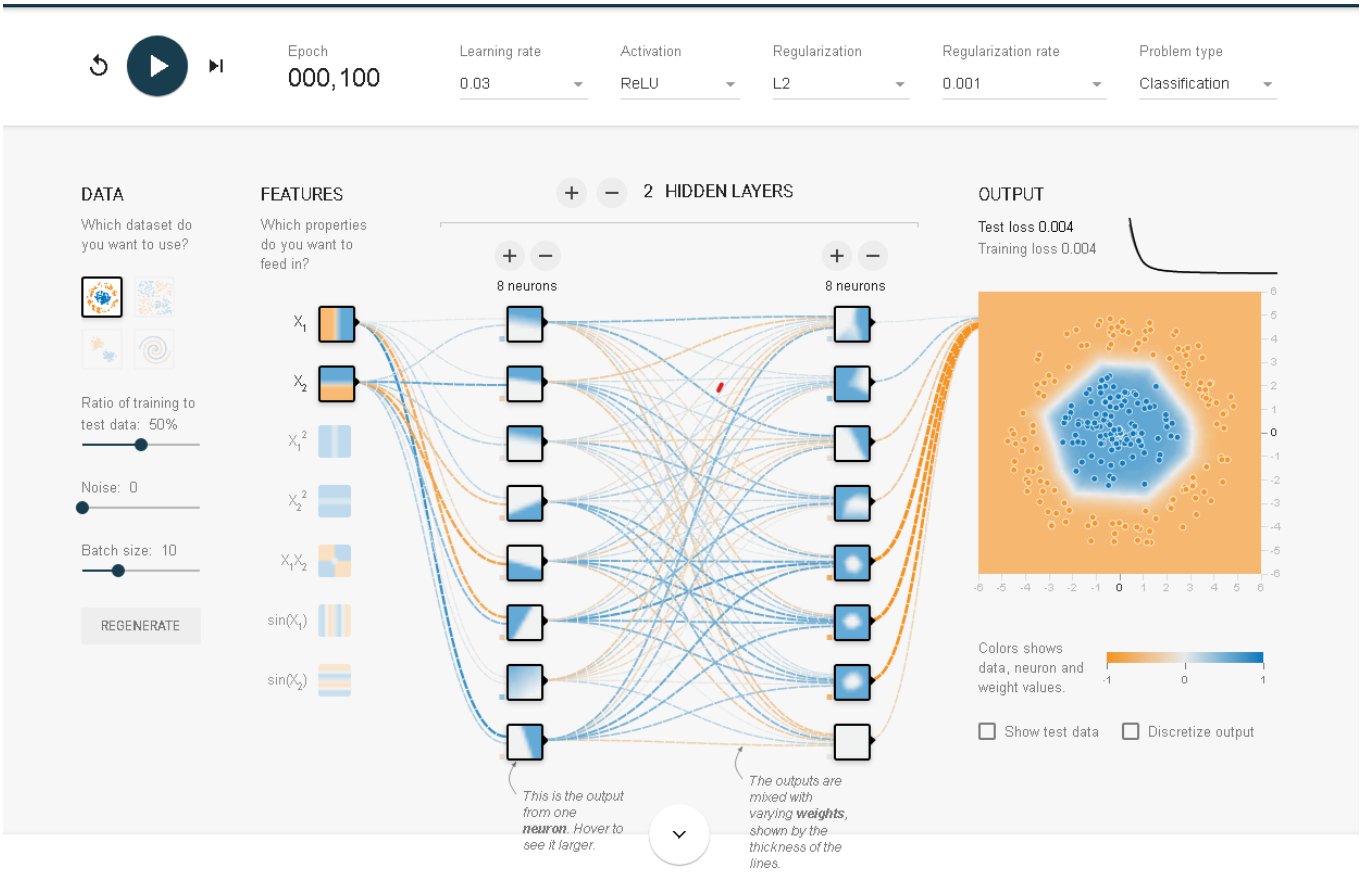
人工智慧-hw04 Explore Tenserflow Playground

40823117L 方國丞

(A) Circle data set:

> Can you obtain a test loss of less than 0.1 using only the raw input features? i.e., using only X1 and X2 as features? Please share a screenshot of your network as described.

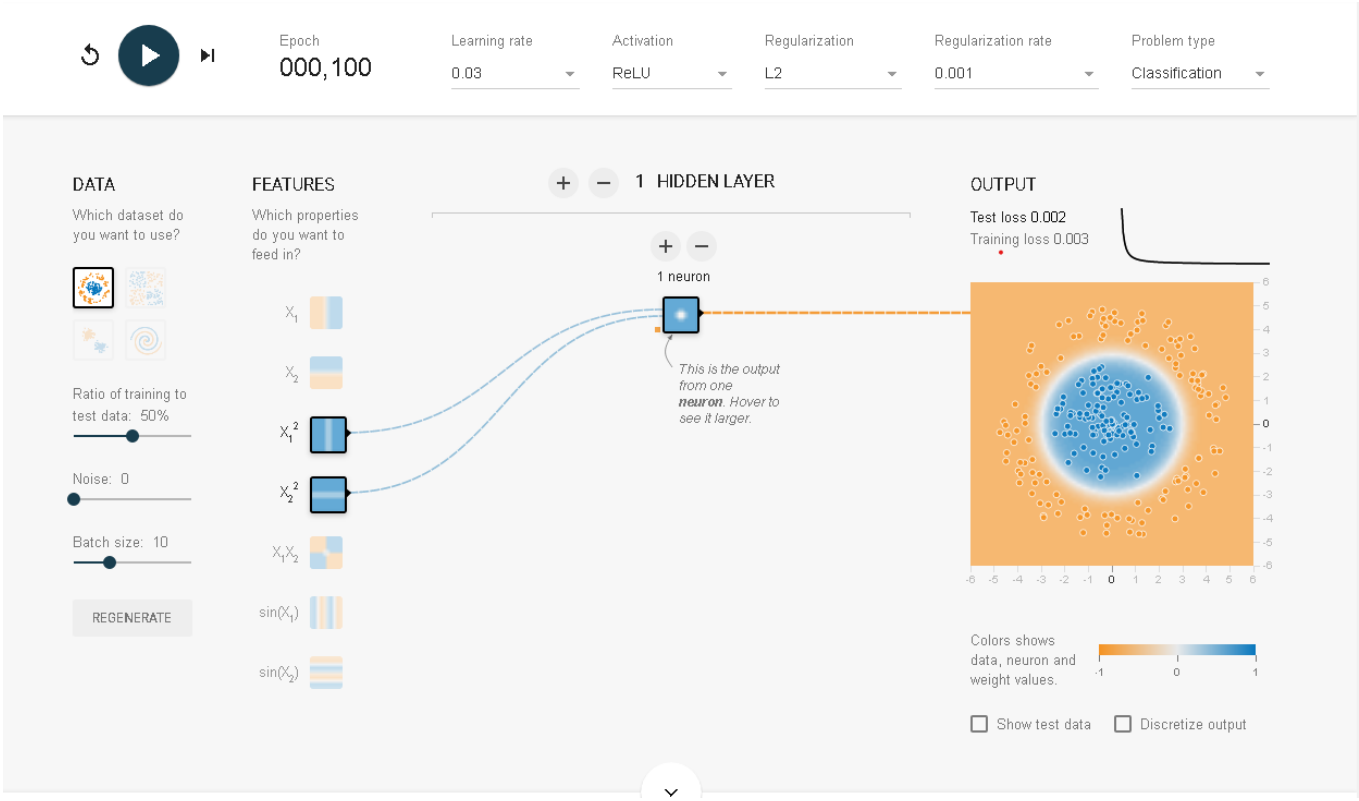
- Network Architecture: 2 layers each with 8 neurons
- Learning Rate: 0.03
- Activation Function: ReLU
- Regularization: L2
- Regularization Rate: 0.001



Result: 100 Epochs with 0.004 test loss(2 layers with 16 neurons)

> Now suppose you use features X1^2 and X2^2, instead of raw features X1 and X2. Your objective now is to obtain a test loss of less than 0.1 in as few neurons as possible. How many neurons would you need to classify the dataset? Explain and justify your answer.

- Network Architecture: 1 layers with 1 neuron
- Learning Rate: 0.03
- Activation Function: ReLU
- Regularization: L2
- Regularization Rate: 0.001

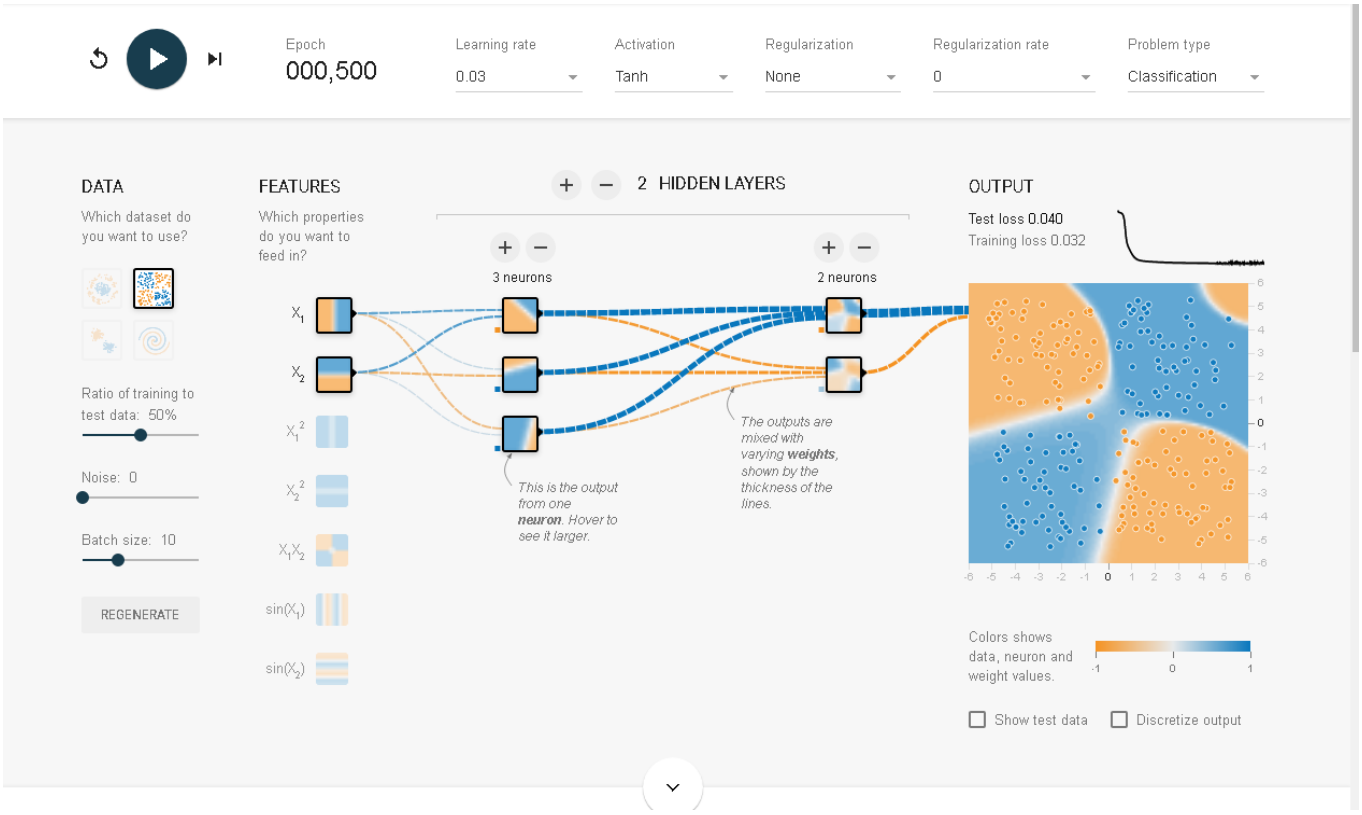


圓形數據集是一個二元分類問題，原本使用X1跟X2的狀況下，是線性不可分的，但如果通過使用特徵X1^2和X2^2將數據轉換為不同的空間，將每個點的座標平方，各點從原始空間轉換到新的空間中。在這個新的空間中，原本不可分的圓形模式變得可以用線性方式分開，因為距離原點更近的數據點更有可能屬於一個類別，而距離遠離原點的數據點更有可能屬於另一個類別。這使用線性分類器可以輕鬆地區分不同類別的點

Result: 100 Epochs with 0.002 test loss(1 layers with 1 neuron)

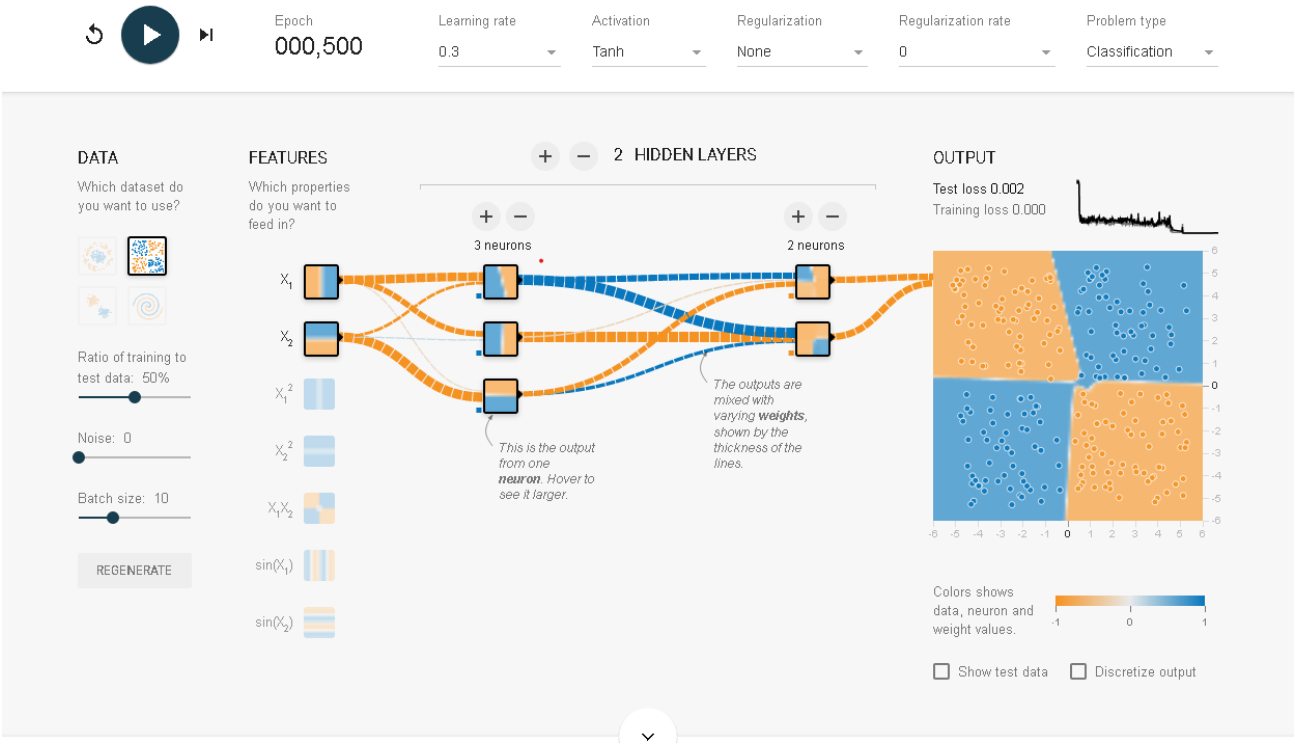
(B) Exclusive-Or data set:

> Consider the following method to train your neural network model with raw features X1, X2. Set the activation function to be tanh, use a learning rate of 0.03 and don't use any regularization. Use a neural network with two hidden layers with the first hidden layer having 3 neurons and the second layer having 2 neurons. Do you obtain a test loss of less than 0.1?

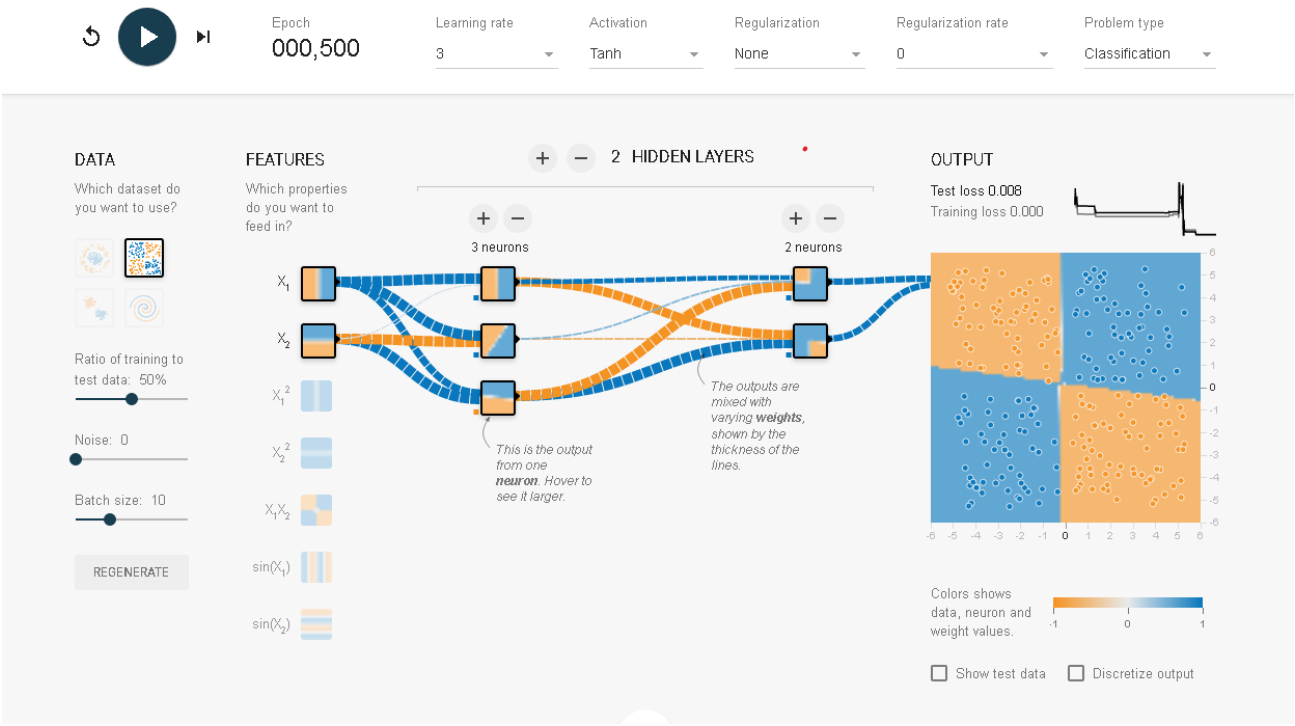


> Now change the learning rate to 0.3 and report the final test loss. Repeat the same for learning rate = 3 now. How does the test loss change as you increase the learning rate? Briefly explain why.

- learning rate: 0.3



- learning rate: 3

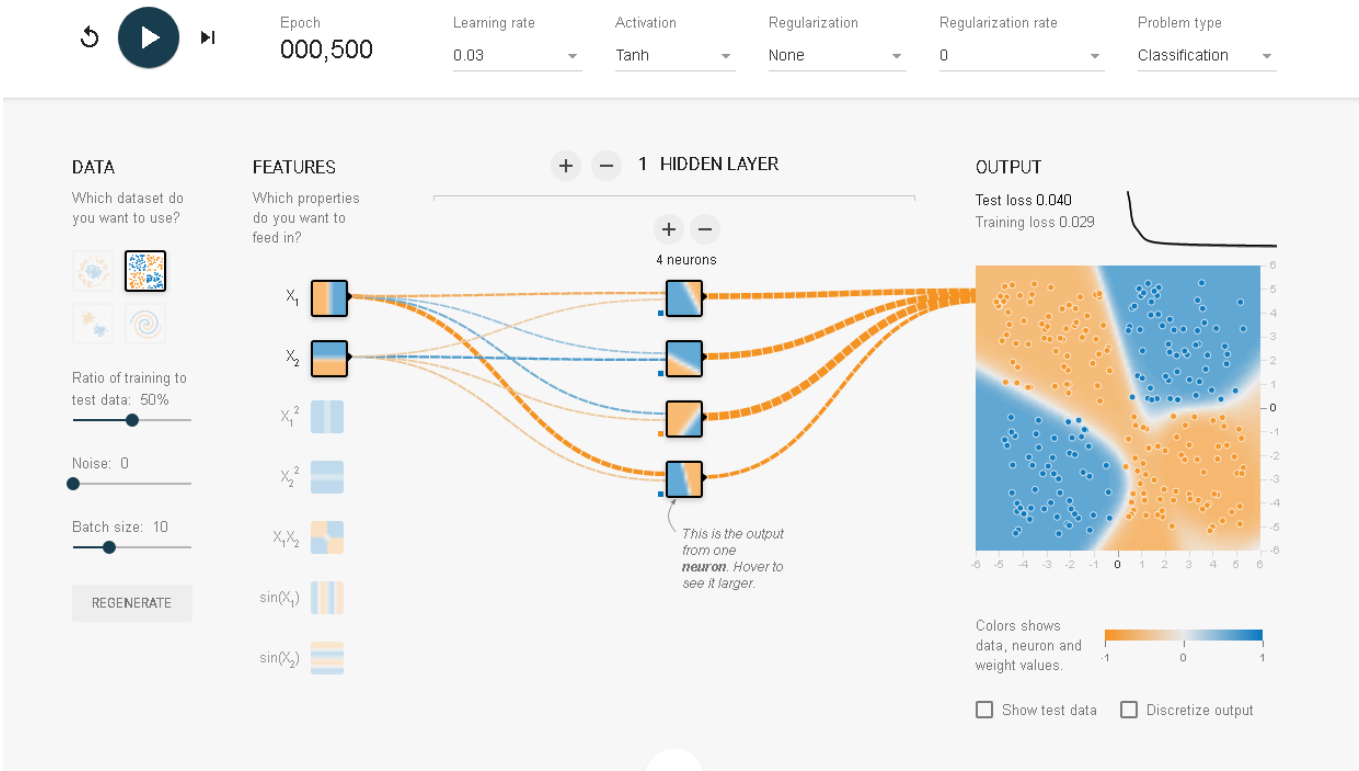


learning rat，代表每次迭代中模型向loss移動的步長，也就是適應該問題的速度，如果太低，找到好答案的時間就會花很久，太高，模型可能會無法收斂，找不到好的答案

> Find another network architecture. that also gives you a test loss of less than 0.1 on this dataset. Please share a screenshot of this network as described earlier.

- Network Architecture: 1 layers with 4 neurons
- Learning Rate: 0.03

- Activation Function: Tanh
- Regularization: None
- Regularization Rate: 0

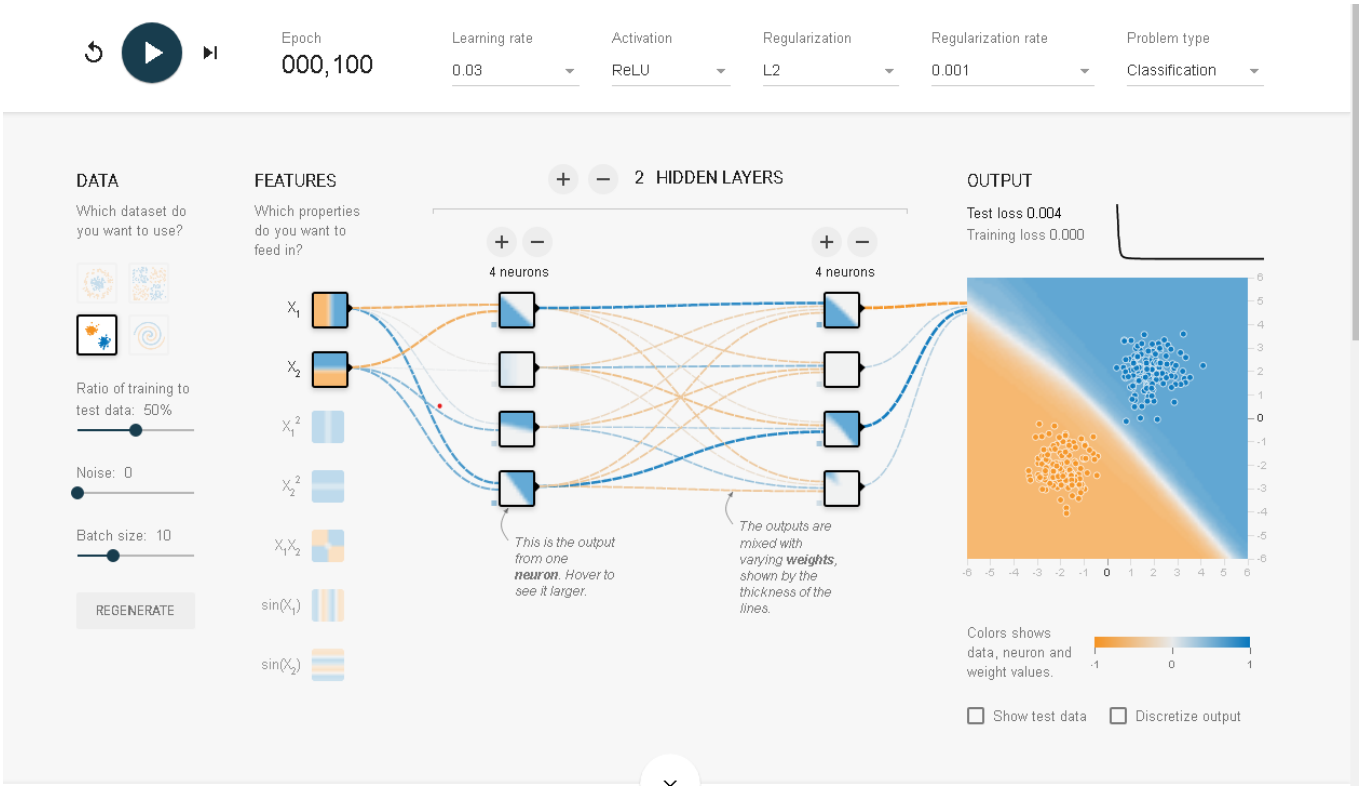


Result: 500 Epochs with 0.040 test loss(1 layers with 4 neurons)

(C) Gaussian data set:

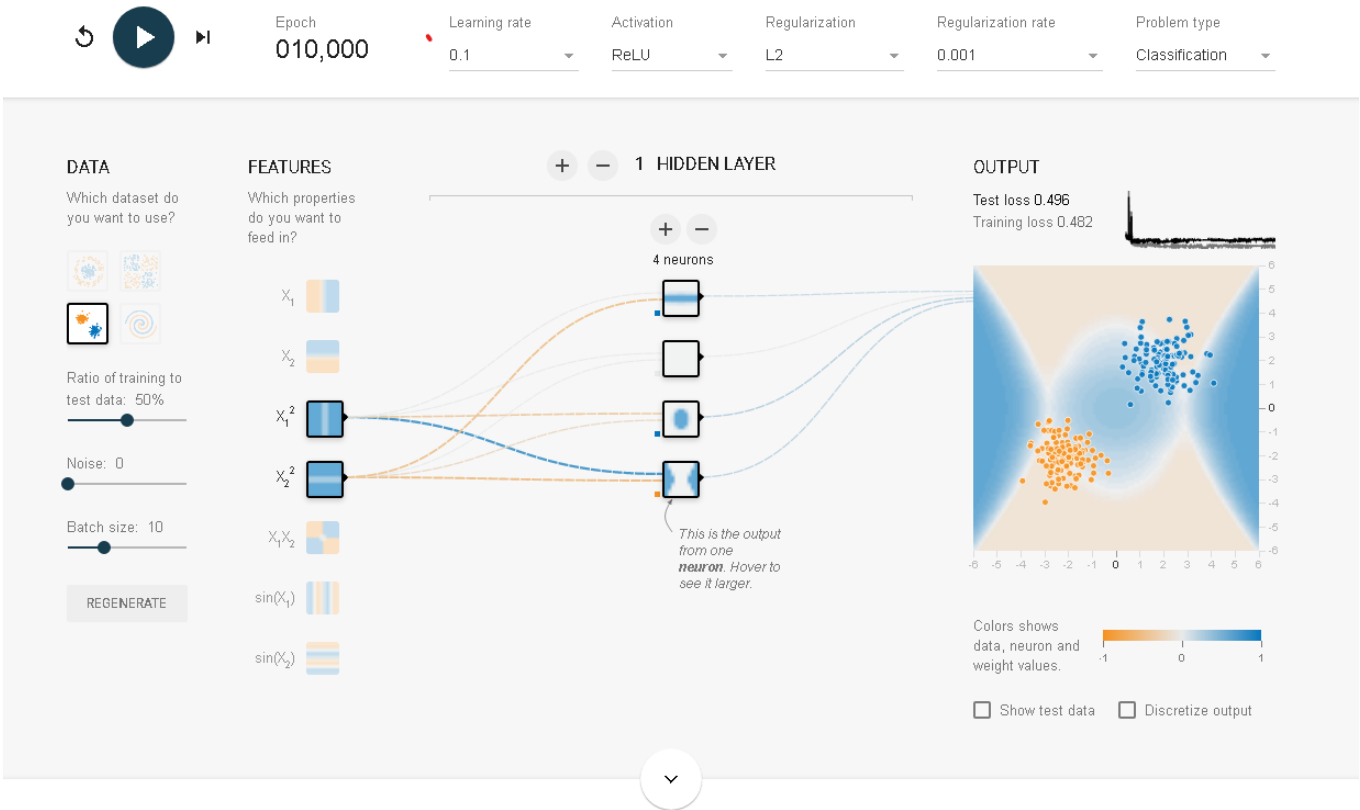
> Can you obtain test loss of less than 0.1 using only the raw inputs? i.e., using only X1 and X2? Try to use as few neurons as possible. Report the number of neurons used in your model and justify your answer.

- Network Architecture: 2 layers with each 4 neurons
- Learning Rate: 0.03
- Activation Function: ReLU
- Regularization: L2
- Regularization Rate: 0.001



Result: 100 Epochs with 0.004 test loss(2 layers with each 4 neurons)

> Now suppose you use features X1^2 and X2^2 instead of X1 and X2. Can you still obtain a test loss of less than 0.1? Justify your answer.



無法達到小於0.1 test loss

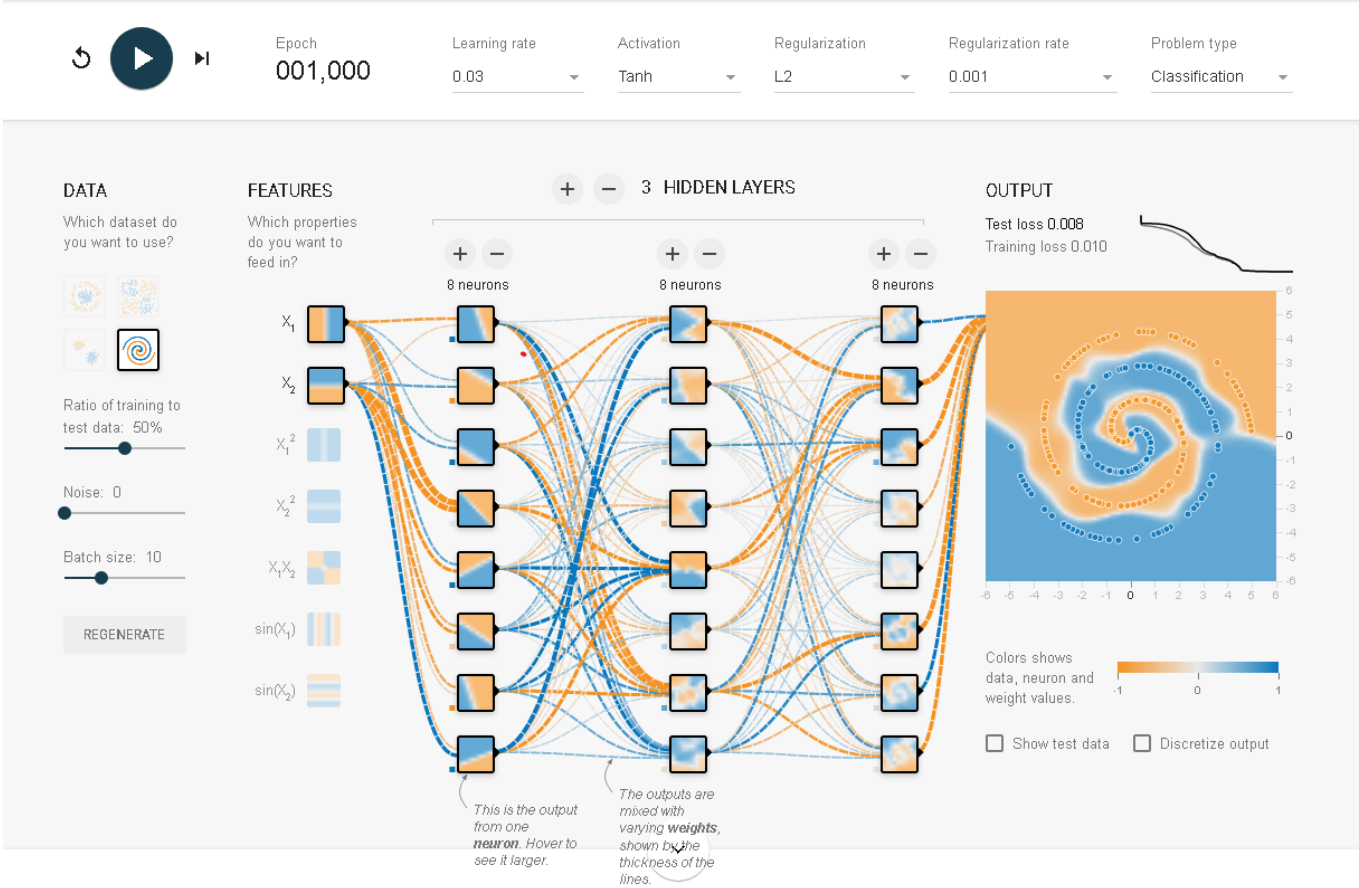
在 Gaussian data set 中，平方空間中，兩種不同類別的點離原點的距離是相近的，因此當特徵從（X1，X2）到（X1^2，X2^2）後，數據變得更難區分，兩個類別點的分布變的重疊

Result: 10000 Epochs with 0.496 test loss(1 layer with 4 neurons)

(D) Spiral data set:

> Can you obtain test loss of less than 0.1 using only the raw inputs? i.e., using only X1 and X2? Please share your screenshot of the network as described.

- Network Architecture: 3 layers with each 8 neurons
- Learning Rate: 0.03
- Activation Function: Tanh
- Regularization: L2
- Regularization Rate: 0.001

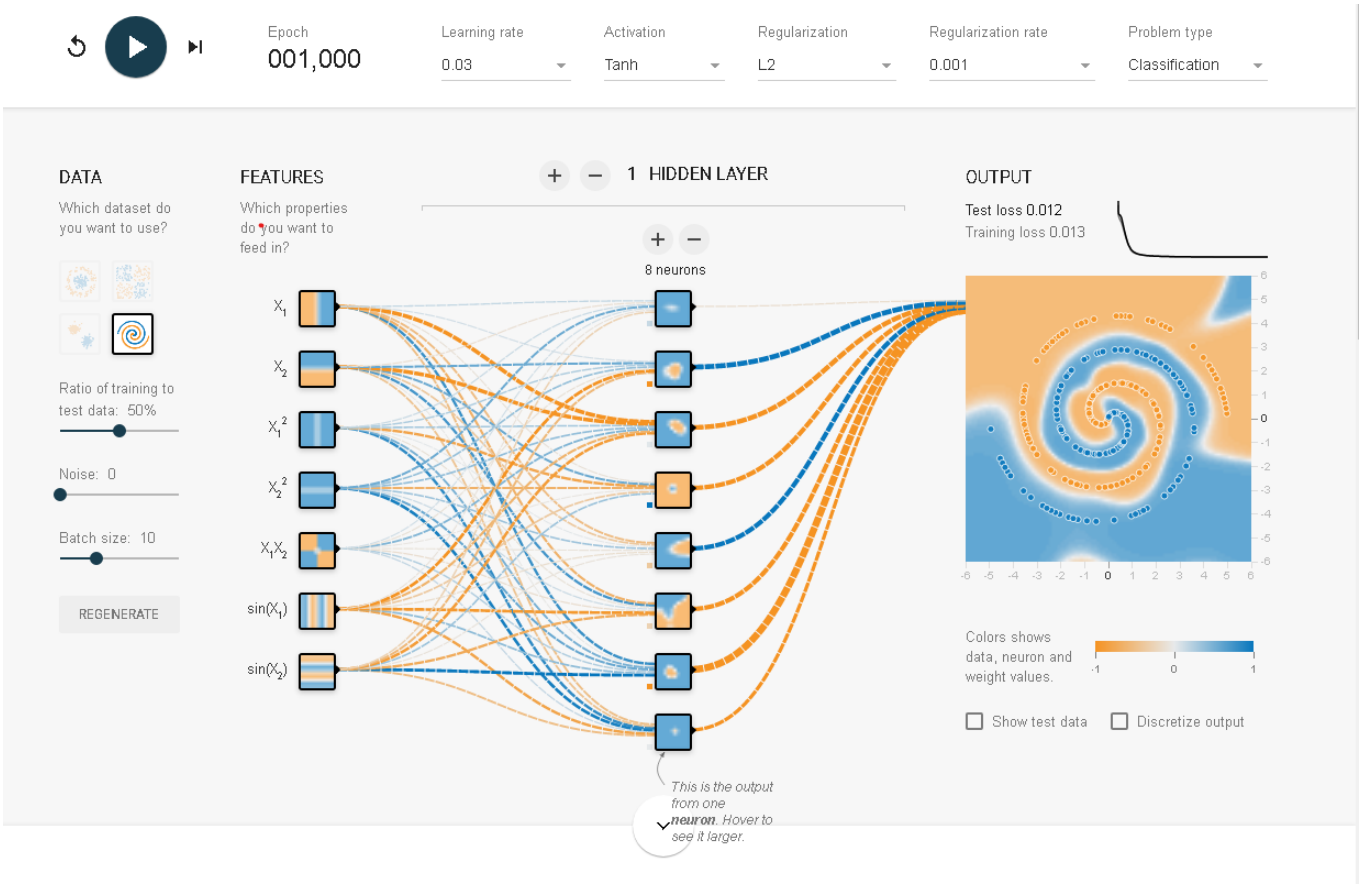


Result: 1000 Epochs with 0.008 test loss(3 layers with each 8 neurons)

> Now you are allowed to use any/all of the 7 different input features (X1, X2, X1^2, X2^2, X1X2, sin(X1), and sin(X2)). Can you obtain test loss of less than 0.1 for the Spiral data set with fewer neurons? Please share your screenshot of the network as described.

- Features: all 7 features

- Network Architecture: 1 layer with 8 neurons
- Learning Rate: 0.03
- Activation Function: Tanh
- Regularization: L2
- Regularization Rate: 0.001



Result: 1000 Epochs with 0.012 test loss(1 layer with 8 neurons)

(E) What is L1 regulations? Explain detailly.

L1 regulations，全名為Least Absolute Shrinkage and Selection Operator，是用來防止類神經網路過擬和的技術，會在test loss中增加懲罰，防止模型過度擬和訓練集

L1 regulations的特性

1. 特徵選擇：L1正則化最為突出的一個特點是，它可以產生稀疏解，其中許多權重正好為零。這是因為L1懲罰機制鼓勵權重趨於零，從而從模型中有效地刪除相應的特徵。在處理高維數據且懷疑許多特徵可能是不相關或多餘的情況下特別有用

2. 韌性： L1 regulations 可以幫助模型對訓練數據中的異常值更加韌性。這是因為它鼓勵模型將答案分散到多個特徵上，而不是過度依賴任何一個特徵

L1 regulations 與損失函數

> Total Loss = Original Loss + λ * Sum of absolute values of weights

(F) What is L2 regulations? Explain detailly.

L2 regulations，全名為可稱作 Ridge regularization，是通過懲罰模型中的大權重防止過擬合的方法

L2 regulations的特性

1. 縮減：L2 regulations 對 loss 添加一個與權重大小的平方成正比的懲罰項，減少大權重的出現。這鼓勵模型尋找權重較小的解，有助於防止過擬合
2. 非稀疏性：與L1不同，L2不會導致稀疏解。即使某些權重相對不重要，它們通常也不會被趨近為零，模型反而會在所有特徵上保持小的非零權重
3. 穩定性：因為L2正則化鼓勵小權重，它可以通過減少權重估計的變異性來幫助穩定模型。這可以使模型對數據的微小變化更加穩定

L2 regulations 與損失函數

> Total Loss = Original Loss + λ * Sum of squares of weights

L2 regulations是機器學習模型中防止過擬合的好用方法，特別是在處理高維數據或具有較多噪音的數據

(G) What is regulation rate? Explain detailly.

簡單來說就是上面兩個問題中的λ，是一種超參數，也是用來防止過擬和

regulation rate 可以有效地控制擬合數據和保持權重之間的平衡。較高的 regulation rate 意味著模型將更注重保持權重較小，潛在地犧牲對數據的擬合。相反，較低的 regulation rate 意味著模型將更注重對數據的擬合，潛在地以具有較大權重為代價

選擇適合的 regulation rate 通常涉及需要逐步實驗和調整，或者使用交叉驗證等技術，取決於具體的數據集和模型的複雜性。目標是找到一個平衡點，使模型在不過擬合的情況下很好地擬合數據，因此在新的、未見過的數據上使用舊的 regulation rate 會表現不佳

(H) What is Noise? Explain detailly.

Noise是理想情況下模型應該學會忽略的部分，Noise就是訓練集中不代表有意義的信息

Noise 來源

1. 測量誤差：沒有測量過程是完美的，測量過程中的小誤差會導致數據中出現 Noise
2. 固有變異性：即使沒有測量誤差，許多現象本身就是變異的。例如，即使是相同年齡和性別的人，身高也會有變化，這種變異性就是一種 Noise
3. 無關信息：有時，數據包含與當前任務無關的信息，這些無關信息可以作為 Noise。例如，如果你根據大氣壓力讀數來預測天氣，與你在第幾秒讀取這些數據無關，就屬於 Noise

(I) What is Batch size? Explain detailly.

Batch size 代表一次迭代中使用的訓練示例數量，每次迭代包括前向傳播、損失計算和反向傳播

Batch size 的選擇對訓練過程有重大影響：

1. 計算效率：較大的 Batch size 可以在計算上更高效，因為它們允許將計算在多個示例之間並行處理
2. 泛化能力：較小的 Batch size 可能會帶來更好的泛化性能（即對新的、未見過的數據的表現更好）。這可能是因為在每一步中使用較小的數據樣本引入的噪音可以作為 regultaion
3. 記憶體要求：較大的 Batch size 需要更多的記憶體，因為它們涉及一次處理更多的數據
4. 收斂：較小的 Batch size 有時可以導致更快地收斂到良好的解，因為它們涉及更頻繁地更新模型的權重

在實際應用中，最優的 Batch size 通常取決於特定的問題和可用的計算資源。為了提高計算效率，它通常被選擇為2的冪次方（32、64、128、256），也會作為一個超參數進行調整