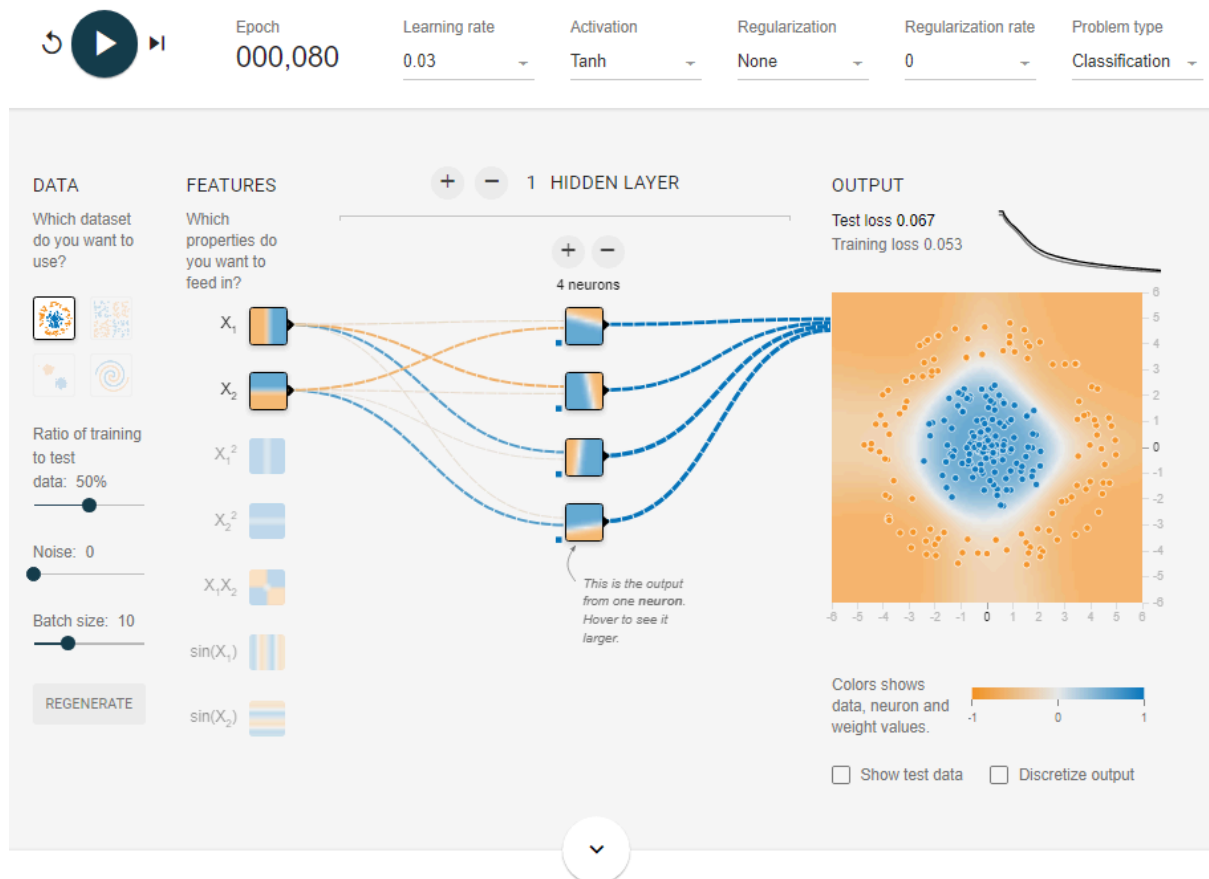


En el siguiente modelo está por defecto los parámetros del modelo

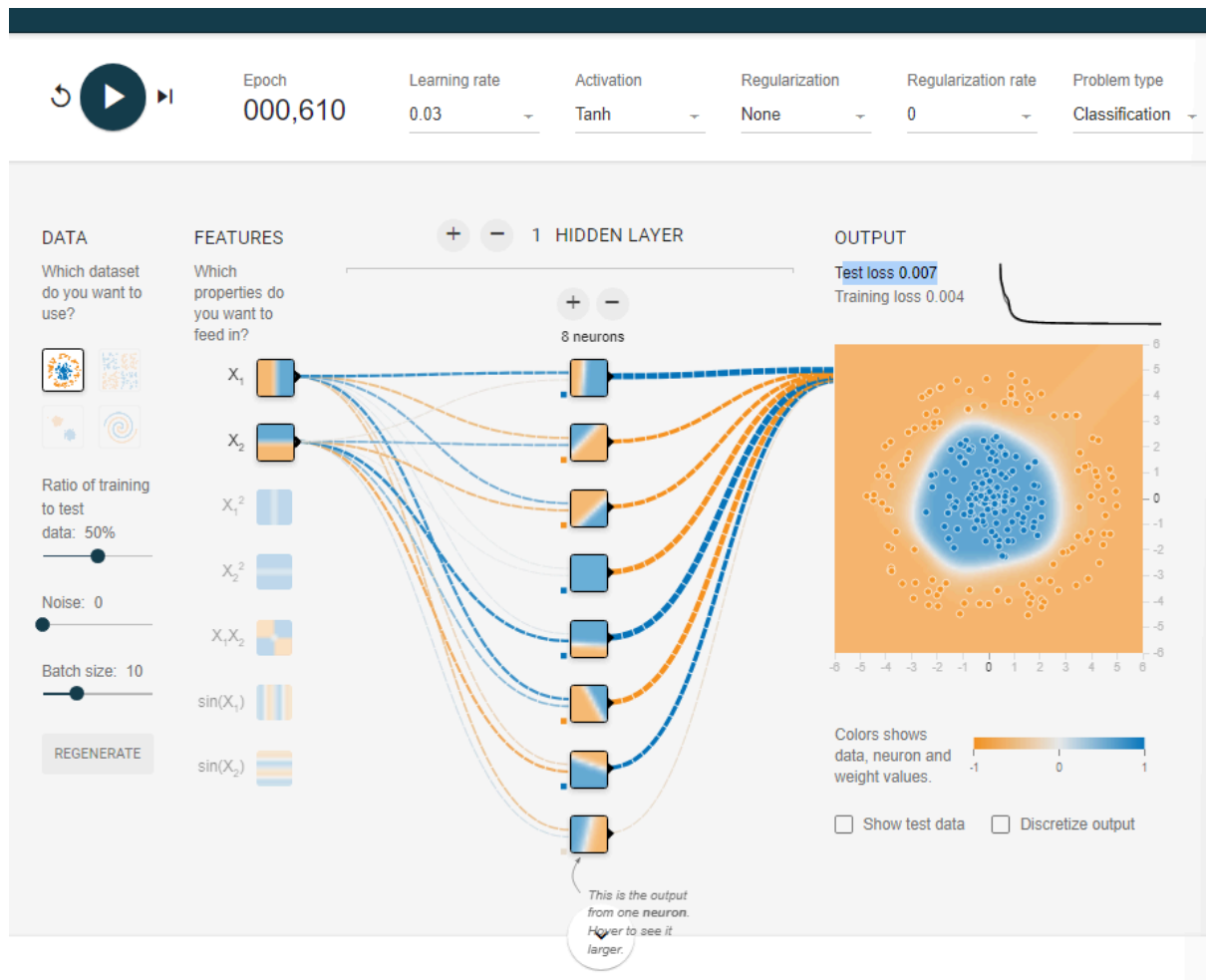


En el siguiente imagen modificamos el número de capas de 2 a 1 con 4 neuronas

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

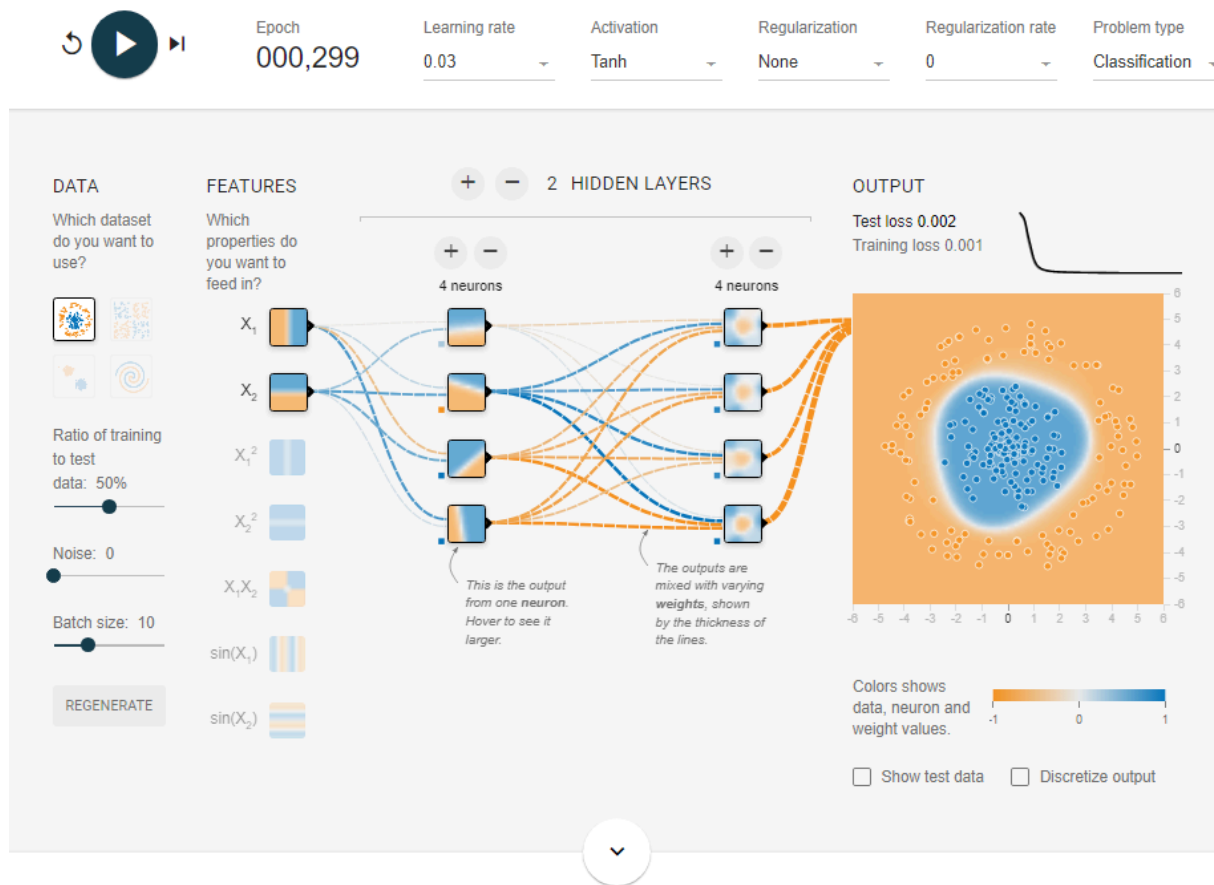


En 80 épocas el modelo ya estaba entrenado como resultado tenemos una tasa de pérdida del 0.067 ahora vamos a modificar y colocar 8 neuronas para ver cómo se comporta.



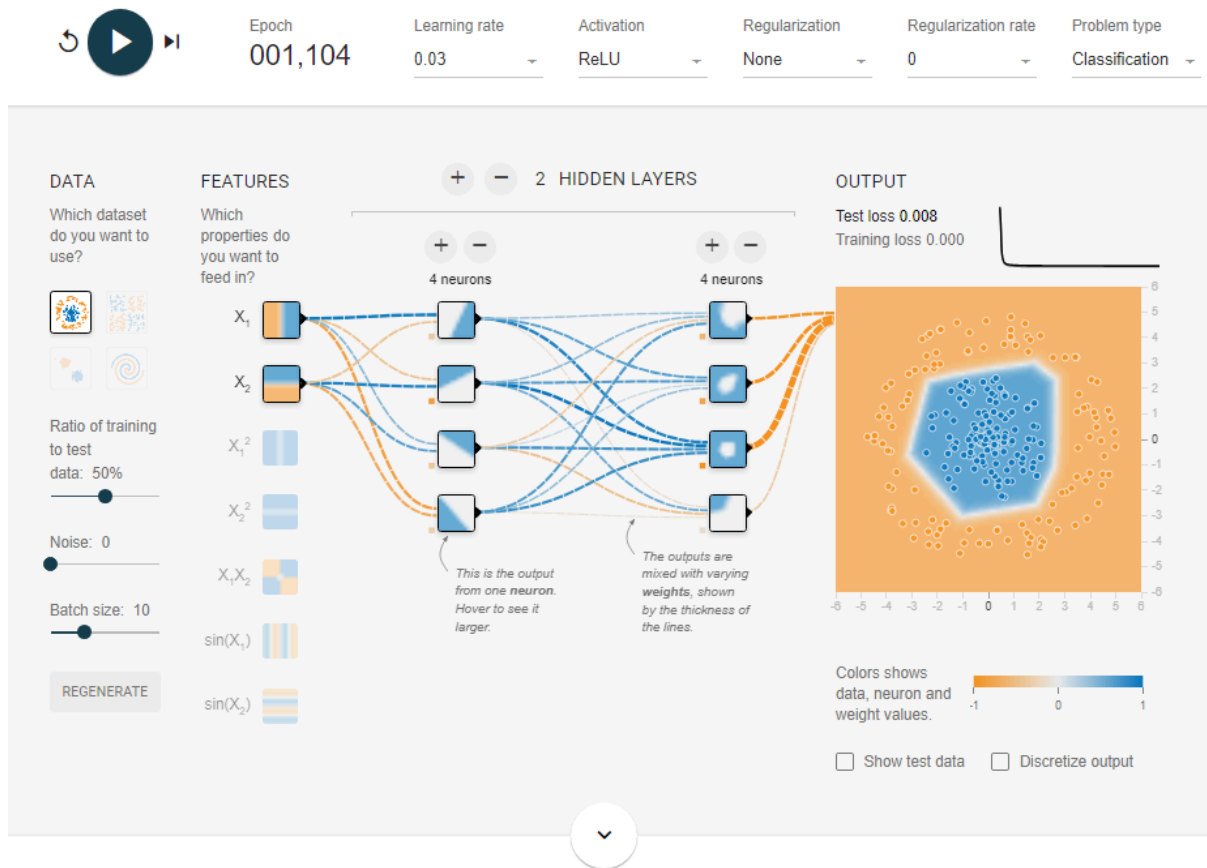
Con el aumento de 8 neuronas tardó más en entrenarse el modelo y subieron el número de épocas, pero la tasa de error bajó bastante en comparación con la anterior, ahora subiremos a dos capas de 4 neuronas para ver el comportamiento.

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.



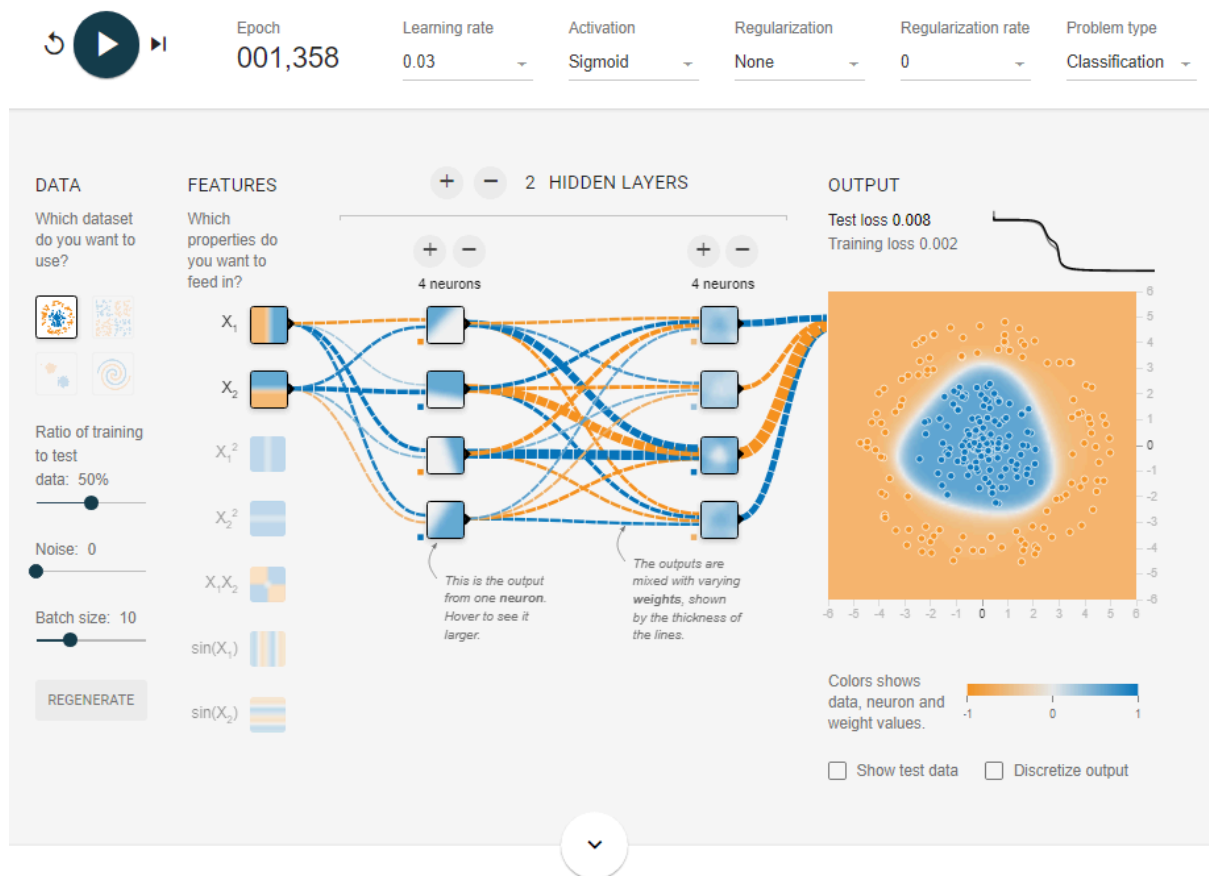
nuestro modelo aprendio mas rapido con dos capas de neuronas neuronas y tuvo una tasa de pérdida menor que el anterior modelo, en la siguiente imagen usaremos otra funcion de activacion:

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.



el modelo aprendió mucho más lento y no fue tan eficiente como con la función tanh, puede ser debido a sus salidas.

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.



en la anterior imagen encontramos una tasa de pérdida del 0.008 con una función de activación sigmoid el modelo aprendió mejor pero en 1358 épocas