

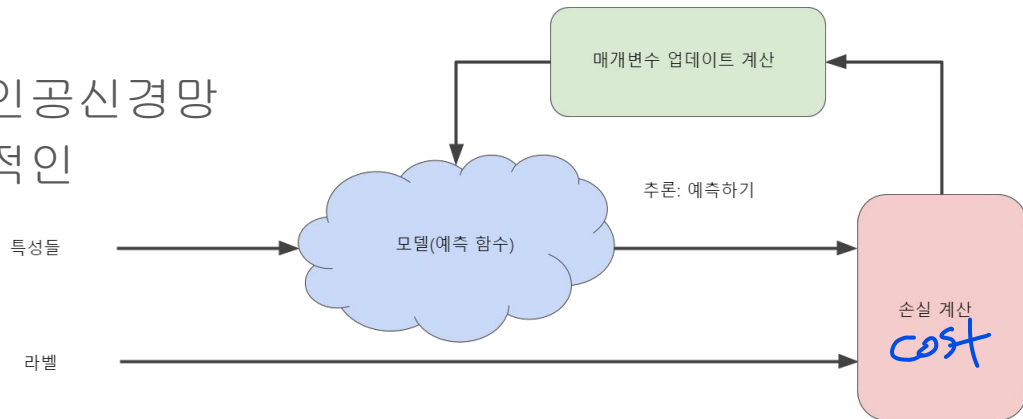
인공신경망 실습

using Neural Network Playground

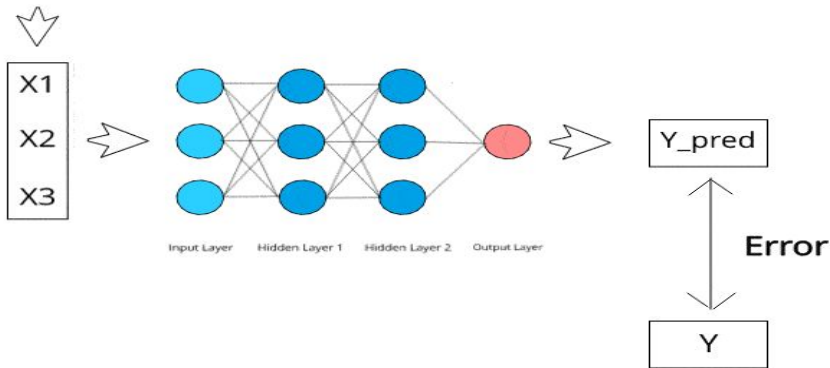
손실(비용) 줄이기: 반복 방식

cost

다음 그림은 머신러닝 알고리즘이 인공신경망 모델을 학습하는 데 사용하는 반복적인 시행착오 과정을 보여줍니다.



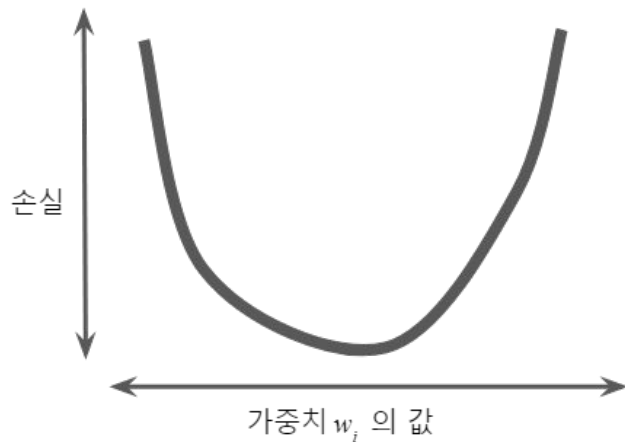
Feed new data



이 '모델'은 하나 이상의 특성 (x_1, x_2, x_3)을 입력하여 하나의 예측 (y_{pred})을 출력합니다.

손실 줄이기: 경사하강법(gradient descent)

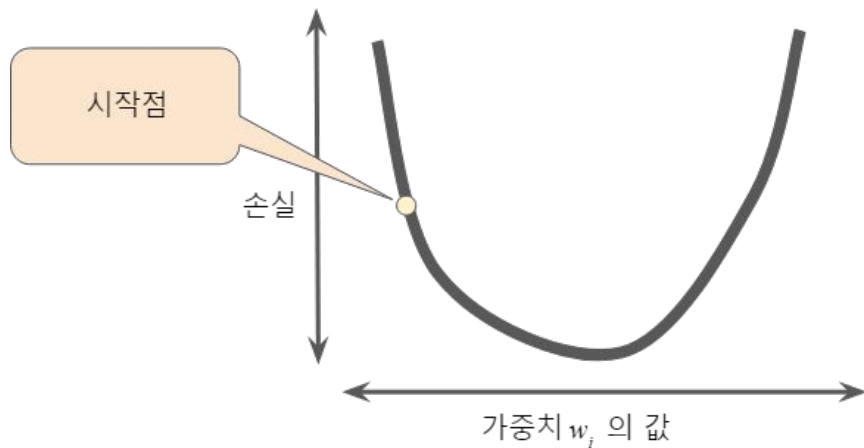
모델의 가능한 모든 값에 대해 손실을 계산할 시간과 컴퓨팅 자료가 있다고 가정합니다. 손실과 모델의 가중치들을 대응한 도표는 항상 **볼록 함수(convex function)** 모양을 할 것입니다. 즉, 도표가 다음과 같이 항상 **그릇 모양(bowl shape)**으로 나타납니다.



볼록 문제에는 기울기가 정확하게 0인 지점인 최소값이 하나만 존재합니다. 이 최소값에서 손실 함수가 수렴합니다. 전체 데이터 세트에 대해 상상할 수 있는 모든 값의 손실 함수를 계산하는 것은 수렴 지점을 찾는 데 비효율적인 방법입니다. 머신러닝에서 널리 사용하는 더 나은 방법인 **경사하강법**을 살펴보겠습니다.

손실 줄이기: 경사하강법

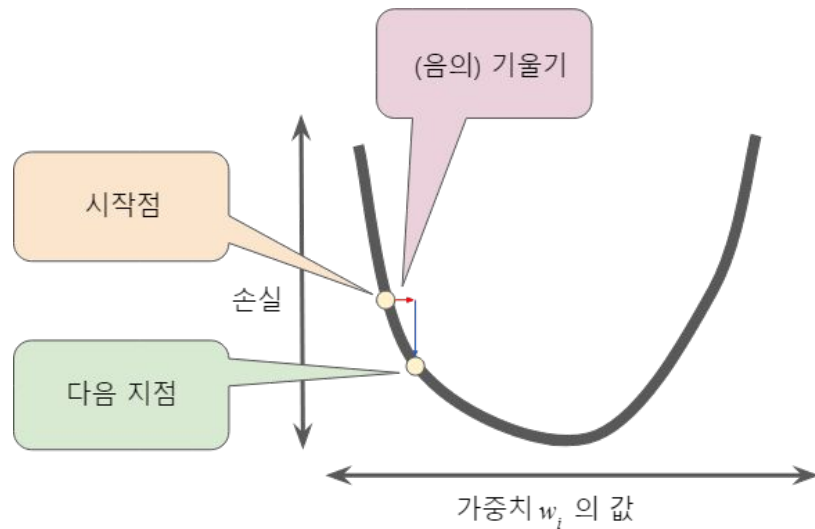
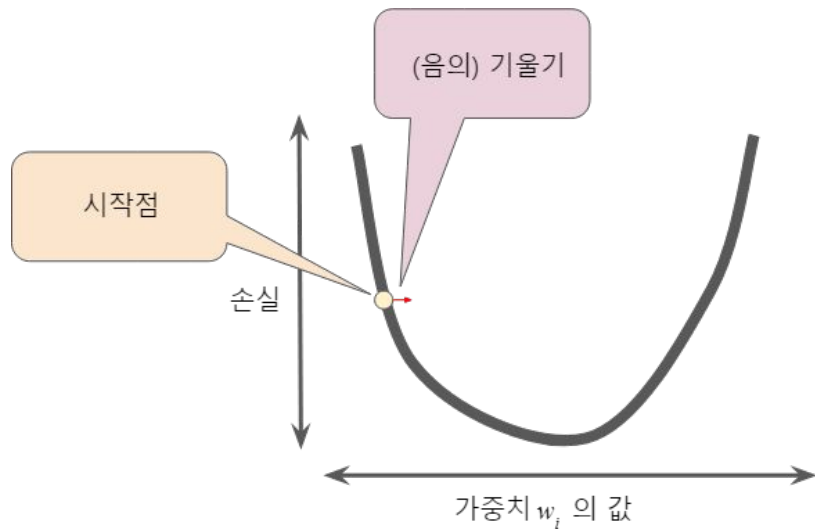
경사하강법의 첫 번째 단계는 가중치에 대한 시작값(시작점)을 선택하는 것입니다. 시작점은 별로 중요하지 않습니다. 따라서 많은 알고리즘에서는 0으로 설정하거나 임의의 값을 선택합니다. 다음 그림에서는 0보다 조금 큰 시작점을 지정했습니다



그런 다음 경사하강법 알고리즘은 시작점에서 **손실 곡선의 기울기를 계산**합니다. 간단히 설명하자면 기울기는 편미분의 벡터로서, 어느 방향이 '더 정확한지' 혹은 '더 부정확한지' 알려줍니다.

손실 줄이기: 경사하강법

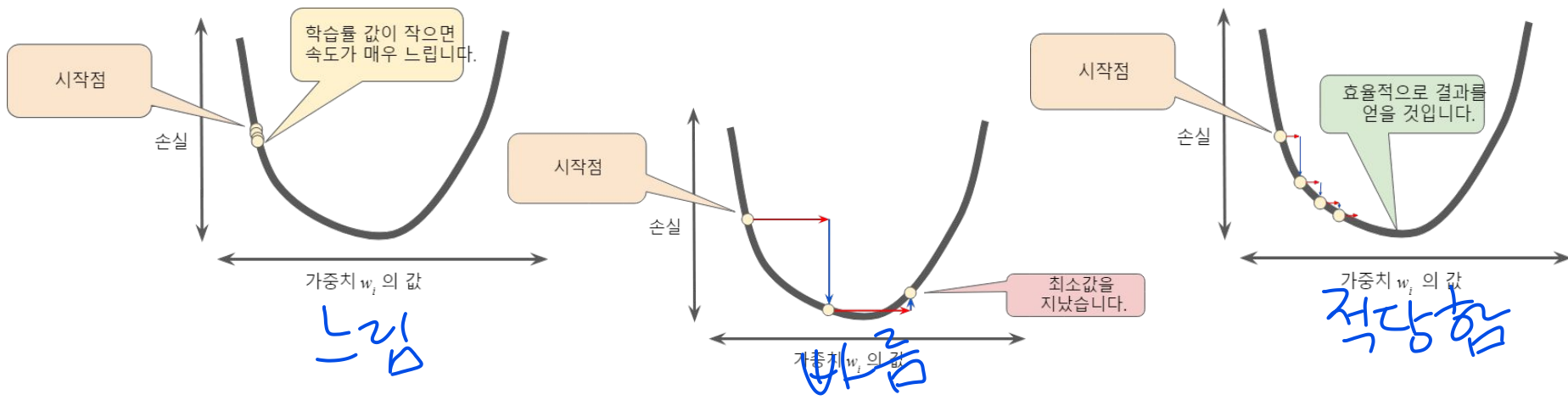
그림에 나온 것과 같이 단일 가중치에 대한 손실의 기울기는 미분 값과 같습니다. 기울기는 항상 손실 함수 값이 가장 크게 증가하는 방향을 향합니다. 경사하강법 알고리즘은 가능한 한 빨리 손실을 줄이기 위해 기울기의 반대 방향으로 가중치값을 이동합니다.



학습률(Learning Rate) 얼마나 이동시킬 것인가

경사하강법 알고리즘은 기울기에 학습률(learning rate) 또는 보폭이라 불리는 스칼라를 곱하여 다음 지점을 결정합니다. 예를 들어 기울기가 2.5이고 학습률이 0.01이면 경사하강법 알고리즘은 이전 지점으로부터 0.025 떨어진 지점을 다음 지점으로 결정합니다. 2.5×0.01

초매개변수(Hyper Parameter)인 학습률은 프로그래머가 머신러닝 알고리즘에서 조정하는 값입니다. 대부분의 머신러닝 프로그래머는 학습률을 미세 조정하는 데 상당한 시간을 소비합니다. 학습률을 너무 작게 설정하면 학습 시간이 매우 오래 걸릴 것입니다.



학습률(Learning Rate)

학습률이
너무 크면
발산하여
더 멀어짐

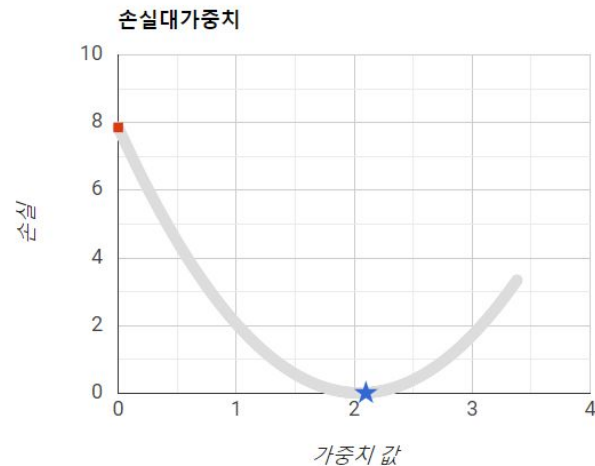
학습률	수렴 단계	<u>수렴</u> 여부
0.03		
0.06		
0.1		
0.3		
0.8		
1.2		
2.1		
2.6		
3.5		

다양한 학습률로 실험하고, 이러한 학습률이 손실 곡선의 최저점에 도달하는 데 필요한 단계 수에 어떤 영향을 미치는지 확인합니다. 그래프 아래에서 실험해 보세요.

학습률 설정:

한 단계 실행: 0

그래프 재설정:



[Link to Learning rate practice](#)

손실 줄이기: 배치 vs 확률적 경사하강법



경사하강법에서 배치(Batch)는 단일 반복에서 기울기를 계산하는 데 사용하는 데이터의 총 개수입니다. 지금까지는 배치가 전체 데이터 세트라고 가정했습니다. 배치가 너무 커지면 단일 반복으로도 계산하는 데 오랜 시간이 걸릴 수 있습니다.



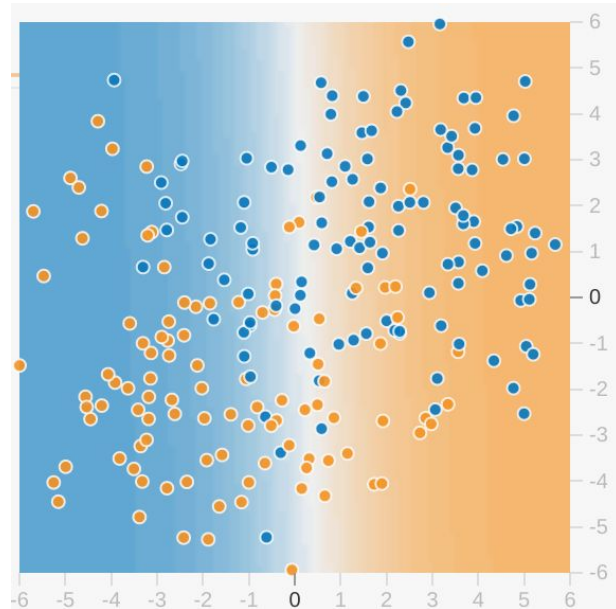
• 확률적 경사하강법(Stochastic GD)은 반복당 하나의 데이터(배치 크기 1)만을 사용합니다. 반복이 충분하면 SGD가 효과는 있지만 노이즈가 매우 심합니다. '확률적 (Stochastic)'이라는 용어는 각 배치를 포함하는 하나의 예가 무작위로 선택된다는 것을 나타냅니다.



• 미니 배치 확률적 경사하강법(미니 배치 SGD)은 전체 배치 반복과 SGD 간의 절충안입니다. 미니 배치는 일반적으로 무작위로 선택한 10개에서 1,000개 사이의 예로 구성됩니다. 미니 배치 SGD는 SGD의 노이즈를 줄이면서도 전체 배치보다는 더 효율적입니다. ⇒ [NN Playground](#) 에서 다양한 배치사이즈로 실습해 보십시오.

손실 줄이기: NN 플레이그라운드 실습

NN 플레이그라운드는 머신러닝의 원리를 알려주는 프로그램입니다. 각 플레이그라운드 실습에서는 데이터 세트를 생성합니다. 이 데이터 세트의 라벨에는 두 가지 값이 가능합니다. 가능한 두 가지 값을 스팸 대 스팸 아님 또는 건강한 나무 대 병든 나무로 생각할 수 있습니다. 대부분의 실습에서는 **다양한 초매개변수(hyper-parameter)를 조정**하여 각 라벨 값을 서로 올바르게 분류(분리 또는 구분)하는 모델을 만드는 것을 목표로 합니다. 단, 대부분의 데이터 세트는 일정 분량의 노이즈를 포함하므로 모든 예를 올바르게 분류하는 것은 불가능합니다.



학습률이나 배치사이즈가 초매개변수?

손실 줄이기: 플레이그라운드 실습

학습 손실: 비용함수의 값
단계가 진행될수록 감소

이 실습의 인터페이스에는 다음과 같은 3개의 버튼이 있습니다.

재설정: ↺ 반복을 0으로 재설정합니다. 모델이 이미 학습한 가중치를 모두 재설정합니다.

단계 : ▶ 반복을 한 단계 진행합니다. 반복할 때마다 모델은 미묘하게 변화하거나 크게 변화합니다.

재생성 : 새 데이터 세트를 생성합니다. 반복을 재설정하지 않습니다.



[Link to NN Playground - Learning Rate](#)

손실 줄이기: 플레이그라운드 실습

작업 1:

- 플레이그라운드 오른쪽 상단의 **학습률 메뉴**를 확인합니다. 학습률이 3으로 매우 높게 지정되어 있습니다.
- '단계' 버튼을 10~20번 정도 클릭하여 높은 학습률이 모델에 어떠한 영향을 주는지 살펴봅니다. 초기 반복을 진행할 때마다 모델이 시각적으로 얼마나 급격히 변화하는지 확인합니다. 모델이 수렴한 것처럼 보인 후에도 불안정성이 나타날 수 있습니다.
- 또한 x1 및 x2에서 모델 시각화로 이어지는 선을 확인합니다. 이 선의 두께는 모델에서 해당 특성의 가중치를 나타냅니다. 즉, 선이 두꺼우면 가중치가 높은 것입니다.

손실 줄이기: 플레이그라운드 실습

작업 2:

- 재설정 버튼을 누릅니다.
- 학습률을 낮춥니다.
- 단계 버튼을 여러 번 누릅니다.

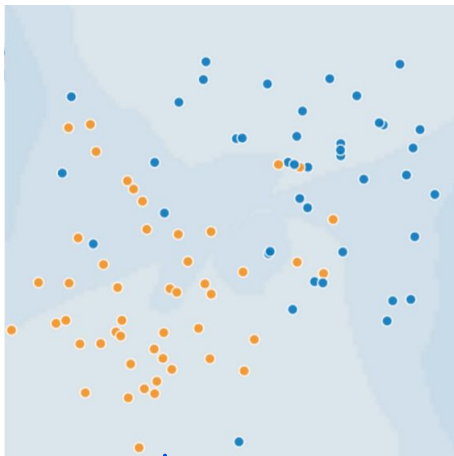
학습률을 낮추면 수렴에 어떠한 영향이 있나요? 모델이 수렴하는 데 필요한 단계 수 및 모델이 얼마나 부드럽고 안정적으로 수렴되는지를 살펴봅니다. 학습률을 이보다 더 낮춰서 실험해 봅니다. 학습률이 얼마나 낮아지면 유용성이 사라지나요?

ANS : 플레이그라운드 실습은 비결정성을 가지므로 사용자의 데이터 세트에 정확히 부합하는 답을 항상 제공할 수는 없습니다. 그러나 여기에서는 학습률 0.1에서 효과적으로 수렴이 이루어집니다. 학습률을 더 낮추면 수렴에 걸리는 시간이 훨씬 더 길어집니다. 즉, 학습률이 이보다 낮으면 속도가 너무 느려서 유용성이 사라집니다.

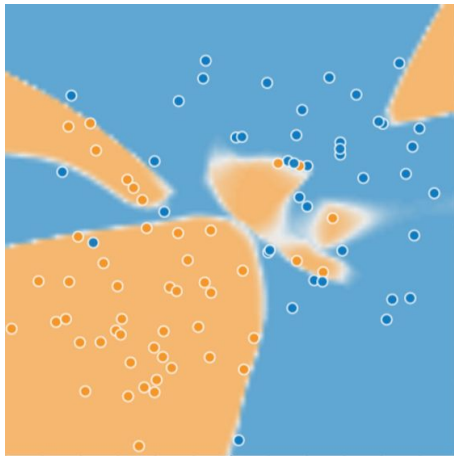
일반화 (Generalization)

그림의 각 점은 숲에서 나무의 위치를 나타낸다고 가정 : 병든(파란색) / 건강한(주황색) 나무

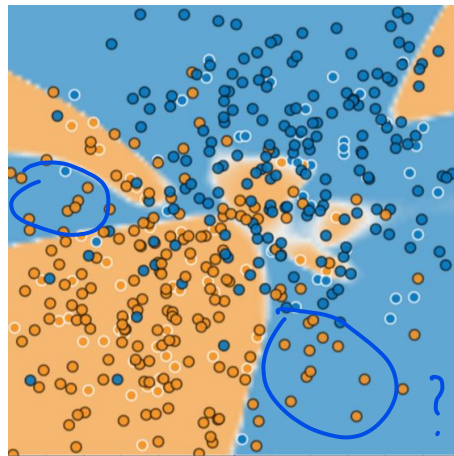
[샘플 자료]



[학습결과]



[새 자료의 결과]



훈련용 데이터에만 적합

특성

모델은 학습한 데이터의 특성에 **과적합(Overfitting)**되었습니다. 과적합 모델은 학습하는 동안 손실이 적지만 새 데이터를 잘 예측하지 못합니다.

일반화 (Generalization)

일반화는 모델이 (이를 만들기 위해 사용된 것과 같은 분포에서 추출된) 이전에 보지 못했던 **새로운 데이터에 제대로 적합할 수 있는** 지를 나타냅니다.

머신러닝 모델이 덜 복잡할수록 (샘플의 특성 때문이 아닌) 더 좋은 경험적 결과를 얻을 가능성이 높습니다.

모델이 테스트 세트에서 효과적으로 동작하면 다음과 같은 경우 새로운 데이터에도 잘 동작할 것이라고 추정할 수 있음

- 테스트 세트가 충분히 큰 경우
- 테스트 세트를 반복적으로 사용하지 않는 경우



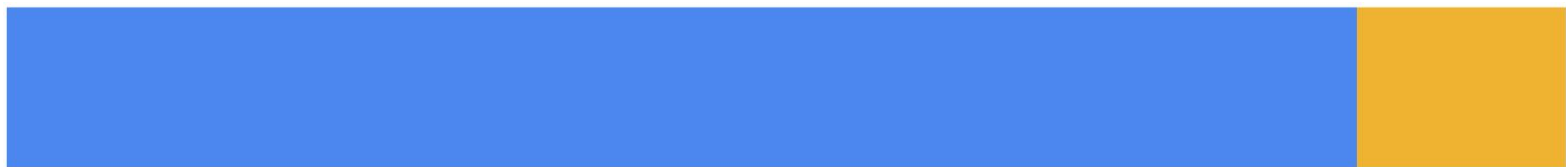
학습 및 평가 자료

- **훈련/학습 자료 (training data) :**

모델을 학습시키기 위한 데이터 세트의 일부분 (~70/80%)

- **시험/평가 자료 (testing data):**

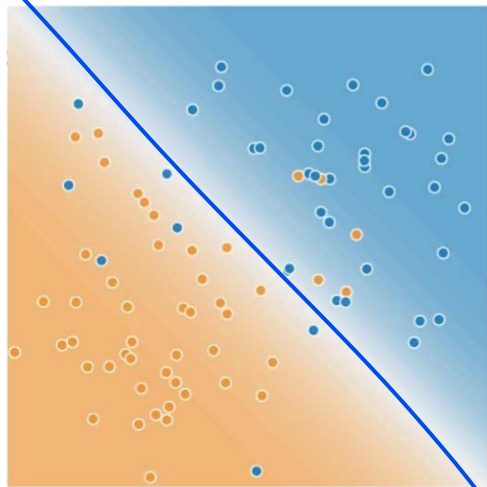
모델을 테스트하기 위한 데이터 세트의 일부분 (~30/20%)



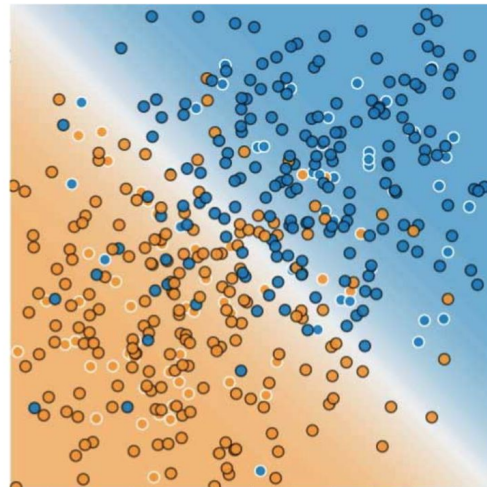
학습 세트

테스트 세트

- 각 **파란색** 점은 한 데이터 클래스(예: 스팸)의 한 가지 예를 나타냅니다.
- 각 **주황색** 점은 다른 데이터 클래스(예: 스팸이 아님)의 한 가지 예를 나타냅니다.
- **배경색**은 해당 색상의 예가 나타날 것으로 **모델에서 예측한 위치**를 나타냅니다.
- **학습 데이터는 윤곽선이 흰색**입니다.
- **평가 데이터는 윤곽선이 검은색**입니다.



학습 데이터



테스트 데이터

결정 경계

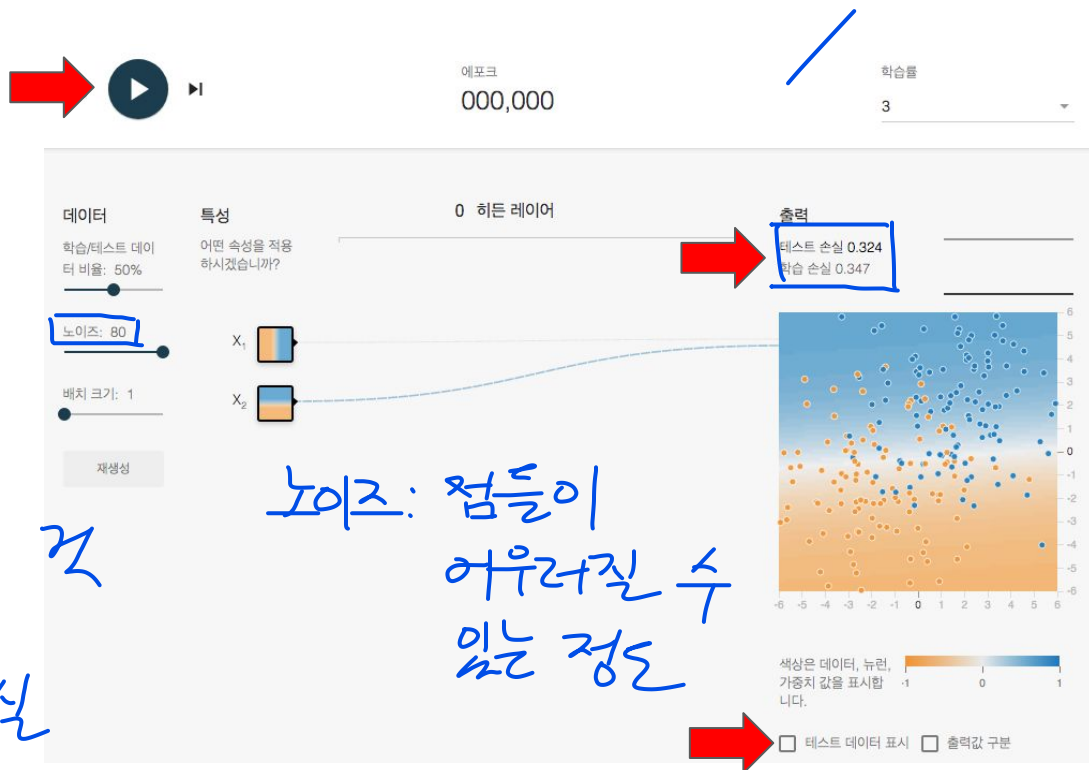
학습 및 평가 세트: 플레이그라운드 실습

시각화에는 기본적으로
학습자료만 표시됩니다.
평가자료도 함께 확인하려면
시각화 바로 아래의 **평가 데이터**
표시 체크박스를 클릭하세요.

텍스트 손실과 학습 손실의
차가 적을수록 잘 학습된 것

일반적으로 테스트 손실 > 학습 손실

학습 손실: 비용 함수의 값



Link to Playground: [NN playground - Linear Decision Boundary](#)

학습 및 평가 세트: 플레이그라운드 실습 (#1)

- 실행/일시중지 버튼 ➡을 클릭합니다.
- 테스트 손실 및 학습 손실 값의 변화를 주시합니다.
- 테스트 손실 및 학습 손실 값이 더 이상 변화하지 않거나 가끔씩만 변화한다면 실행/일시중지 ➡ 버튼을 다시 눌러 플레이그라운드를 일시중지합니다.

Question: 테스트 손실과 학습 손실 사이의 차이 값이 유의미한가요? ○

ANS: 학습률을 초기 설정과 같이 3으로 설정하면 테스트 손실 > 학습 손실보다 훨씬 큽니다.

학습 및 평가 세트: 플레이그라운드 실습 (#2)

- 재설정 버튼을 누릅니다.
- 학습률을 수정합니다. & 실행/일시중지 버튼을 누릅니다.
- 플레이그라운드를 150회 이상 반복 실행합니다.

Question : 새 학습률을 적용한 결과로 테스트 손실과 학습 손실 간의 델타 값이 낮아졌나요, 높아졌나요? **학습률**과 **배치 크기**를 모두 수정하면 어떻게 되나요? 차이

ANS : 학습률을 예를 들어 0.001로 낮추면 테스트 손실이 학습 손실과 매우 유사한 값까지 하락합니다. 대부분의 실행에서는 배치 크기를 늘려도 학습 손실 또는 테스트 손실에 유의미한 영향이 없습니다. 그러나 실행 중 소수에서는 배치 크기를 20 이상으로 늘릴 때 테스트 손실이 학습 손실보다 약간 낮은 값으로 하락합니다.

플레이그라운드의 데이터 세트는 무작위로 생성됩니다. 그 결과 이 답이 사용자의 답과 정확히 일치하지 않을 수도 있습니다.

학습 및 평가 세트: 플레이그라운드 실습(#3)

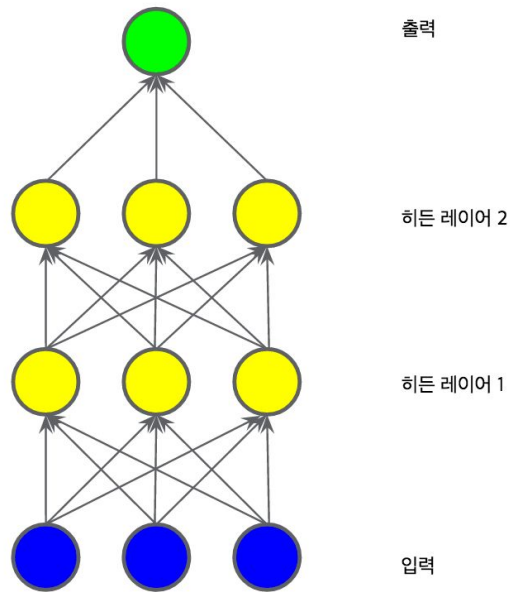
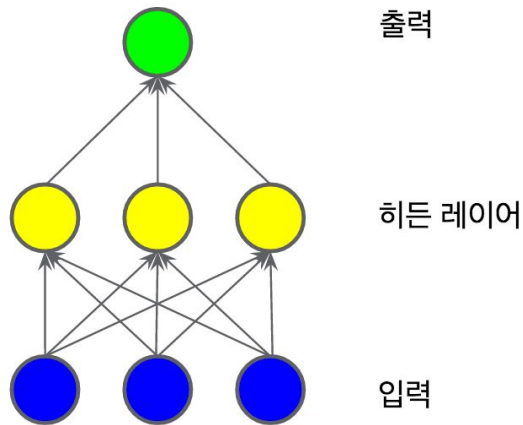
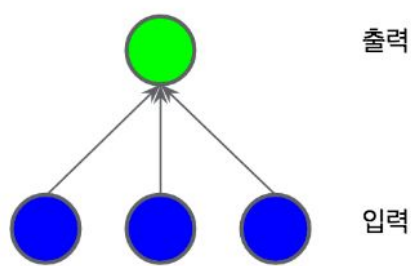
학습/테스트 데이터 비율이라는 슬라이더를 통해 평가 데이터와 학습 데이터의 비율을 조절할 수 있습니다. 예를 들어 90%로 설정하면 학습 세트에 평가 세트보다 훨씬 많은 예가 포함됩니다.

- '학습/평가 데이터 비율'을 50%에서 10%로 줄입니다.
- 학습률 및 배치 크기에 변화를 주면서 관찰 결과를 기록합니다.

Question : 학습/평가 데이터 비율을 바꾸면 (#2)에서 확인된 최적의 학습 설정이 변화하나요? 그렇다면, 이유가 무엇일까요?

ANS : 학습/평가 데이터 비율을 50%에서 10%로 줄이면 학습 세트의 데이터 포인트 수가 급격히 감소합니다. 데이터가 이렇게 적은데도 배치 크기와 학습률이 높으면 학습 모델이 최저점을 넘나들며 불규칙적으로 크게 변화하는 현상이 반복됩니다.

비선형 신경망 모델



출력을 입력의 함수로 표현하고 단순화하면 입력의 또 다른 가중 합을 얻게 됩니다. 이 합계는 비선형 결정 경계 문제를 효과적으로 모델링하지 않습니다.

활성화 함수 없이 여러층을 만들어 봤자
비선형 결정 경계를 만들 수 없다

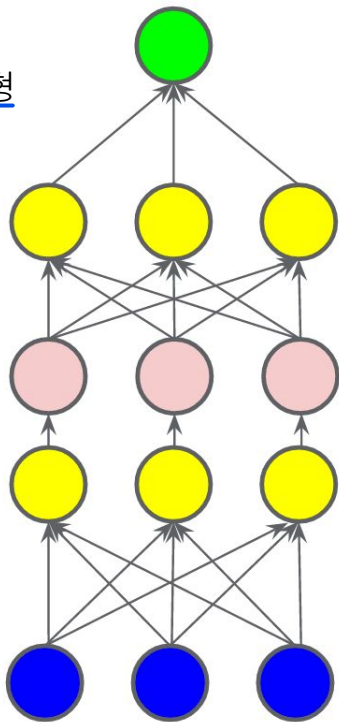
비선형 결정 경계를 만드려면

비선형 신경망 모델

비선형 문제를 모델링하기 위해
비선형성을 직접 도입할 수 있습니다.
우리는 각 히든 레이어의 노드가 비선형
함수를 통과하도록 할 수 있습니다.

보통의 경우 비선형 변환을
그리지 않습니다.

오른쪽 그림의 그래프로 나타낸
모델에서 히든 레이어 1의 각 노드
값이 비선형 함수로 변환된 후에
다음 레이어의 가중 합으로
전달되었습니다. 이 비선형 함수를
활성화 함수라고 합니다.



이제 활성화 함수를
추가하였으므로 레이어를 (히든)
추가하면 효과가 더 큽니다.

출력 비선형성을 누적하면 입력과
예측 출력 간의 매우 복잡한
관계를 모델링 할 수 있습니다.

히든 레이어 2

비선형 변환 레이어(활성화 함수라고도 함)

히든 레이어 1

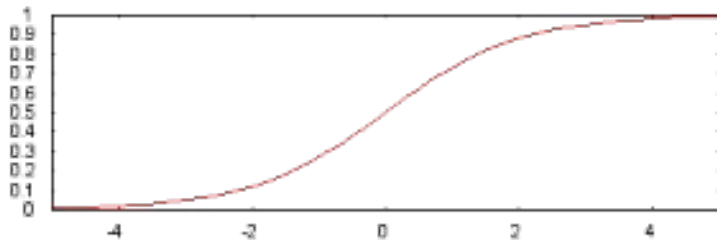
입력

비선형 신경망 모델 - 일반적인 활성화 함수

시그모이드 활성화 함수는 가중합을 0과 1 사이의 값으로 변환합니다.

$$F(x) = \frac{1}{1 + e^{-x}}$$

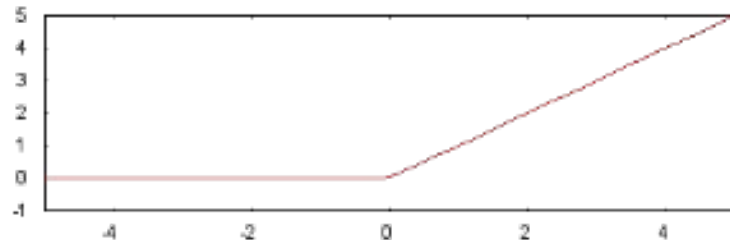
시그모이드



ReLU (Rectified Linear Unit) 활성화 함수는 시그모이드와 같은 매끄러운 함수보다 조금 더 효과적이지만, 훨씬 쉽게 계산할 수 있습니다.

보다 작으면 항상 0 이고
보다 크면 값 그대로,
 $F(x) = \max(0, x)$

ReLU



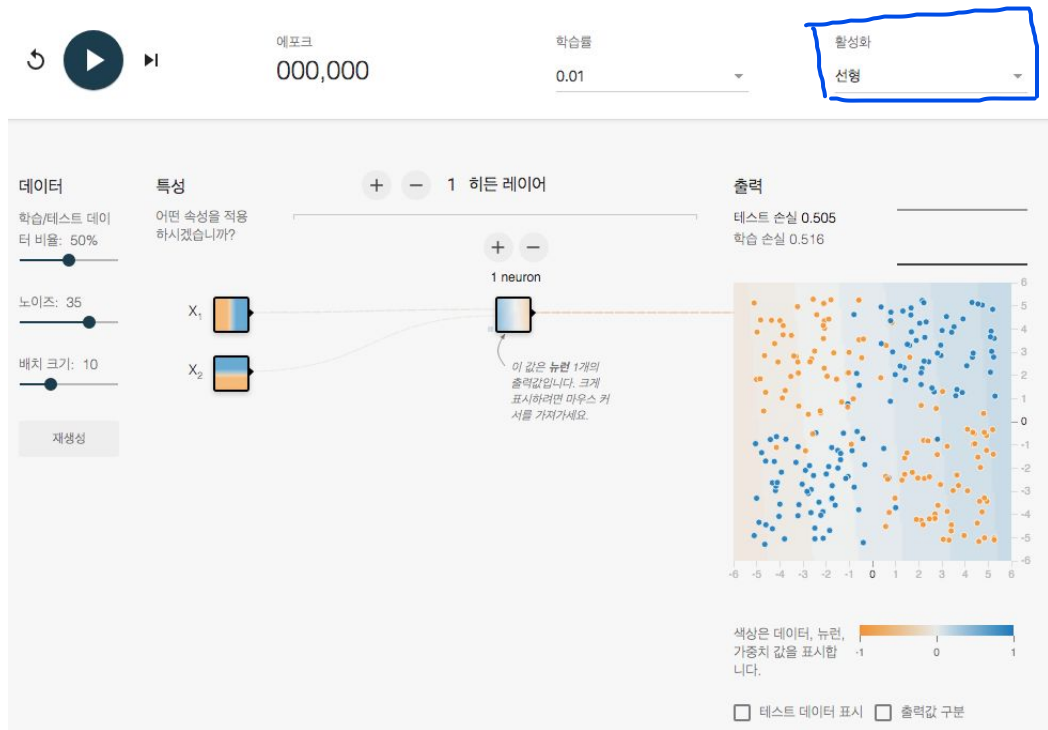
장점: vanishing gradient 약화

비선형 신경망 모델

이 실습에서는 **비선형 신경망**을 학습시켜 보게 됩니다.

과제 #1: 주어진 모델은 두 개의 입력 특성을 하나의 뉴런으로 결합합니다. 이 모델이 비선형성을 학습할 수 있을까요? 모델을 실행하여 추측을 확인해 보세요.

ANS: 활성화가 선형으로 설정되어 있으므로 이 모델은 어떠한 비선형성도 학습할 수 없습니다.
손실은 매우 높습니다.



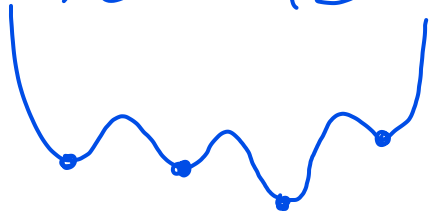
NN Playground Link : [NN Playground - Nonlinear Decision Boundary](#)

과제 #2: 히든 레이어의 뉴런 수를 1개에서 2개로 늘려보고 선형 활성화에서 **ReLU**와 같은 비선형 활성화로 변경해 보세요. 비선형성을 학습하는 모델을 만들 수 있나요?

ANS: 비선형 활성화 함수는 비선형 모델을 학습할 수 있습니다. 하지만 뉴런이 2개인 히든 레이어는 모델을 학습하는 데 시간이 오래 걸립니다. 이 실습은 비결정적이므로 일부 시도에서는 효과적인 모델을 학습하지 못합니다. 하지만 다른 시도에서는 상당히 효과적으로 학습할 수 있습니다. 여러번 시도해 보십시오.

local minimum 때문에 시작 위치나 속력 속도가 달라서

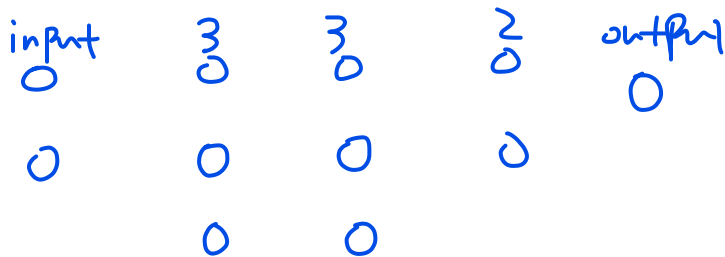
각 실행 2개의 결과를 매번 다르게 보여준다



과제 #3: 히든 레이어 및 레이어당 뉴런을 추가하거나 삭제하여 실험을 계속해 봅니다. 또한 자유롭게 학습률 및 기타 학습 설정을 변경해 보세요. 테스트 손실을 0.177 이하로 얻는 데 사용할 수 있는 가장 적은 노드 및 레이어 수는 얼마인가요?

ANS: Playground의 비결정적인 특징이 이 실습을 통해 빛을 발합니다. 몇 번의 실행으로 히든 레이어 3개의 테스트 손실이 매우 낮아지며 다음과 같은 결과를 도출했습니다.

- 1 레이어에는 뉴런이 3개 있습니다.
- 2 레이어에는 뉴런이 3개 있습니다.
- 3 레이어에는 뉴런이 2개 있습니다.

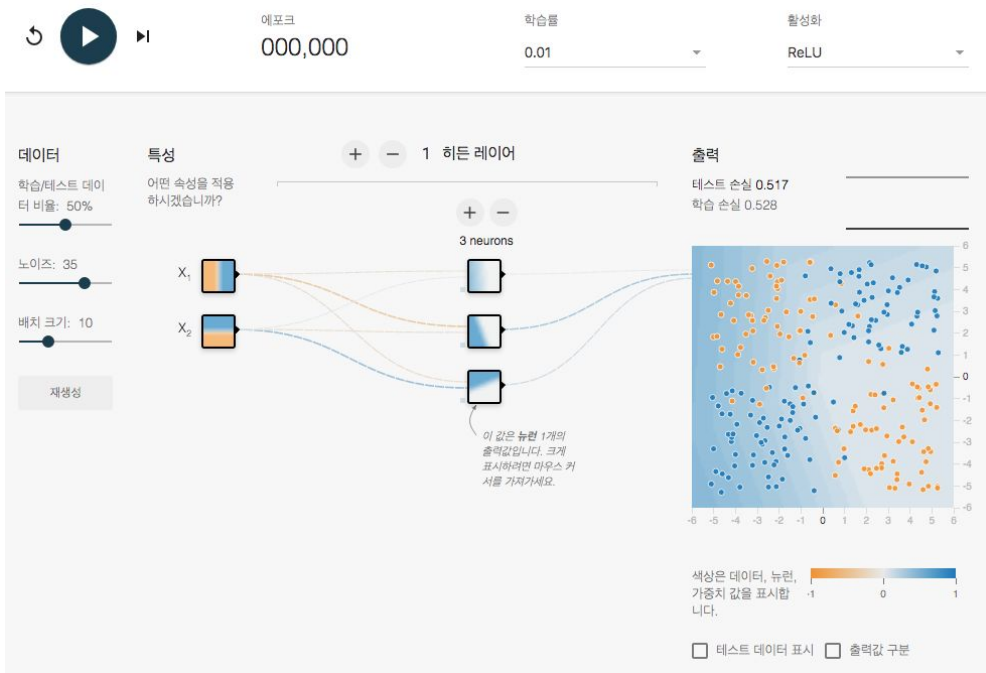


하지만 또 다른 실행에서는 같은 설정으로 매우 높은 손실이 발생했습니다.

비선형 신경망 모델 - 초기화와 반복성

이 실습에서는 학습용 신경망의 반복성과 초기화의 중요성을 살펴봅니다.

과제 1: 주어진 모델을 4~5회 실행합니다.
매번 시도하기 전에 네트워크 초기화 버튼을 눌러 임의로 새롭게 초기화합니다.
(네트워크 초기화 버튼은 재생 버튼 바로 왼쪽에 있는 원형의 초기화 화살표입니다.)
수렴을 보장하기 위해 **매 시도마다 최소 500단계를** 실행하도록 합니다. 각 모델 출력이 어떤 형태로 수렴하나요? 이 결과가 **비볼록 최적화에서 초기화의 역할**에 어떤 의미를 가지나요?



NN Playground Link : [NN Playground - initialization](#)

비선형 신경망 모델 - 초기화와 반복성

ANS: 각 시도마다 학습된 **모델의 형태**가 달랐습니다. 테스트 손실 수렴 결과는 최저와 최고가 거의 2배까지 차이가 날 정도로 다양했습니다.

과제 2: 레이어 한 개와 노드를 몇 개 더 추가하여 모델을 약간 더 복잡하게 만들어 보세요. 과제 1의 시도를 반복합니다. 결과에 **안정성이 보강되나요?**

ANS: 레이어와 추가 노드를 추가하여 더 반복적인 결과를 얻었습니다. 매 시도마다 결과 모델은 거의 같은 형태였습니다. 또한 테스트 손실 수렴 결과는 매 시도마다 변화가 적었습니다.

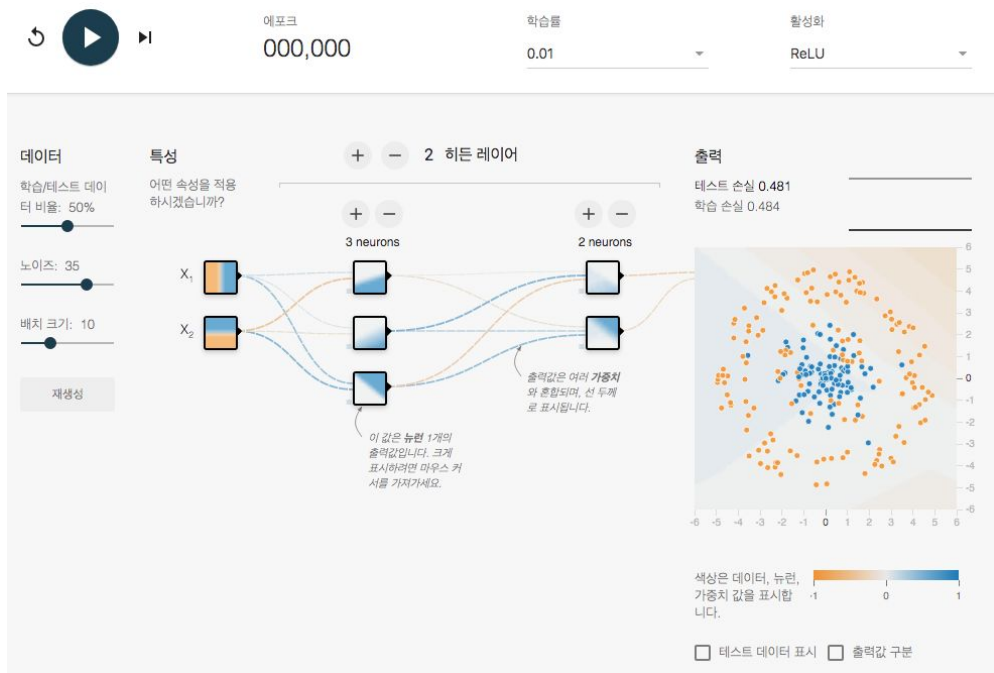
적절한 레이어와 노드 수를 적용하면

초기화해는 반복성 유지됨

비선형 신경망 모델 - 원형 결정경계

이 데이터 세트는 **잡음이 있는 원형**입니다. 분명 선형 모델은 여기에서 실패하며 결정 경계 확정이 어려울 수 있습니다.

과제 #1: X_1 과 X_2 만 사용하여 가능한 한 최고의 모델로 학습시켜 봅시다. 자유롭게 레이어와 뉴런을 추가 또는 삭제하고 학습률, 배치 크기와 같은 학습 설정을 변경해 보세요. 얻을 수 있는 최고의 테스트 손실은 얼마인가요?

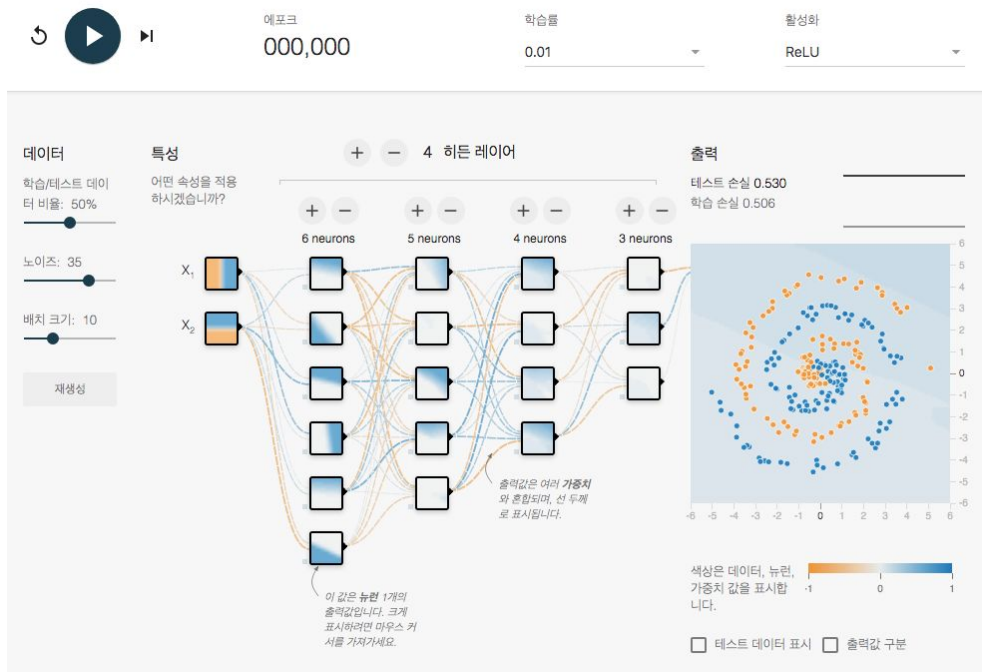


NN Playground Link : [NN Playground - circle](#)

비선형 신경망 모델 - 나선형 결정경계

이 데이터 세트는 **잡음이 있는 나선형**입니다. 분명 선형 모델은 여기에서 실패하겠고, 결정 경계 구성이 어려울 수 있습니다.

과제 #1: X_1 과 X_2 만 사용하여 가능한 한 최고의 모델로 학습시켜 봅시다. 자유롭게 레이어와 뉴런을 추가 또는 삭제하고 학습률, 배치 크기와 같은 학습 설정을 변경해 보세요. 얻을 수 있는 최고의 테스트 손실은 얼마인가요?



NN Playground Link : [NN Playground - spiral](#)

Homework - Link : [NN Playground - spiral](#)

당신의 최적 초파라메타를 찾아 찾아 적으시오. 훈련/테스트 비율은 **70%**로, 잡음은 **20%**로 하시오. 학습/테스트 손실 **< 0.1** 여야 합니다.

data type	활성화	배치크기	# 히든레이어	# 뉴런	에포크	학습손실	테스트 손실
(예)	ReLU	30	2	6, 2	156	0.05	0.09
two blobs							
XOR							
Circle							
Spiral							