

TensorFlow Spiral 데이터 완벽 정복: 좋은 성능 vs. 나쁜 성능 비교 분석

TensorFlow Playground의 **Spiral(나선형) 데이터셋**은 신경망의 핵심 원리를 이해하는 데 가장 좋은 예제 중 하나입니다. 단순한 직선으로는 절대 두 데이터를 나눌 수 없기 때문이죠. 이 데이터셋을 통해 왜 특정 설정이 실패하고, 특정 설정이 성공하는지 명확하게 알아보겠습니다.

1. 실패 사례: 왜 모델은 나선형을 학습하지 못할까요?

가장 단순한 모델로 나선형 데이터를 학습시켜보겠습니다. 결과는 아마 예상대로 처참할 겁니다.

🔧 나쁜 성능을 내는 설정 예시

- **Problem type:** Classification
- **Features:** x1,x2 (가장 기본적인 좌표 2개만 사용)
- **Hidden Layers:** 1개 층, 뉴런 2개 (최소한의 구성)
- **Activation:** Linear (치명적인 선택)
- **Learning Rate:** 0.3 (너무 높음)
- **Regularization:** L2, Regularization rate = 0.1 (과하게 높음)

결과 및 분석

결정 경계가 **단순한 직선 하나로** 나타나며, 복잡하게 얽힌 나선형 패턴을 전혀 구분하지 못합니다. 아마 Test loss는 0.5에 가깝게 나올 것이며, 이는 동전을 던져서 맞추는 것과 같은 수준입니다.

실패의 핵심 원인: 예시로 이해하기

1. 활성화 함수 (Activation): Linear

- **문제점:** Linear 함수는 이름 그대로 '직선'적인 관계만 표현할 수 있습니다.
- **쉬운 비유:** 마치 '직선 자' 하나만 가지고 구불구불한 나선형을 따라 그리려는 것과 같습니다. 아무리 노력해도 직선 자로는 곡선을 그릴 수 없습니다. 나선형 데이터는 본질적으로 '비선형(Non-linear)' 문제이므로, 직선만 그을 수 있는 Linear 함수는 최악의 선택입니다.

2. 모델의 복잡도 (Hidden Layers & Neurons)

- **문제점:** 은닉층 1개와 뉴런 2개는 모델이 학습할 수 있는 패턴의 복잡도를 극도로 제한합니다.
- **쉬운 비유:** '**점 2개만 찍어서 복잡한 그림을 완성하라**'는 미션과 같습니다. 나선형의 세밀한 경계를 표현하기에는 모델의 표현력이 턱없이 부족합니다.

3. 학습률 및 규제 (Learning Rate & Regularization)

- **문제점:** 너무 높은 학습률(0.3)은 최적의 지점을 찾지 못하고 계속 헤매게 만듭니다. 반면, 너무 강한 규제(0.1)는 모델이 조금이라도 복잡해지려는 시도를 '벌'을 주어 방해합니다.

- 쉬운 비유: 눈을 가리고 산을 내려오는데, 보폭(Learning Rate)은 너무 크고, 옆 사람(Regularization)은 계속 똑바로만 가라고 소리치는 상황과 비슷합니다. 구불구불한 등산로를 제대로 찾아 내려올 수 없습니다.

2. 성공 사례: 나선형의 비밀을 푸는 열쇠

이제 모델에게 적절한 도구들을 쥐여주고 나선형 문제를 풀게 해보겠습니다.

좋은 성능을 내는 설정 예시

- **Problem type:** Classification
- **Features:** $x_1, x_2, x_{12}, x_{22}, x_1x_2, \sin(x_1), \sin(x_2)$ (다양한 파생 특징 추가)
- **Hidden Layers:** 3개 층, 각 층에 8, 6, 4개의 뉴런 (깊고 충분한 구성)
- **Activation:** Tanh 또는 ReLU (최적의 선택)
- **Learning Rate:** 0.03 (안정적)
- **Regularization:** 없음(None) 또는 L2, Regularization rate = 0.001 (매우 약하게)

결과 및 분석

결정 경계가 데이터의 나선형 모양을 아름답게 따라가며 두 데이터를 거의 완벽하게 구분합니다. Test loss는 0.01 이하로 떨어지며 매우 안정적으로 수렴합니다.

성공의 비결

4. 활성화 함수 (Activation): Tanh 또는 ReLU

- **비결:** Tanh, ReLU와 같은 비선형 함수는 모델이 '곡선'을 학습할 수 있게 해줍니다. 이것이 성공의 가장 결정적인 열쇠입니다.
- **쉬운 비유:** 이제 모델은 '**'유연하고 부드러운 곡선 자'**를 갖게 되었습니다. 이 자를 자유자재로 구부려서 나선형의 복잡한 경계를 그대로 따라 그릴 수 있게 된 것입니다.

5. 모델의 복잡도 (Hidden Layers & Neurons)

- **비결:** 여러 개의 층과 충분한 뉴런은 모델이 **단계별로 복잡한 패턴을 학습**하게 합니다.
- **쉬운 비유:** 첫 번째 층이 '간단한 곡선 조각들'을 찾아내면, 두 번째 층은 그 조각들을 '조합하여 더 긴 곡선'을 만듭니다. 그리고 마지막 층이 이 곡선들을 '연결하여 최종적인 나선형'을 완성하는 방식입니다. 층이 깊어질수록 더 추상적이고 복잡한 형태를 만들 수 있습니다.

6. 다양한 특징 (Features)

- **비결:** x_{12}, x_{22} 같은 추가 특징은 모델에게 '더 많은 힌트'를 줍니다. 예를 들어 $x_{12} + x_{22}$ 은 원점으로부터의 거리를 의미하는데, 나선형 패턴에서 이는 매우 중요한 정보입니다.
- **쉬운 비유:** 탐정이 사건을 해결할 때, 기본적인 단서(x_1, x_2) 외에 '지문, 발자국, CCTV 영상'(x_{12}, x_{1x2} 등)과 같은 **추가적인 단서를 확보**하는 것과 같습니다. 단서가 많을수록 정답을 훨씬 쉽고 빠르게 찾을 수 있습니다.

III Spiral 데이터 성능 비교 요약

구분	나쁜 설정 (성능 낮음)	좋은 설정 (성능 높음)	이유
활성화 함수	Linear	Tanh / ReLU	나선형은 '비선형' 경계가 필수. 직선 자 vs 곡선 자의 차이.
은닉층/뉴런	1층, 뉴런 2개	3층, 각 6~8개 뉴런	단순한 그림 vs 복잡한 그림. 모델의 표현력 차이.
사용 특징	x1,x2	x1,x2,x12,x22...	기본 단서 vs 추가 단서. 문제 해결의 힌트 제공.
학습률	0.3 (너무 큼)	0.03 (적절)	성급한 학습(발산) vs 안정적인 학습(수렴).
규제	L2, $\lambda=0.1$ (강함)	없음 또는 $\lambda=0.001$ (약함)	지나친 단순화 강요 vs 자유로운 학습 허용.

★ 최종 정리

Spiral 데이터셋은 "복잡한 문제에는 그에 맞는 복잡한 모델과 도구가 필요하다"는 딥러닝의 핵심을 명확하게 보여줍니다.

- **실패 원인:** Linear 활성화 함수와 너무 단순한 네트워크 구조.
- **성공 비결:** 비선형 활성화 함수, 충분히 깊고 넓은 네트워크, 그리고 문제 해결에 도움이 되는 파생 특징의 조합입니다.