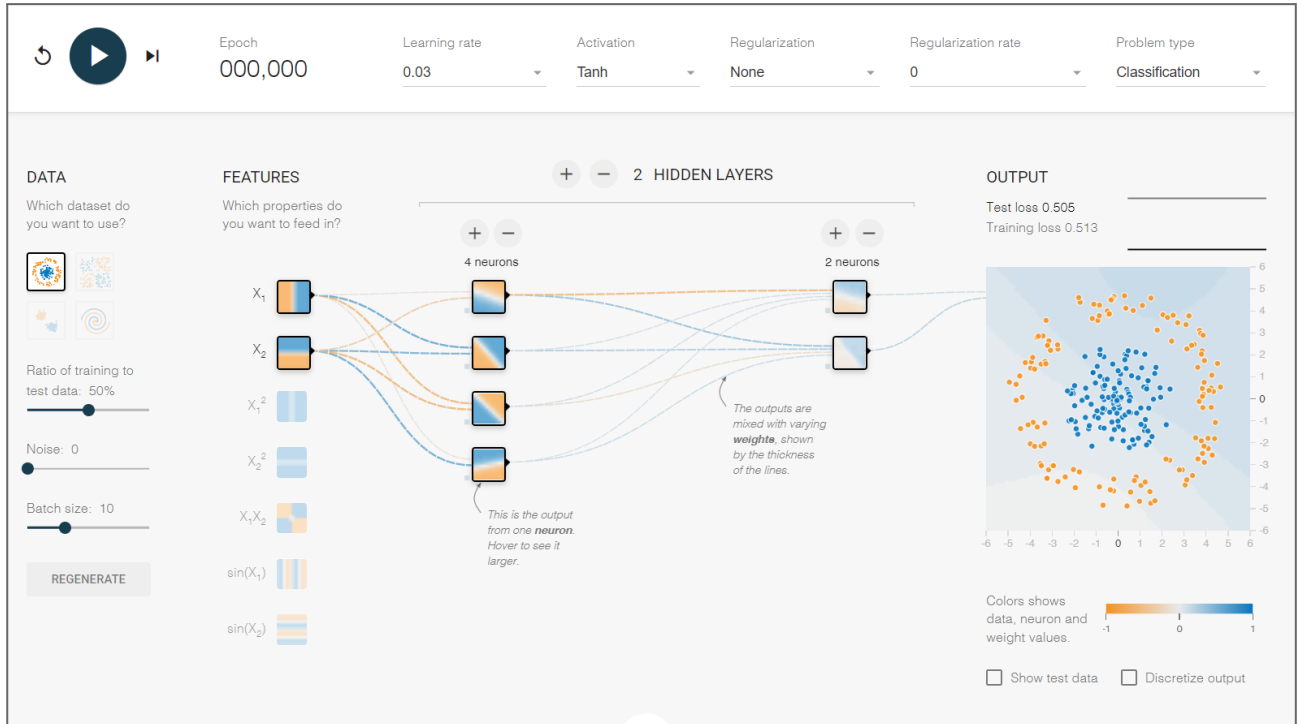


## SAA 03 - Tarea Resuelta

Entrar en [playground.tensorflow.org](https://playground.tensorflow.org)

El aspecto inicial es este:



Podemos observar en la columna de la izquierda el tipo de dataset que se va a utilizar para el entrenamiento (distribución en “circle”, en “exclusive or”, “Gaussian”, o “Spiral”). También podemos regular el % de datos que se van a emplear para el entrenamiento vs la comprobación de confiabilidad del resultado, así como el ruido y la cantidad de datos que empleará en cada iteración.

En la zona central de la aplicación definimos las capas de la red neuronal y cuántas neuronas tendrá cada una de ellas.:

**Features:** Son las propiedades de entrada que buscará el modelo. Para estos ejercicios iniciales no recomendamos usar más de dos datos de entrada.

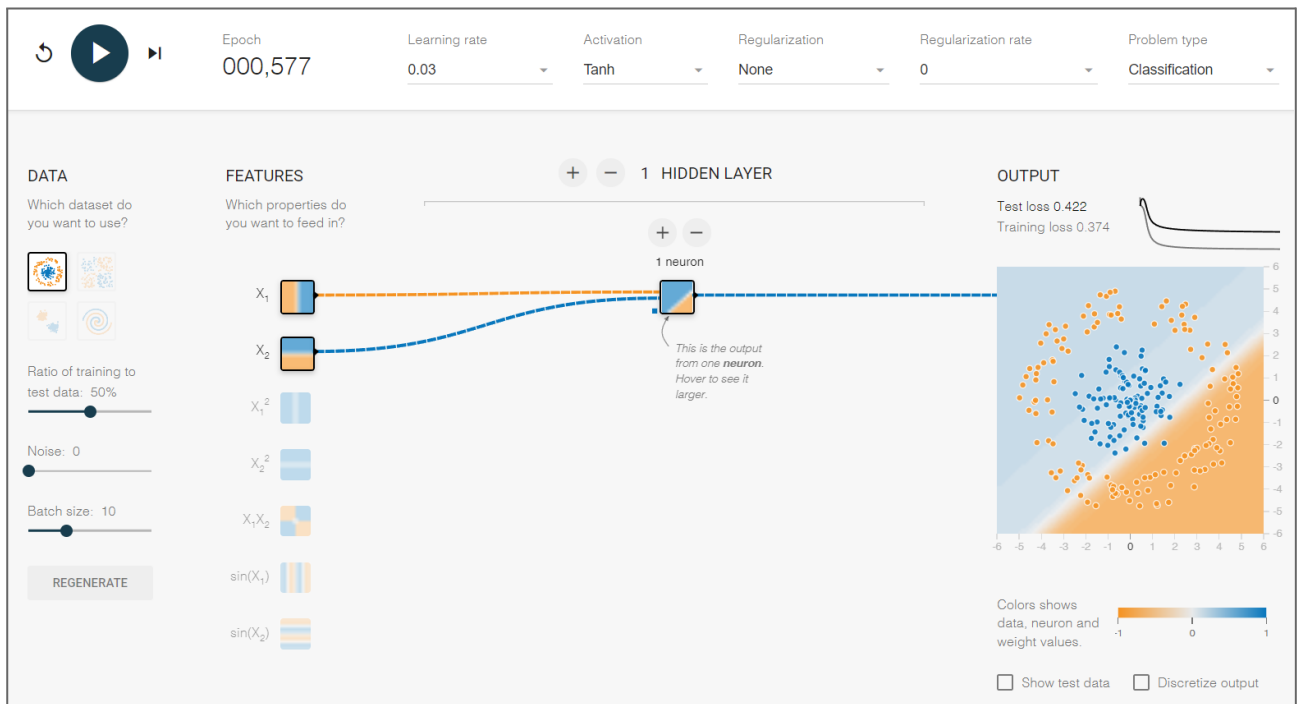
**Hidden Layers:** Son las capas de neuronas ocultas.

**Output:** La capa de salida o resultado. Cuando pongamos el modelo a funcionar se irá modificando el fondo separando en dos áreas distintas la distribución de puntos. Así podemos ver en tiempo real si nuestro modelo acierta o no.

En la banda superior hay posibilidad de ajustar también algunos parámetros.

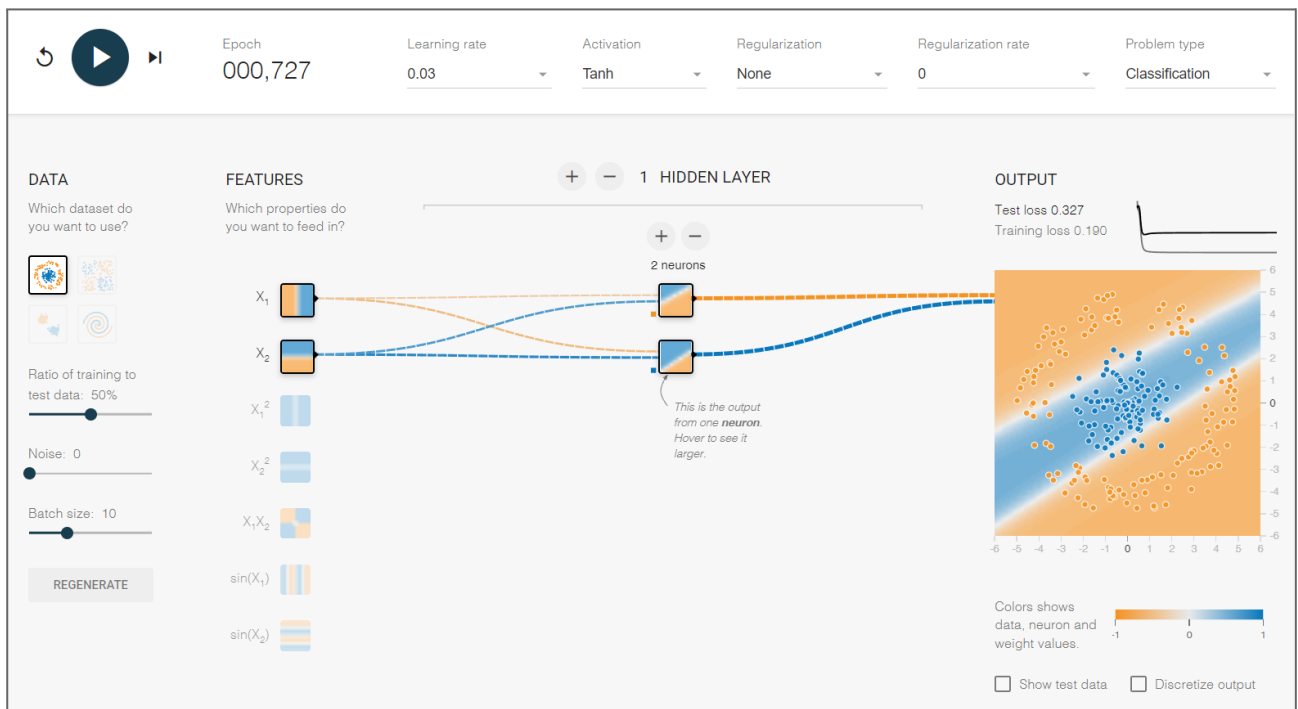
**Caso 1: Encontrar la estructura mínima necesaria para lograr una clasificación exitosa del dataset “circle”.**

Si probamos lo mínimo: una única capa con una única neurona, no logramos resolver la clasificación:

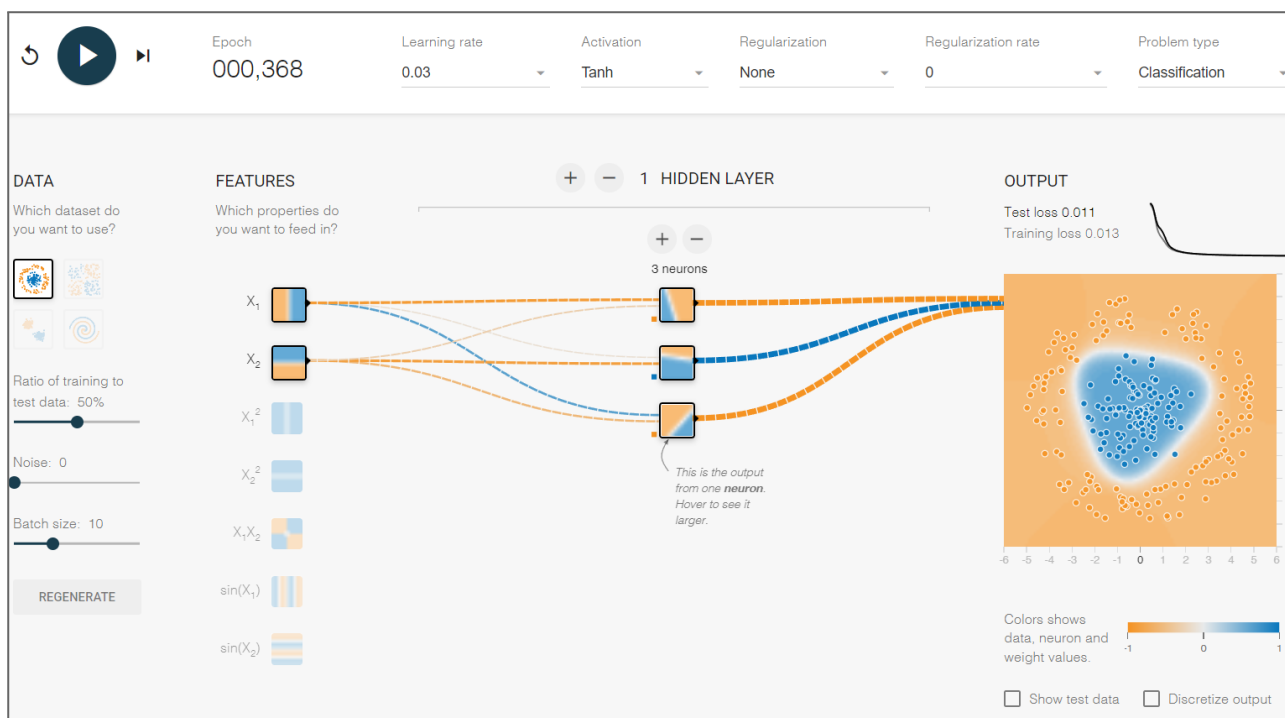


No pasa de dibujar una recta diagonal...

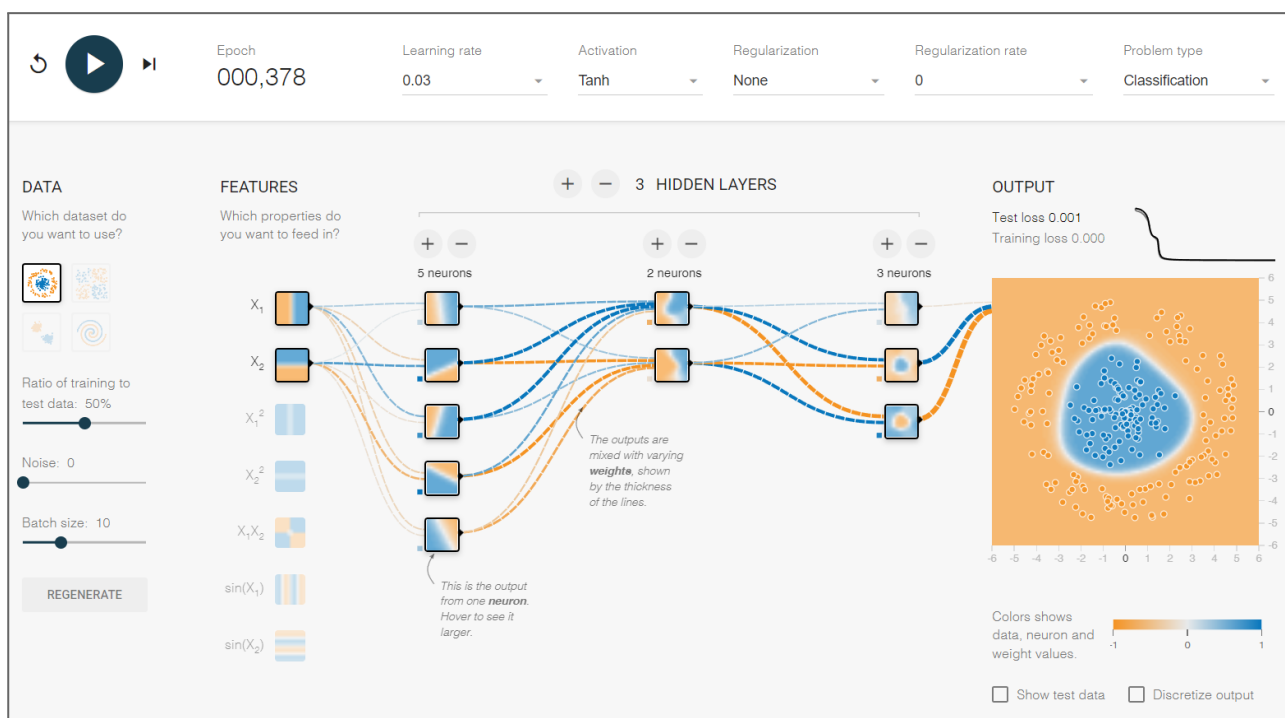
Si metemos dos neuronas en una única capa oculta, mejora un poco, pero sigue habiendo puntos naranjas en zona azul:



Pero al meter ya tres neuronas en la capa oculta... permitimos generar una especie de triángulo que separa adecuadamente los puntos de cada color en áreas distintas. Y además lo hace en un plazo de tiempo (o número de iteraciones) bastante corto:



Si añadimos más neuronas y/o más capas seguiremos encontrando buenos resultados, pero no necesariamente con menor cantidad de iteraciones o gasto de tiempo, por ejemplo:



Así que con una única capa oculta que contenga tres neuronas obtenemos un modelo capaz de realizar la clasificación.

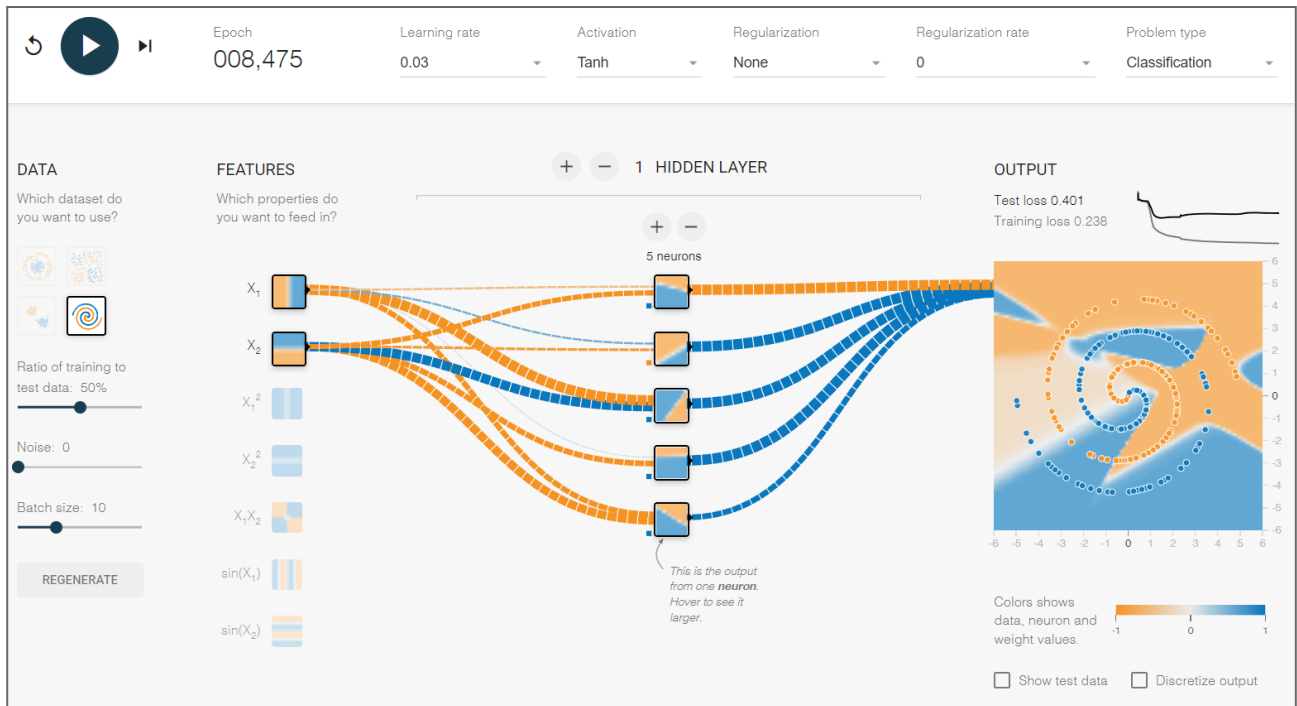
## Caso 2: Encontrar una estructura que logre una clasificación exitosa del dataset “spiral”.

En el caso de la distribución de puntos en forma espiral es más complejo encontrar una “fórmula” que dibuje las áreas de forma correcta.

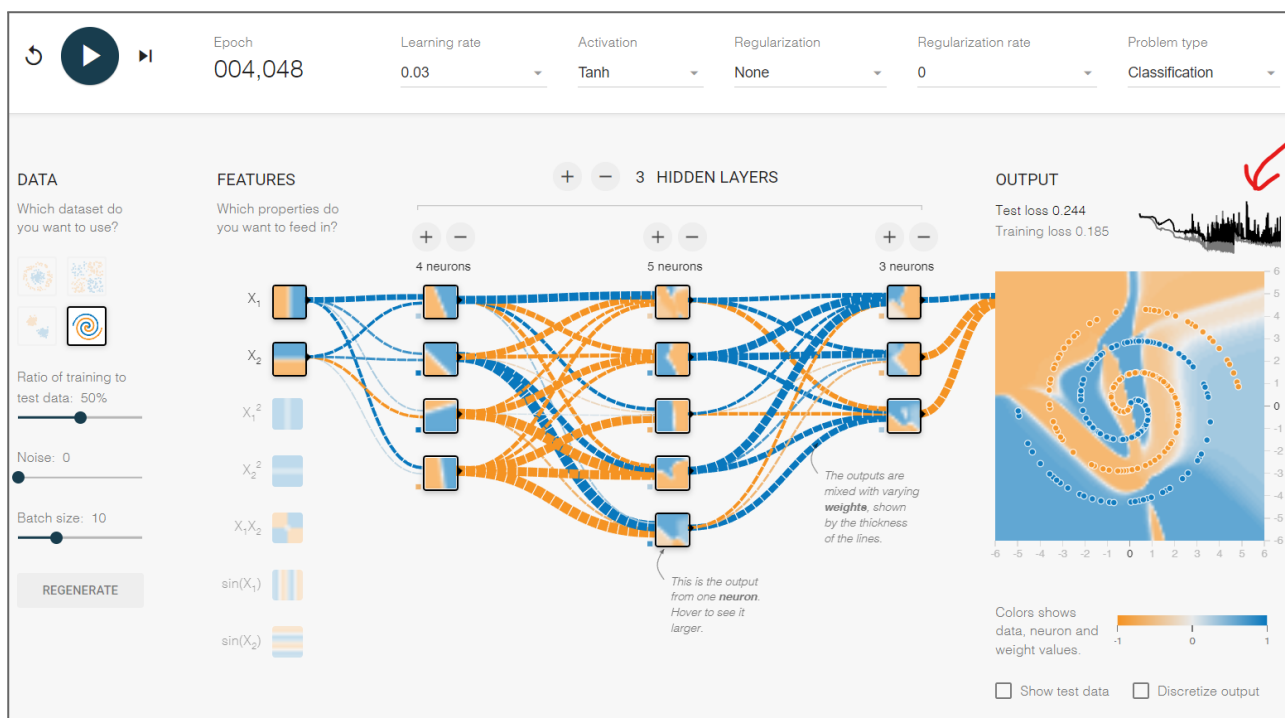
El alumnado puede hacer diversas pruebas, con una o varias capas ocultas, con diferente número de neuronas en cada caso.

Por ejemplo:

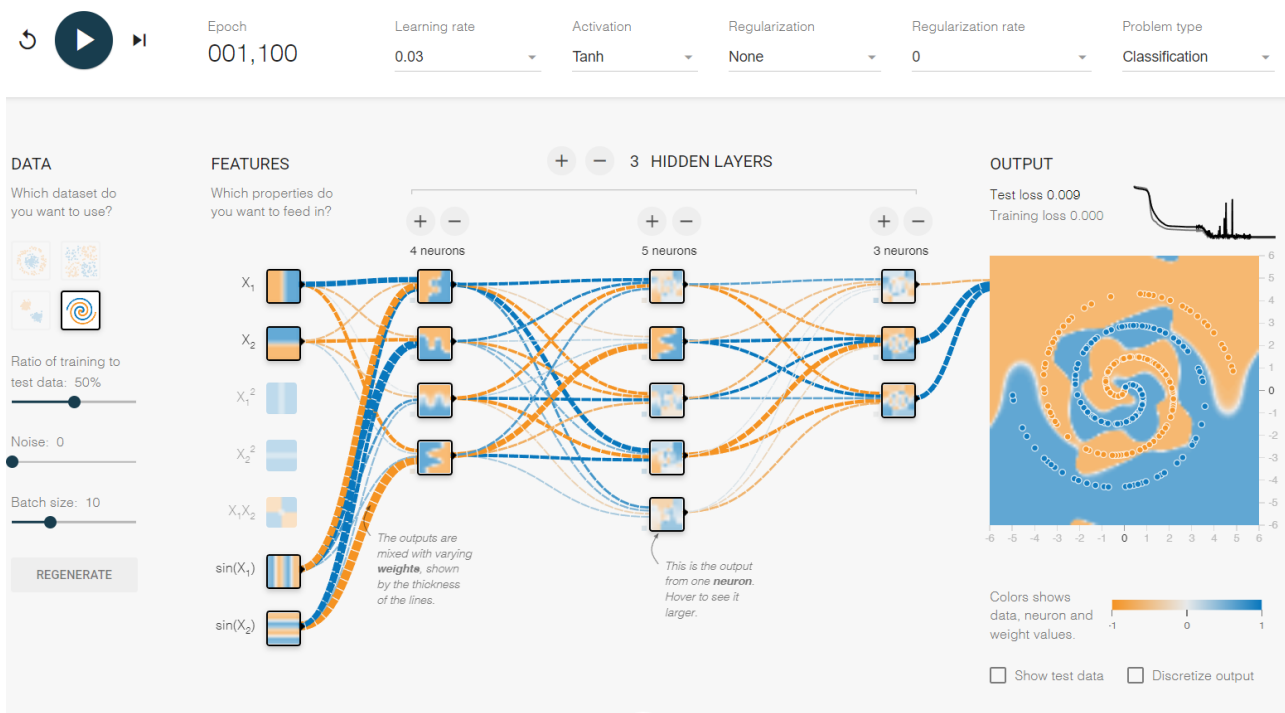
- Una capa con cinco neuronas: Por mucho tiempo que dejemos iterando el modelo, no logra encontrar una forma de clasificar satisfactoria.



- Tres capas ocultas, con cuatro, cinco y tres neuronas respectivamente. Nos conduce a un modelo inestable (muchas oscilaciones en la gráfica arriba a la derecha).

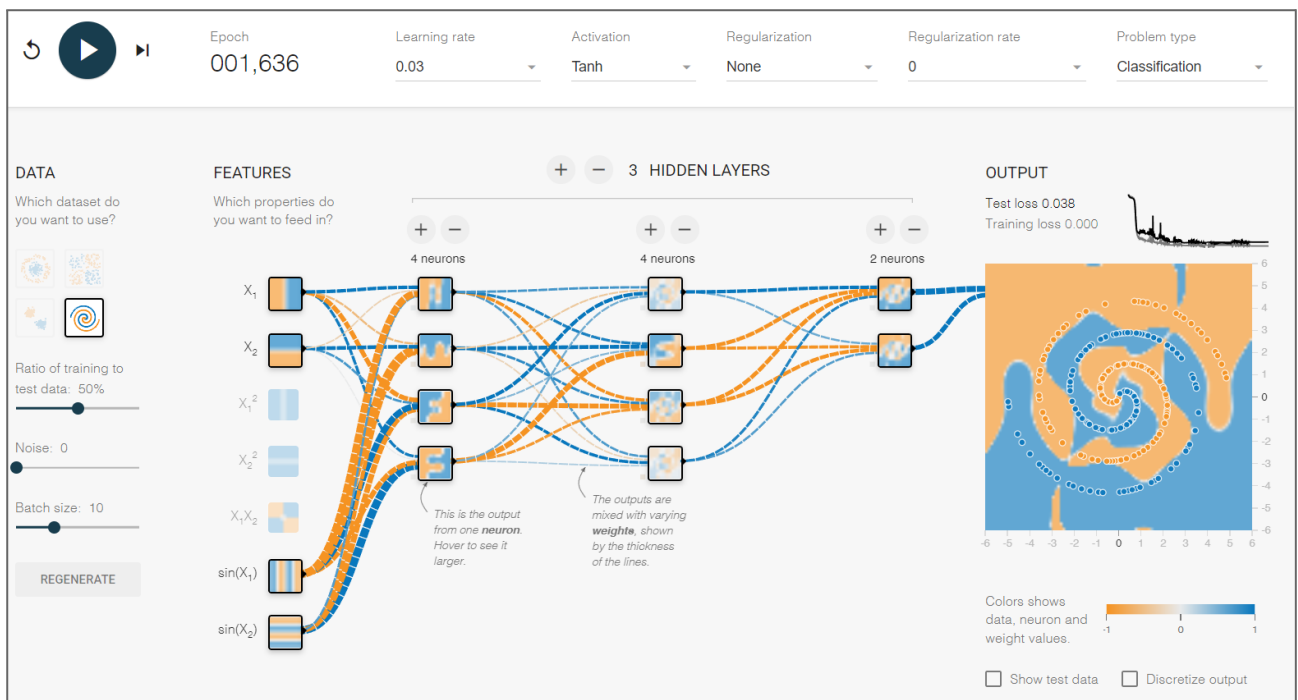


Hasta que no introduzcamos algún parámetro de entrada más, va a ser muy complicado encontrar un modelo correcto. En concreto, al incluir en las propiedades de entrada  $\sin(X_1)$  y  $\sin(X_2)$ , logramos en un plazo de tiempo razonable una solución satisfactoria:

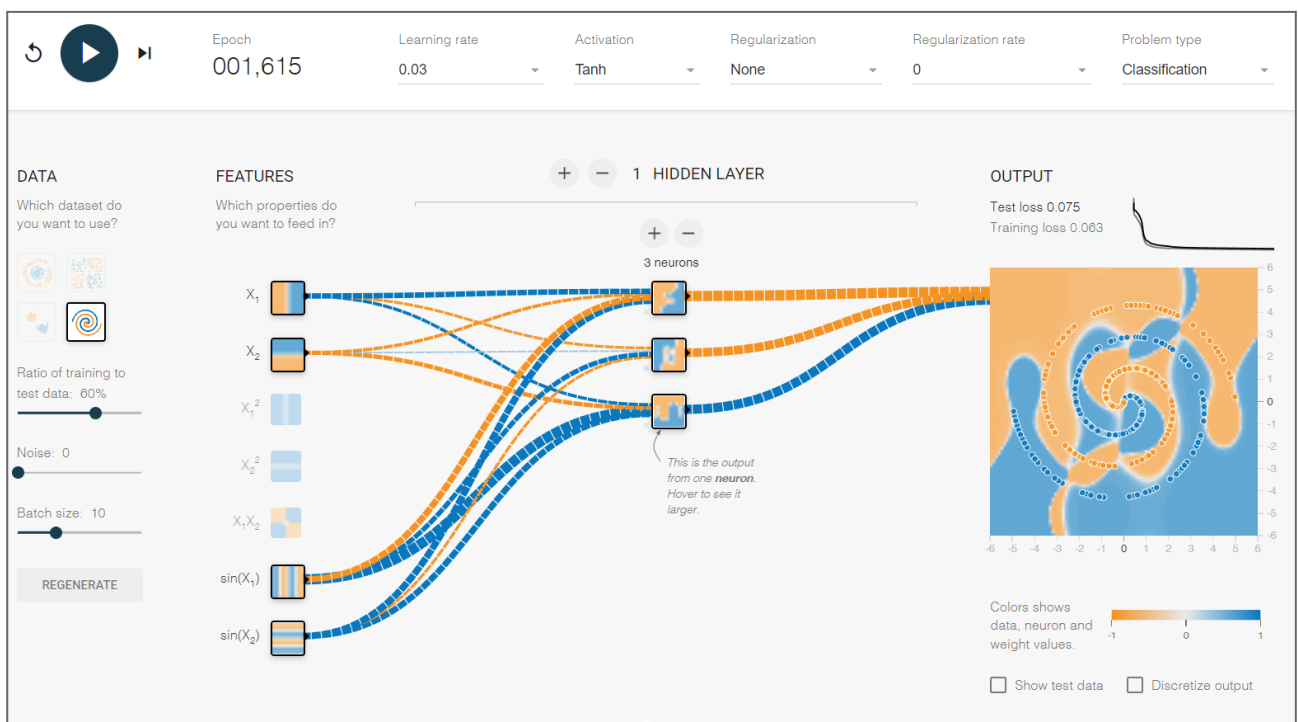


El alumnado podría ir seguir haciendo intentos reduciendo el número de capas y neuronas. Por ejemplo:

- Tres capas ocultas con cuatro, cuatro y dos neuronas respectivamente.



- Una única capa con tres neuronas:



De manera que la solución óptima en cuanto complejidad de la red neuronal se puede encontrar con un modelo similar al propuesto en esta última imagen, con una única capa oculta y tres neuronas, siempre que intervengan en la capa de entrada cuatro propiedades, siendo dos de ellas de tipo *sin*.

Pero este modelo tarda más (necesita más iteraciones) para encontrar la solución óptima. Podemos añadir más capas y neuronas para obtener buen resultado en menor tiempo.