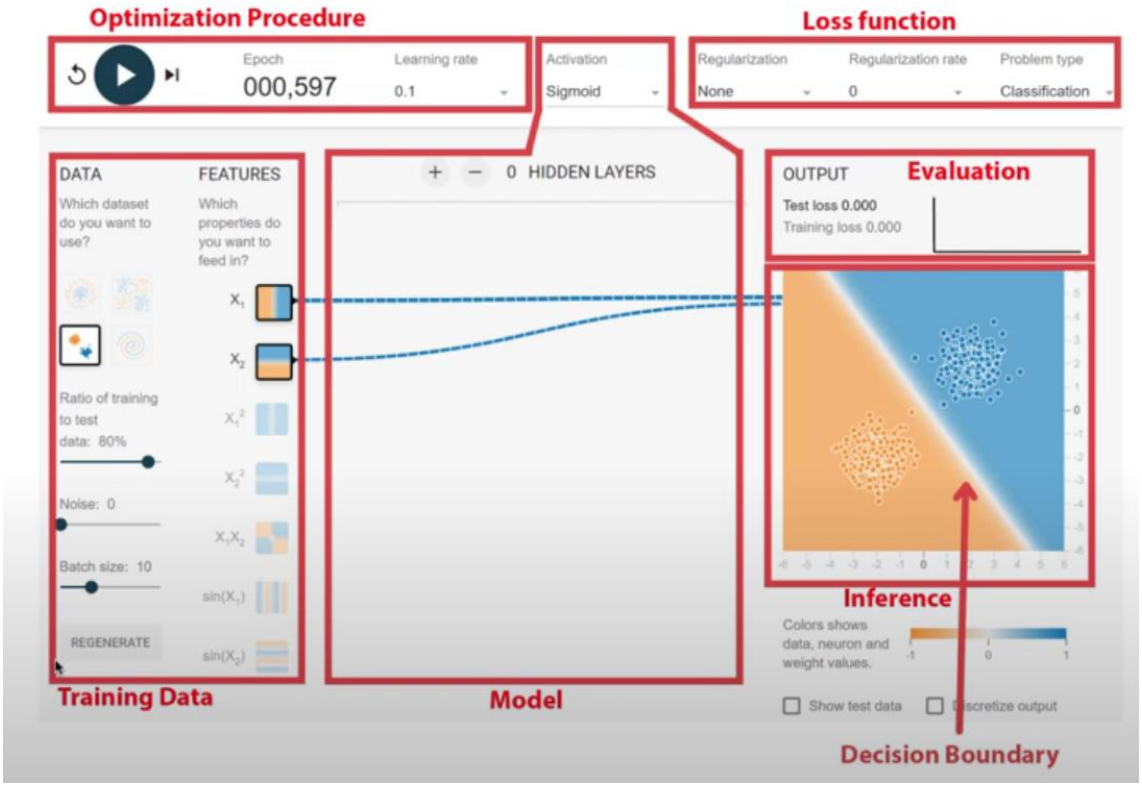
En esta oportunidad tomare probaremos inicialmente con la Data en entrada de forma circular Usando la herramienta PlayGround de TensorFlow (https://playground.tensorflow.org/).

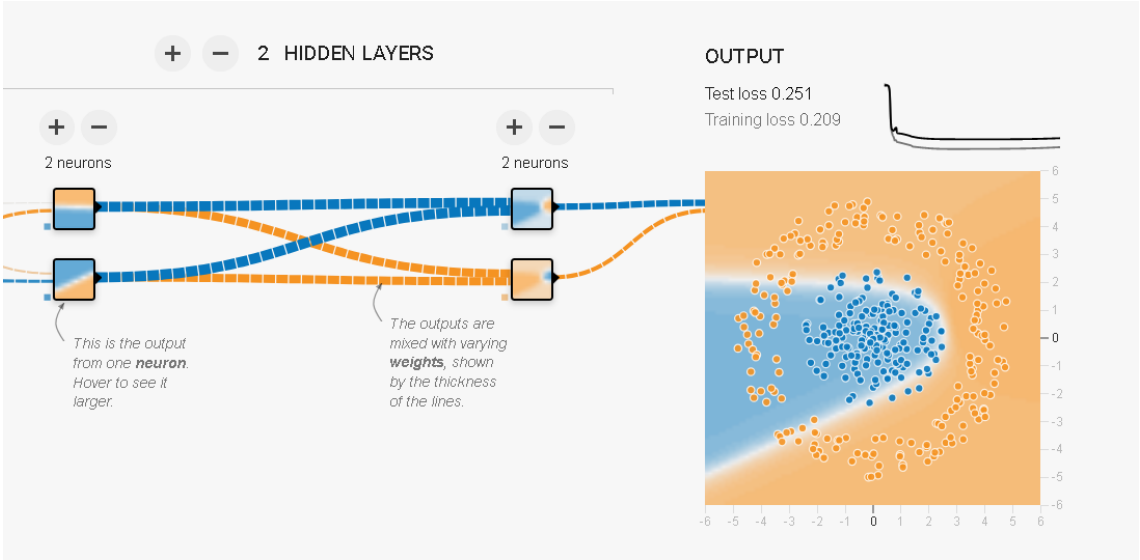


Empezamos: pruebas

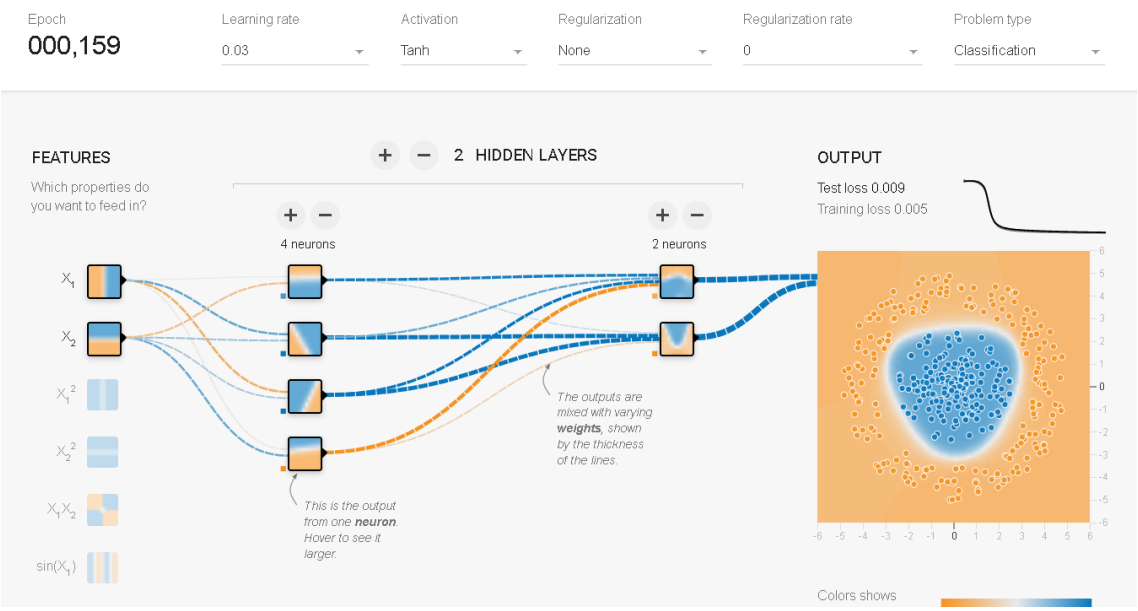
INICIO

- Data training: Seleccionamos Data Circula,80% training y 20% evaluación, tomaremos 2 características tomares x1 y x2.
- Parámetros de Optimización vemos que se inicial inicialmente algunos pesos aleatorios Partiremos de Learning rate=0.03.
- Function activación: se va elegir Tan por esta ocasión, revisamos y hay un prden de mejoras pero se
- Modelo: Empezaremos con una Capa y 2 neuronas
- Loss function no lo usaremos por ahora
- Problema escogimos clasificación porque tenemos 2 características, vamos a evaluar las ponderaciones que daría el modelo a cada clase

CASO1: Se observa que no entrena bien el modelo y además tiene error de entrenamiento del 0,2, los pesos son mayores viendo el grosor de las salidas, vamos a aumentar 1 capa con 4 neuronas mas

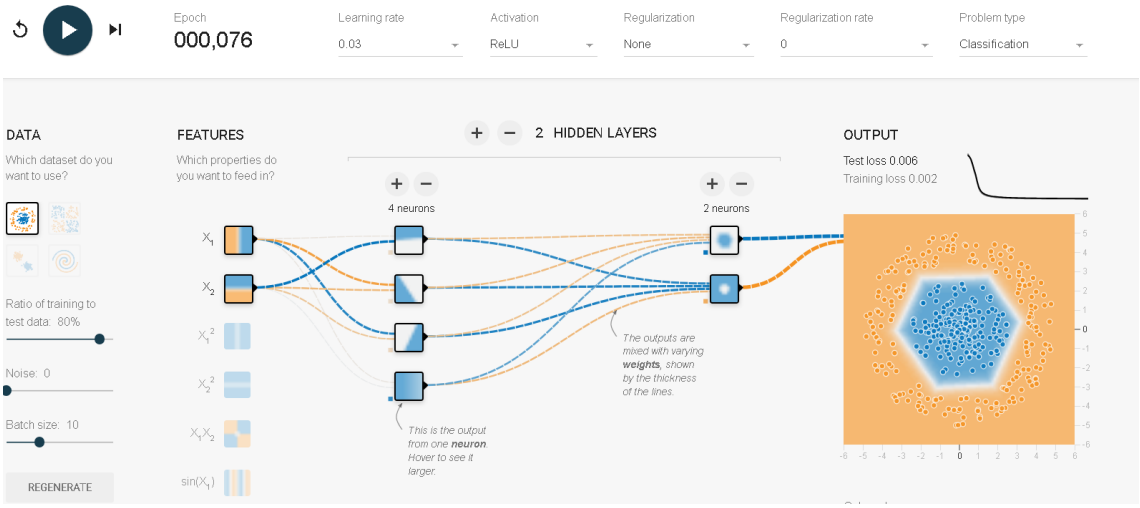


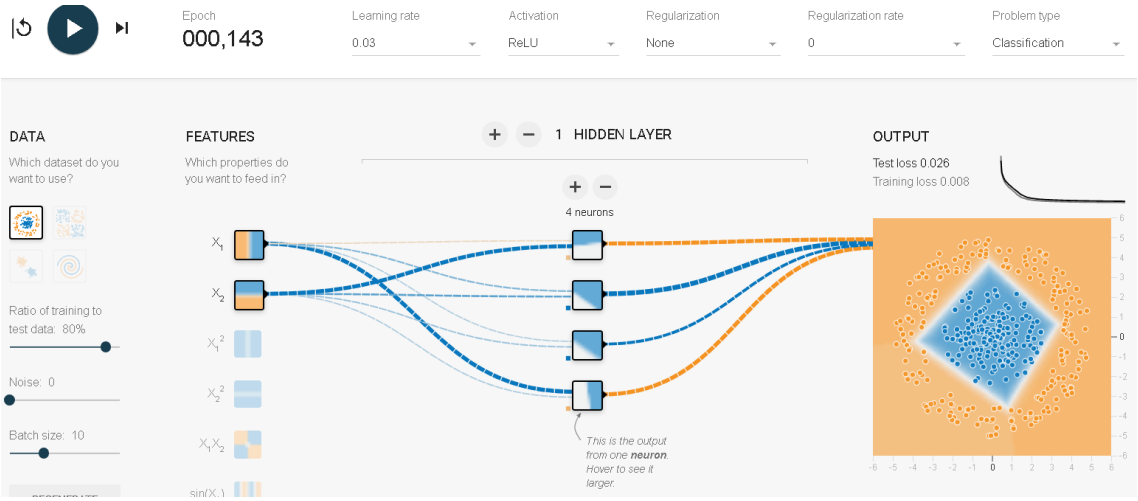
CASO2: Utilizaremos 2 Capas y 6 neuronas, Conseguimos la forma circular por la función de activación con menos épocas solo 159 aprox épocas y test training de 0,005 y se observa gráficamente que la red neuronal pudo entrenar de manera correcta, vemos en la capa 1 que la neuronas formaron input de las formas verticales y horizontales 4 , consiguiendo más formas complejas incluye diagonales que no existía, esto es producto también de la función de activación que elegimos tangente ,que ayudo a esto, vamos a cambiar en el siguiente caso a Relu para ver si efecto



Caso3: Cambiamos a Relu la función de activación, se observa que con la función de activación Relu es mucho mejor se llega a menos épocas consiguiendo errores bien bajos y se observa resultado forma hexagonal dada por la función Relu que es lineal resultado en menos épocas 176 , erro training de 0,02 se consigue buen resultado, vemos también que la segunda capa no

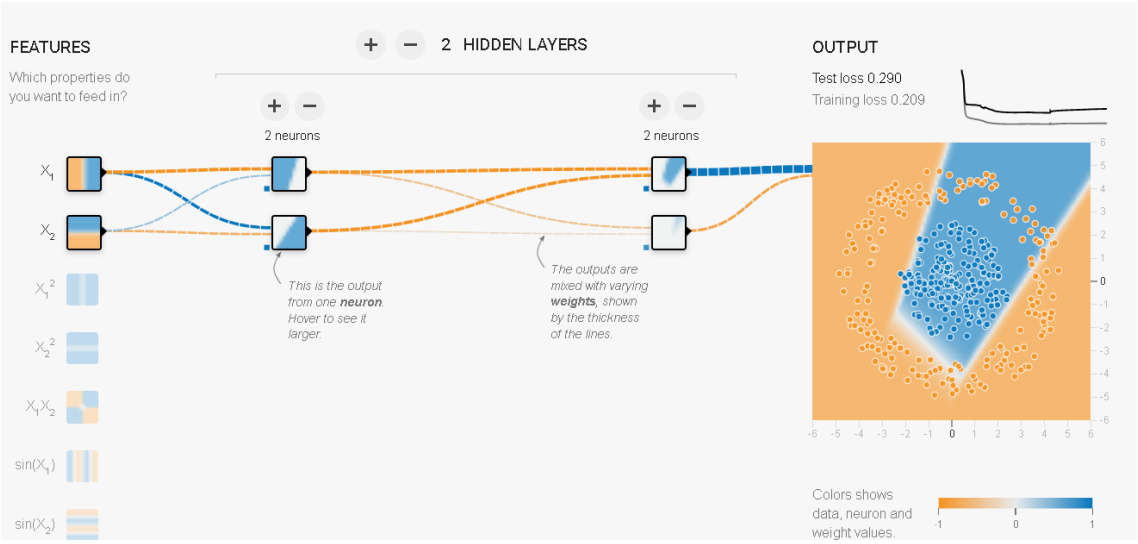
está aportando nada, así que eliminamos esta capa concluiremos que no necesitamos capas profundas para este modelo porque con una sola capa y se conseguir buenos resultados en la siguiente muestra. Vemos en el siguiente grafico

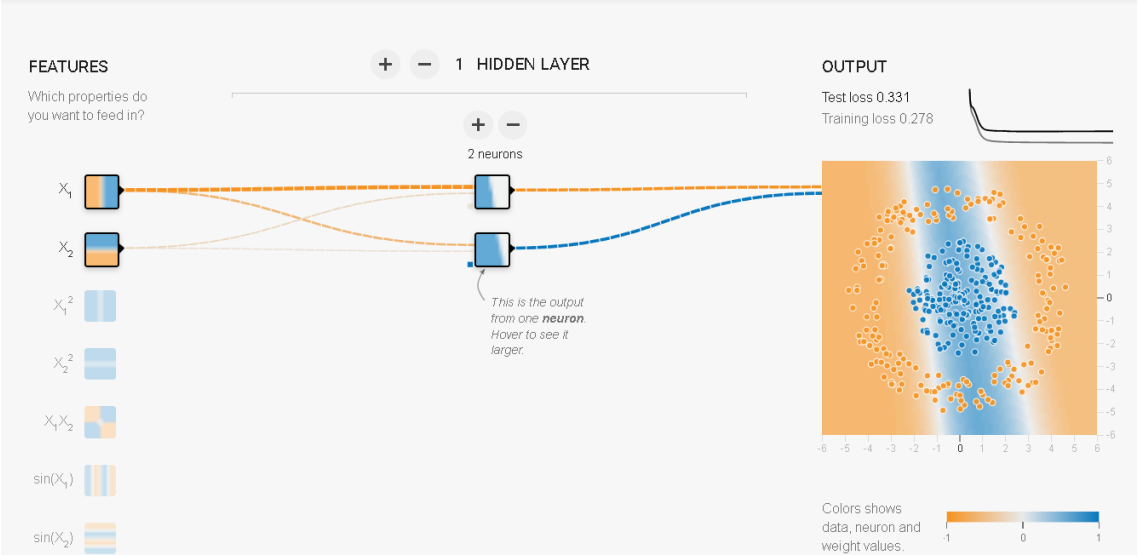




Caso4: En los siguiente paso evaluamos con menos neuronas por capa tomamos (2 neuronas x capa) en do modelos en una capa y con dos capas, en ninguno de los dos modelos se obtuvo un modelo eficiente, estos modelos no convergen, las salidas de los errores de entrenamiento y validación varían mucho, esto es muy claro que tiene una **varianza muy elevada**, reiniciamos

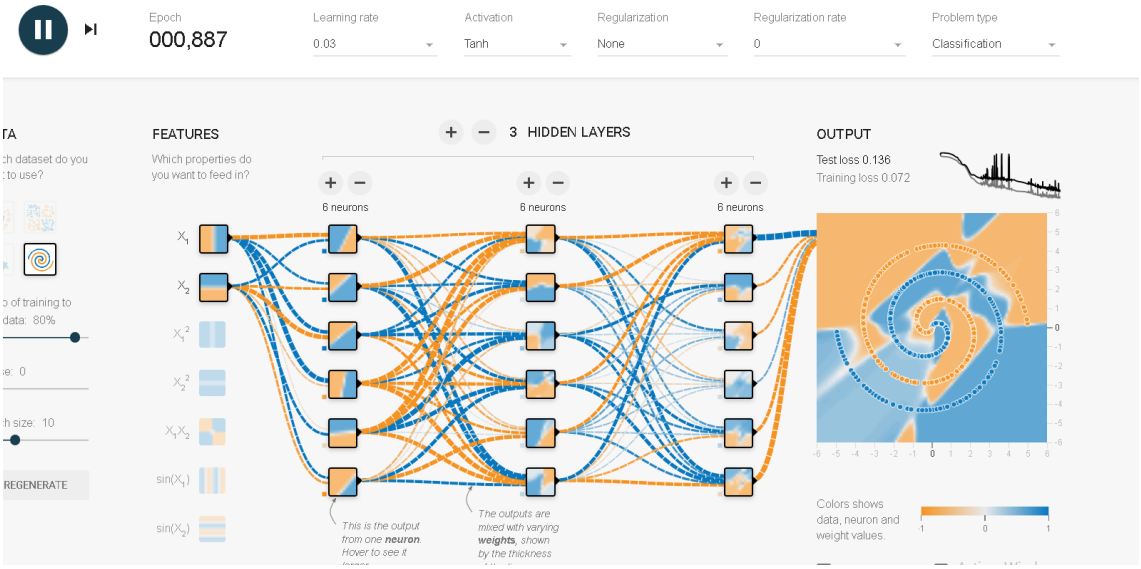
varias veces pero **no llega a encontrar mínimos locales**. Lo podemos observar en los 2 cuadros siguientes.



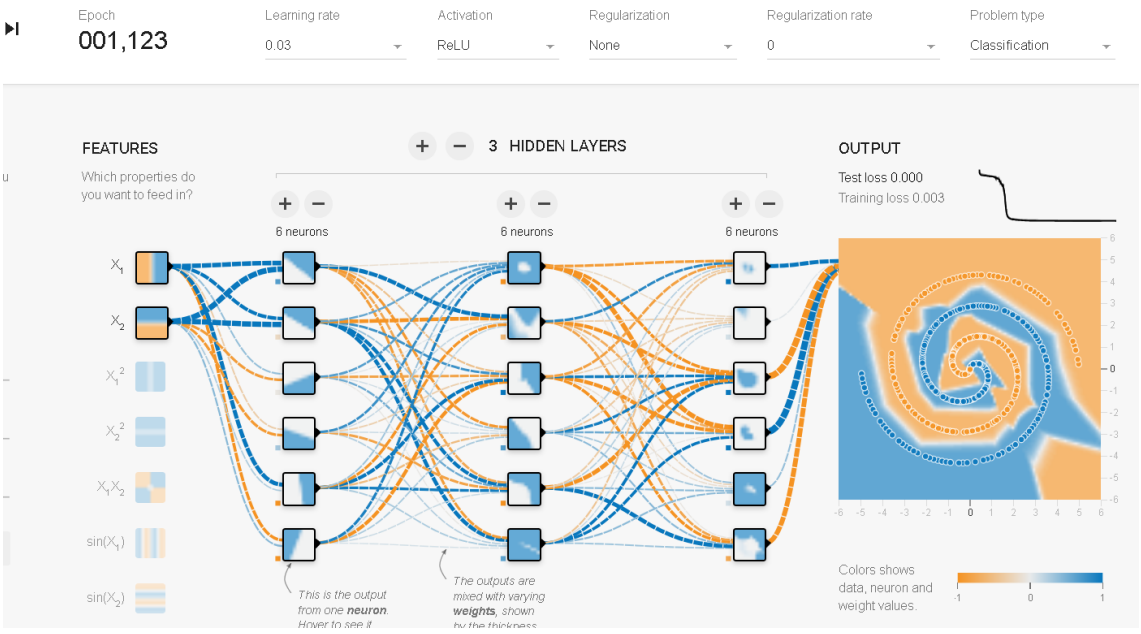


Caso 5: hicimos más compleja la red tomamos los datos de modelo entrada espiral, Por lo tanto hay datos de entrada que necesitaran análisis con más capas y otras no, vamos para concluir que pasaría en una red como datos de espiral que es más compleja

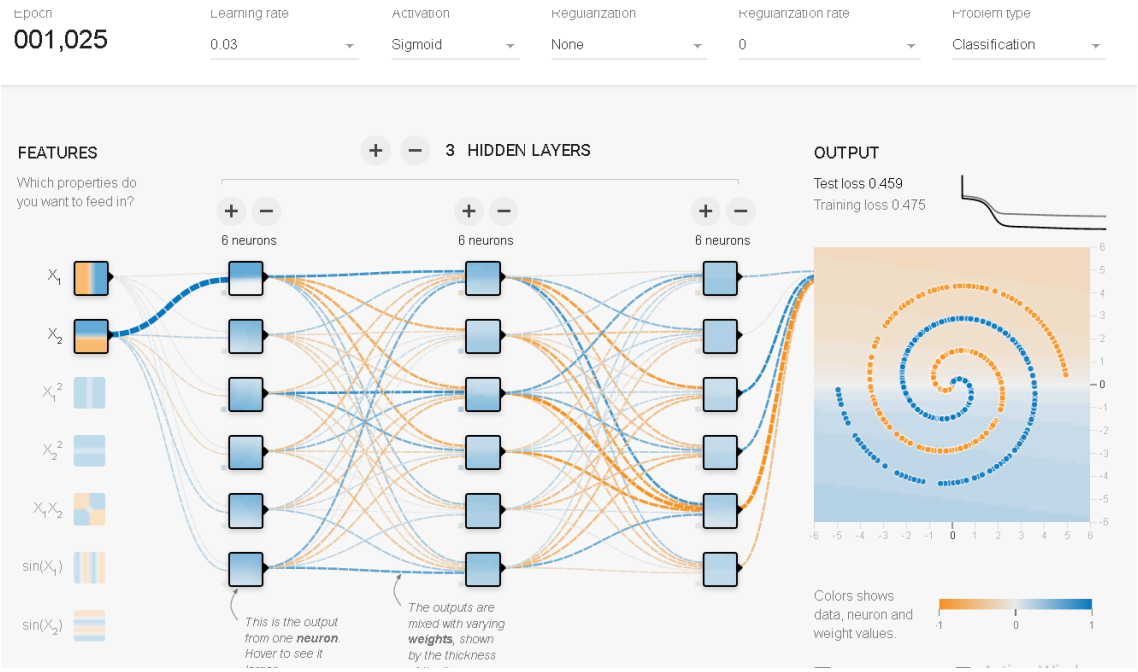
Para empezar vamos a crear 4 capas ocultas con 6 neuronas cada una, bajo una función de activación Tangente, se notó que se necesitaba más tiempo de entrenamiento, esto pasa en modelos muy complejos con muchas capas, desde ya también se notó que por ser más compleja el modelo se necesitas más consumo de recursos, intento formar la gráfica pero llegó a sobre ajustarse veamos y las épocas siguieron aumentando.



Caso6: vamos a Cambiar la función tangente por la de activación Relu y noto que mejoro bastante, Aprendió el modelo, pero toma muchas épocas para llegar a entrenarse.



Caso7: Probaremos con otra funciona la sigmodial, el resultado se empeora ,si nosotros cambiamos a la función de activación sigmodial, no logra clasificar los datos, siguen avanzando las épocas con mucho costo operacional , aquí se ve claramente que la red está teniendo a lo que se llama desvanecimiento de gradiente, y está pasando con la función de activación sigmoide en este caso, hace que cambie los pesos en la red neuronal



Los hiperparametros afectan bastante así como la arquitectura a seleccionar, al inicio tomamos 0,03 pero antes habíamos tomado 3 ó 0,0001, el enteramiento oscila y no converge se tomó de referencia para alcanzar el mínimo de la función de perdida en 0,03,a una próxima y mejor ajustada para medir el modelo, Cada tipo de datos tiene una forma de modelo que se ajusta y entrena mejor que otros y un tipo de función de activación que sea la mejor. Hay que ajustar bien las capas y neuronas, experimentando con distintas redes porque nos demuestra que de ello depende el aprendizaje de patrones complejos de datos, unas opciones también seria validar pruebas con ruido o integrando a neuronas más características de entrada o usar para regularización para mejorar más el modelo.