**4. 딥러닝과 인공신경망**

(1) 딥러닝이란

머신러닝의 알고리즘에는 많은 종류가 있습니다. 그 중에서 인공신경망(Artificial Neural Network)을 기반으로 한 방법을 딥러닝(Deep Learning)이라고 합니다.

대표적인 딥러닝 인공지능으로는 구글의 딥마인드에서 개발한 알파고(AlphaGo)가 있습니다.

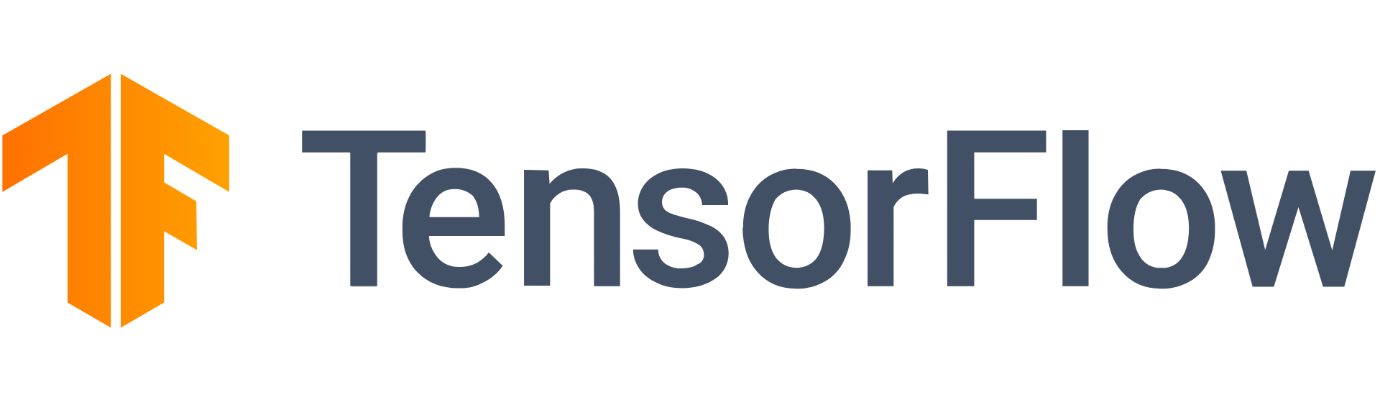
이세돌과의 대국으로 국내에 많이 알려진 알파고는 2014년부터 본격적으로 개발이 시작되어 2015~2017년엔 프로토타입 버전이 공개되고, 2017년 10월 최종 버전이 공개되었습니다



초기 딥러닝 알고리즘은 데이터 양의 부족, 부족한 컴퓨터 성능으로 인해 크게 주목받지 못했습니다.

하지만 최근에는 빅 데이터의 발전, 하드웨어의 발전(특히 GPU)으로 인해 딥러닝이 크게 발전하고 주목받고 있습니다.

이러한 딥러닝의 발전으로 2015년 구글은 딥러닝 라이브러리인 텐서플로(TensorFlow)를 오프소스로 공개했습니다. 물론 텐서플로 외에도 테아노(Theano), 파이토치(PyTorch) 등의 딥러닝 라이브러리도 있습니다.



그 중 우리가 사용하게 될 텐서플로는 공개와 동시에 큰 인기를 끌어 현재까지도 가장 널리 사용되는 딥러닝 라이브러리 중에 하나입니다.

이 라이브러리들은 모두 딥러닝 프로그램을 쉽게 구현할 수 있도록 다양한 기능을 제공해주는 라이브러리입니다.

(2) 인공신경망

위의 내용에서 딥러닝을 소개하며 짧게 인공신경망을 언급했습니다.

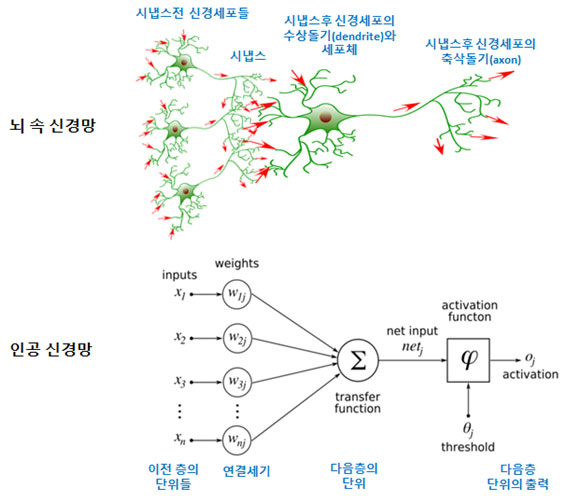
이번엔 이 인공신경망에 대해 더 자세히 알아보겠습니다.

지금까지 읽으면서 눈치채셨겠지만 인공지능을 만들기 위한 방법엔 여러가지가 있습니다.

그 중 인간의 뇌 동작 방식을 모방해서 기계가 학습할 수 있도록 고안된 알고리즘입니다.

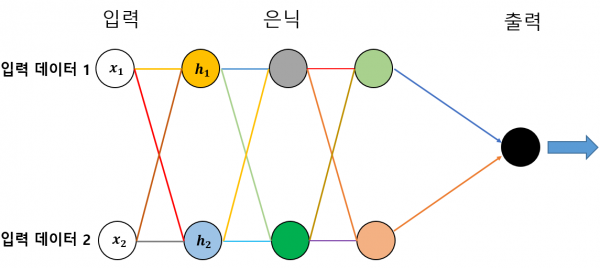
우리 두뇌에는 뉴런(Neuron)이라고 하는 세포들이 촘촘하게 연결되어 있고 이를 신경망이라 합니다.

인공 신경망은 이 뉴런이 서로 연결되어 있는 모습과 원리를 이용한 것입니다.



* 모양 뿐만 아니라 원리도 비슷합니다.

(3) 딥러닝의 원리



* 곧 비슷한 모양을 다시 보게 될 것입니다. 잘 기억해주세요

우선 그림을 봅시다. 이 그림에서 원 하나는 하나의 뉴런을, 뉴런을 연결하는 선은 데이터의 전달(좌 -> 우)을 나타냅니다. 이 인공 신경망으로 데이터 1, 2(각 x1, x2)가 입력된다고 합시다.

여기서 x1에 들어온 데이터 1은 h1, h2로 보내지며, x2에 들어온 데이터 2 역시 h1, h2로 보내집니다. 여기서 주목해야 할 점은 같은 데이터지만 다른 색으로 전달(다른 정보가 전달) 되었으며 전달받은 뉴런 역시 색이 각자 다릅니다(데이터를 처리한 값이 각각 다릅니다). 이렇게 데이터들이 계속 다른 정보로 변하면서 여러 뉴런을 거쳐 출력층으로 나아가게 됩니다.

이렇게 여러 레이어로 구성된 인공 신경망을 **심층 신경망**이라고 부르고, 이 심층 신경망을 이용해 학습과는 과정을 **딥러닝**이라고 부릅니다.

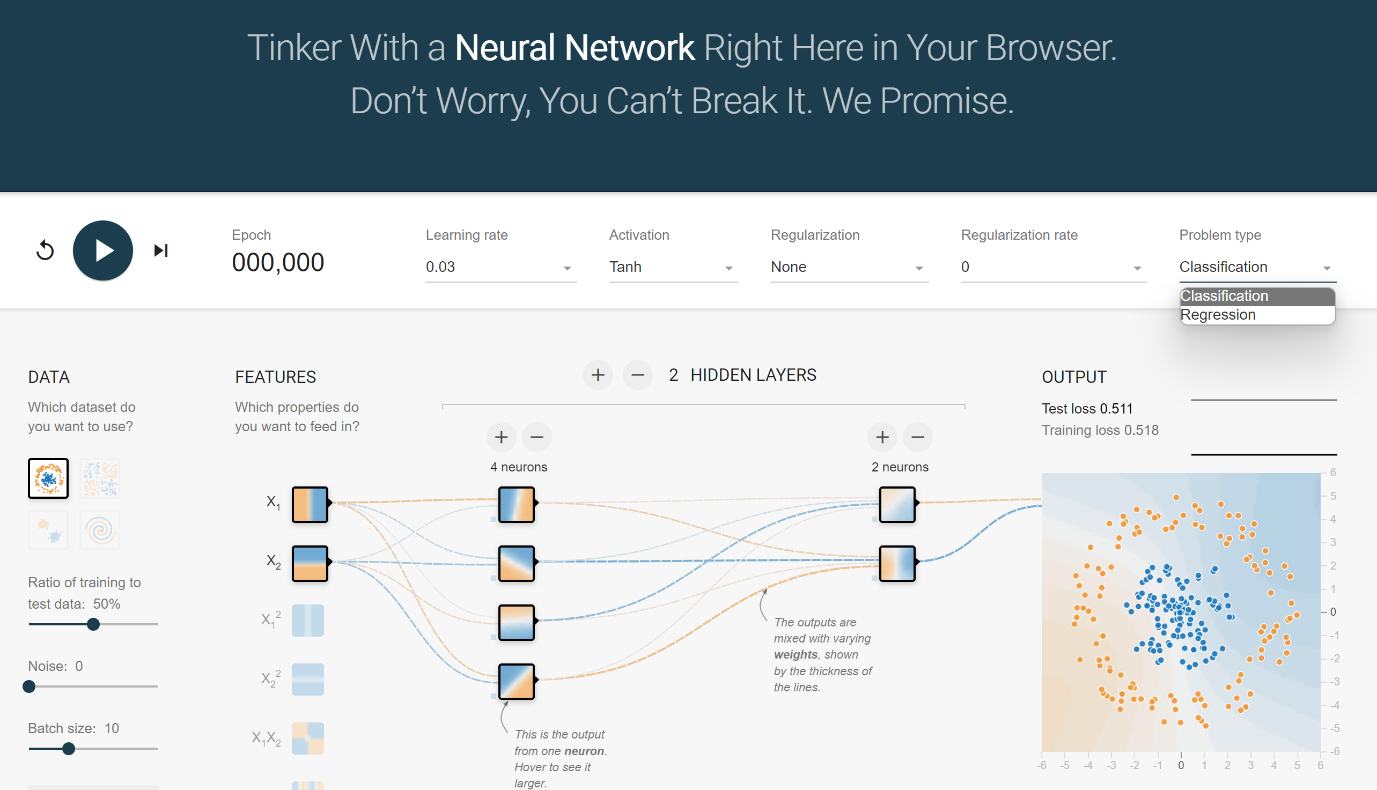
(4) 텐서플로 플레이그라운드로 딥러닝 체험하기

앞에서 살펴본 인공 신경망과 딥러닝의 개념으로는 이해가 어려울 수 있습니다.

그래서 개념을 넘어 체험을 해보겠습니다. 바로 딥러닝 놀이터인 텐서플로 플레이그라운드를 이용해서 말입니다.

1. 텐서플로 플레이그라운드 접속하기

- <https://playground.tensorflow.org/>

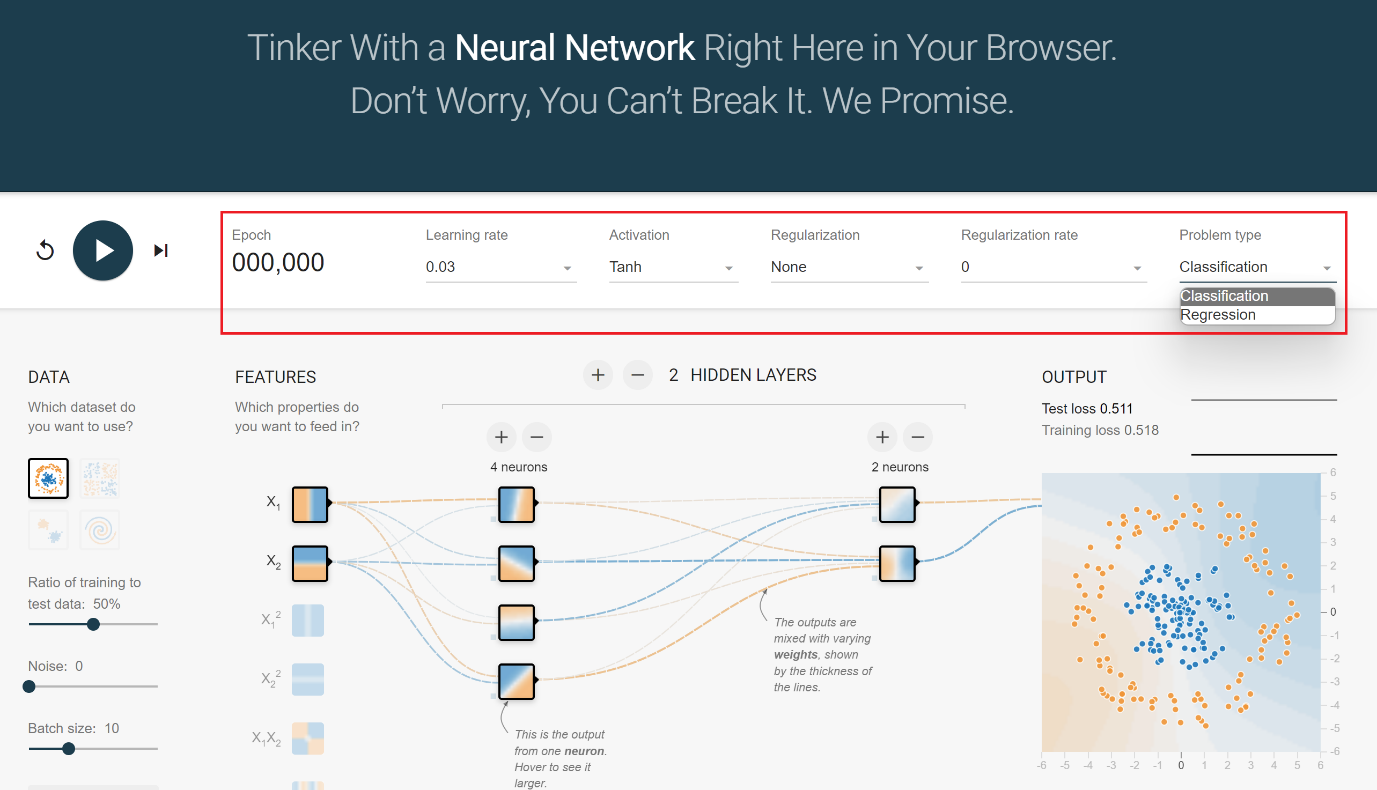


* 우리는 이것을 부술 수 없다는 문장이 보입니다. 저도 노력해봤지만 실패했습니다.

들어가면 굉장히 전문적으로 보이는 화면이 있습니다. 하지만 우리는 딥러닝의 원리에 대해 알고 있습니다.

위에서 본 심층신경망 그림과 비슷해 보이지 않나요? 그렇다면 잘 배우셨습니다.

대충 가지고 놀아도 되지만 일단 영어는 어려워 보이니 위쪽 메뉴부터 하나하나 한국어로 풀어보겠습니다.



Epoch (학습을 진행한 횟수) : 진행률을 의미합니다.

Learning rate (학습률) : 딥러닝에서는 정답 값과 예측 값의 오차를 줄이기 위해 가중치 값을 수정합니다. 이 가중치를 한 번에 얼마나 수정할지를 뜻합니다.

Activation (활성화 함수) : 하이퍼볼릭 탄젠트(tanh), 시그모이드(Sigmoid), 선형(Linear), 렐루(ReLU) 중 하나를 선택할 수 있습니다. 아직은 몰라도 됩니다.

Regularization (정규화) : 이전에 보지 못한 데이터를 예측하는데 실패하는 것을 줄이기 위한 것입니다.

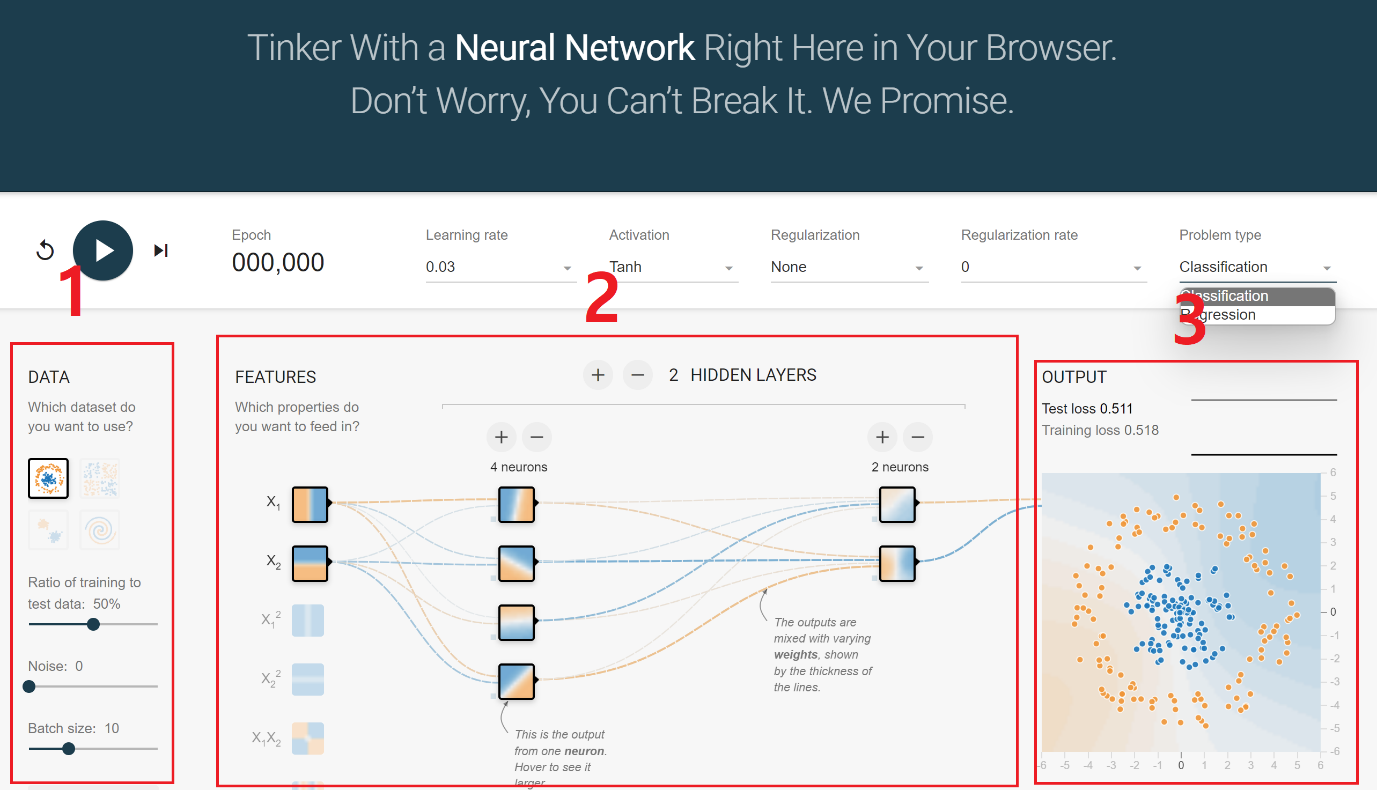
Regularization rate (정규화율) : 정규화할 때 어느 정도로 값을 수정할지 정해줍니다.

Problem type (문제 유형) : 문제를 분류(Classification) 혹은 회귀(Regression)로 선택할 수 있습니다.

분류 (Classification)는 데이터를 주황색과 파란색으로 분류하도록 학습시키는 것을 의미합니다

회귀 (Regression) 는 연속된 데이터의 값을 예측하도록 학습시키는 것을 의미합니다

슬슬 감이 오시나요? 이제 아래쪽 메뉴입니다.



1. 데이터 (DATA) : 학습할 데이터의 유형을 선택합니다.

1-1. 훈련 데이터에 대한 검증 비율 (Ratio of training to test data) : 데이터 중에서 얼마나 많은 양을 검증할 것인지를 의미합니다. 너무 많은 데이터가 들어오는 것도 좋지 않기 때문에 실제 딥러닝 인공지능에서는 20~30% 정도의 비율을 사용합니다.

1-2. 노이즈 (Noise) : 얼마나 분류가 덜 되었는지를 의미합니다.

1-3 배치 사이즈 (Batch size) : 한번에 얼마나 많은 양의 데이터를 처리할지를 의미합니다.

2. 신경망의 구조 설계

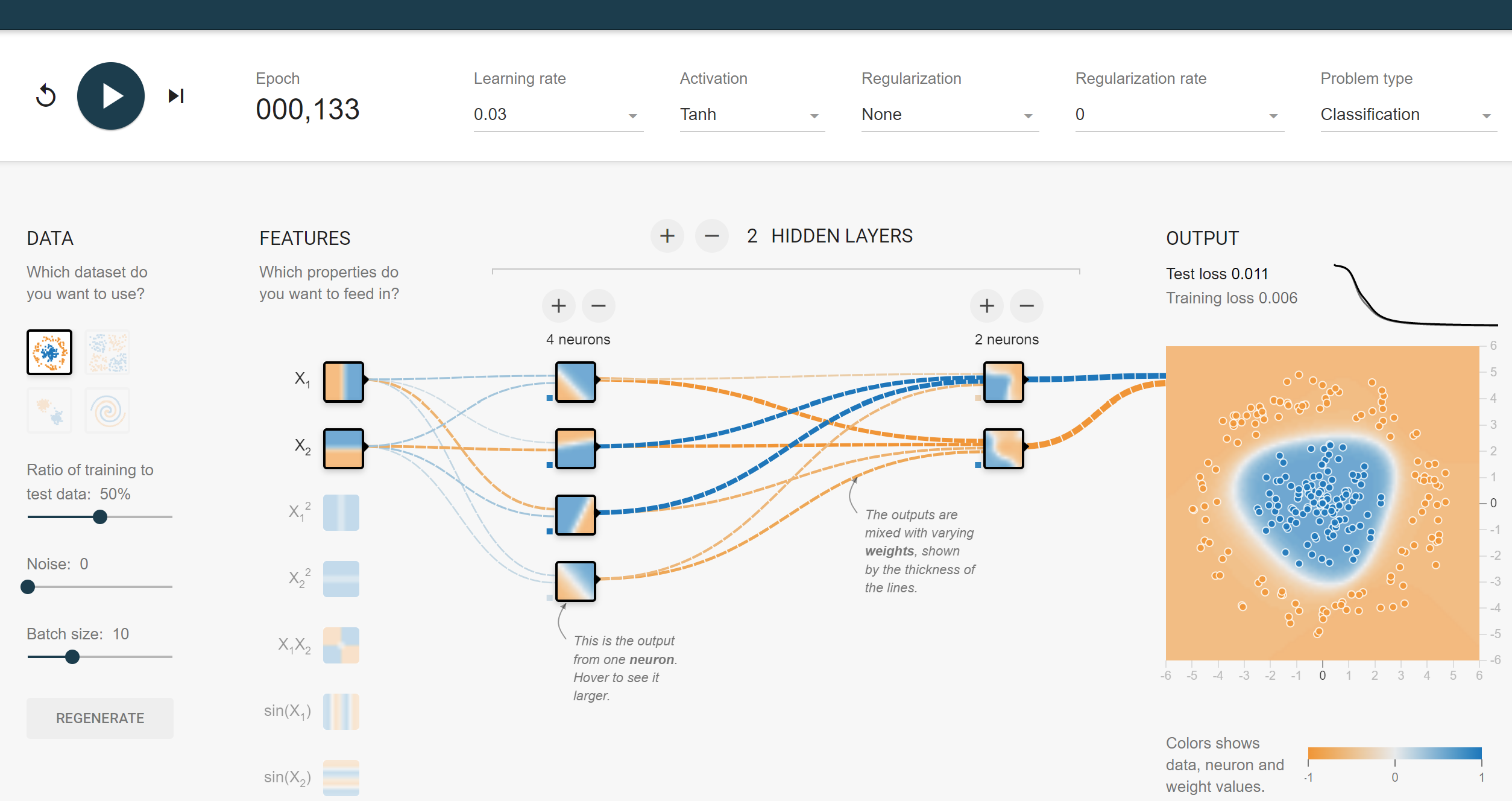
2-1. 특징 (FEATURES) : 초기 모양을 선택합니다. 학습이 진행되면서 이 모양들을 변형시키고 전달하며 1번에서 선택한 데이터의 모양과 비슷하게 맞춰갑니다.

2-2. 히든 레이어 (HIDDEN LAYERS) : 이전 내용인 딥러닝의 원리에서 보신 그림 기억나시나요? 은닉 층입니다. 얼마나 많은 단계를 거치며 분류할지를 결정합니다.

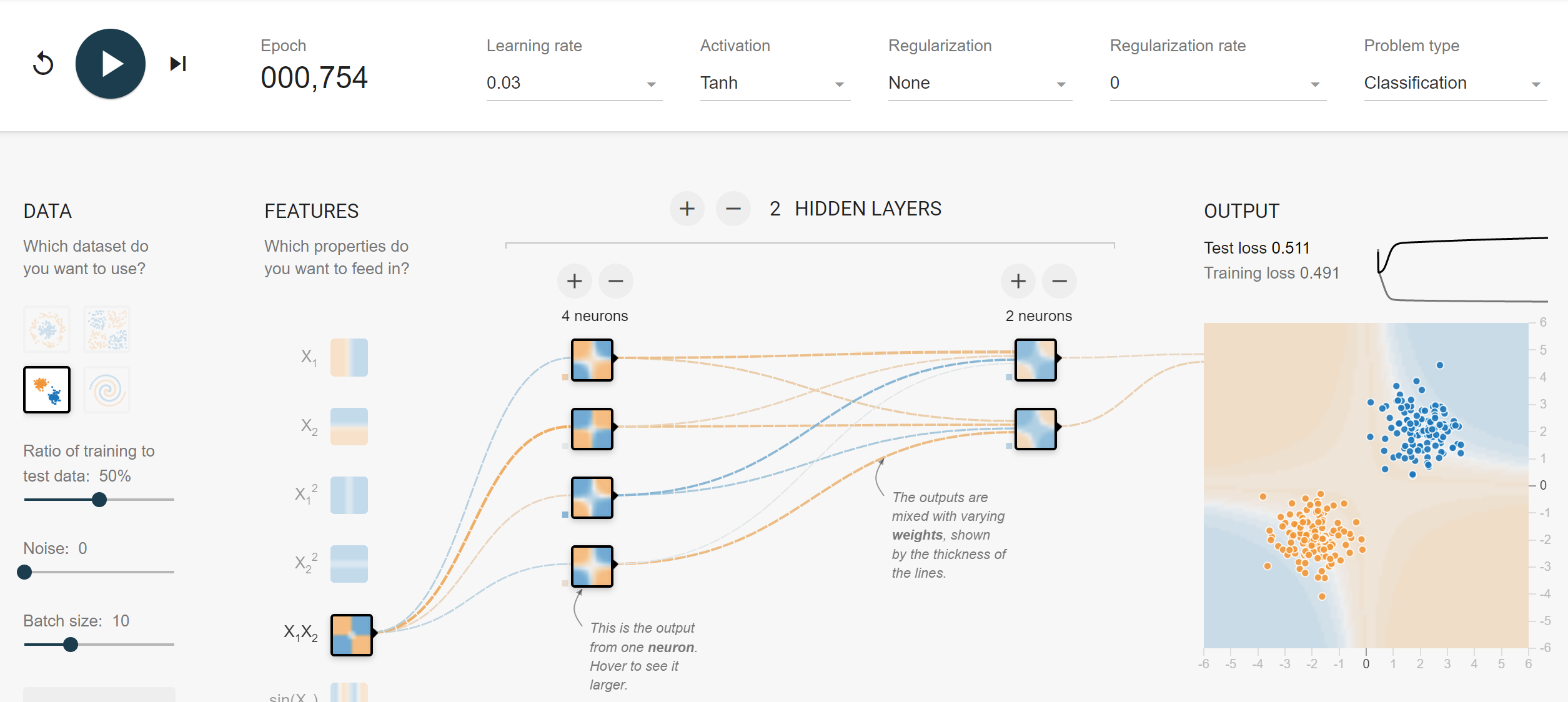
2.3 뉴런 (neurons) : 인간의 뉴런과 비슷한 역할입니다. 뉴런이 많을수록 다양한 패턴으로 이어질 확률이 높지만 덩달아 이상한 길로 빠질 확률도 존재합니다.

3. 출력 (OUTPUT) : 딥러닝 인공지능이 공들을 색깔별로 분류한 결과입니다.

일단 좌상단의 재생버튼을 눌러 실행시켜 봅시다.



처음 주어진 데이터는 파란색 점들을 주황색 점들이 둘러싸고 있는 구조였으며, 이 딥러닝 인공지능이 이 패턴을 잘 학습하여 적절하게 분류하는 과정과 그 결과를 볼 수 있습니다

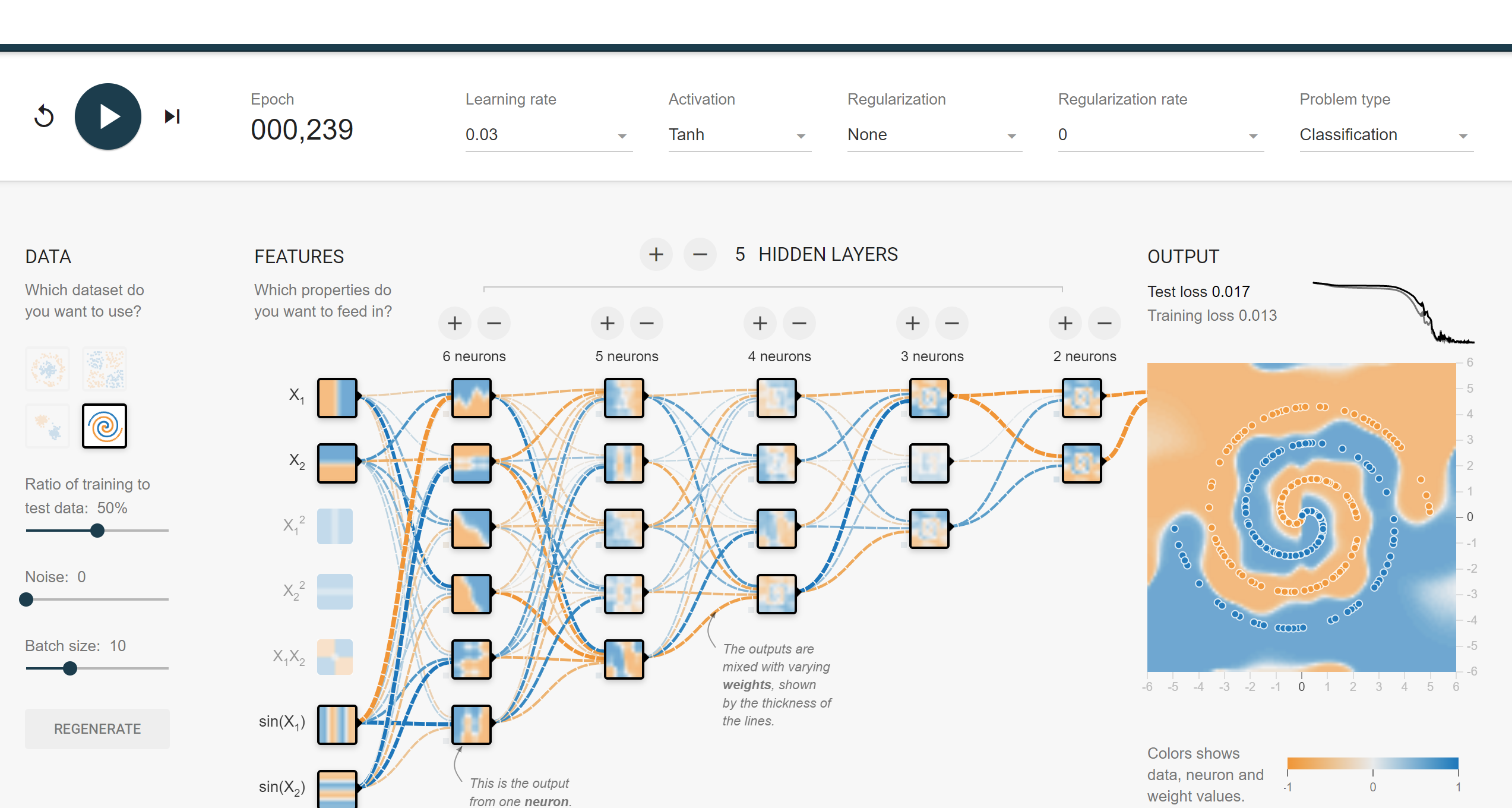


그런데 이 때 데이터와 맞지 않는 FEATURES를 고르자 분류에 어려움을 겪는 모습입니다.

이렇게 인공지능은 얼마나 많은 데이터를 얼마나 오래 넣고 얼마나 많은 과정을 거치는지 보다

어떤 데이터를 넣을지, 얼마나 적정한 층과 뉴런을 설정할지를 고르는 것이 중요하다는 것을 알 수 있습니다.

여러가지 훈련을 시켜보며 데이터의 중요성, 층과 뉴런에 대한 감을 잡아 보시기를 바랍니다.



우리가 인공지능 개발자의 몫을 잘 수행한다면 복잡한 데이터도 빠르게 구분할 수 있는 딥러닝 인공지능을 만들 수 있을 것입니다.