

나만의 인공지능 · 1강

두뇌는 이렇게 만듭니다

인공지능을 만든다는 것이 무슨 일인지, **가장 기초적인 구조**만 봅니다.

어려운 말은 하나도 쓰지 않습니다.

3단계

만들고 · 가르치고 · 열어본다

2개

두뇌 안에 든 숫자

0원

오늘 드는 비용

넣으면 나오는 상자입니다

학교에서 배운 함수를 기억하시죠. 넣으면 나옵니다.

1 → [함수] → 5

인공지능도 똑같이 생겼습니다. 넣는 것이 숫자가 아닐 뿐입니다.

「사랑해」 → [번역하는 상자] → 「I love you」

고양이 사진 → [알아보는 상자] → 「고양이입니다」

이 상자 하나를 만드는 것이 오늘 하는 일입니다.

상자를 사람이 안 만듭니다

지금까지

사람이 상자를 만들어 줬습니다

「화씨는 섭씨 곱하기 1.8 더하기 32」라고 **사람이 알려 주고**, 컴퓨터는 계산만 했습니다.

인공지능

기계가 상자를 스스로 만듭니다

예시만 보여 줍니다. **0도는 32도, 8도는 46도...** 규칙은 알려 주지 않습니다. 기계가 찾아냅니다.

그래서 「학습」이라고 부릅니다. **규칙을 주는 대신, 예시를 주고 규칙을 찾게 하는 것**입니다.

세 단계가 전부입니다

1

만든다

빈 두뇌를 하나 준비합니다. 아직 아무것도 모릅니다.

2

가르친다

예시를 보여 줍니다. 두뇌가 조금씩 자기를 고칩니다.

3

열어본다

무엇을 배웠는지 안을 들여다봅니다.

챗GPT를 만드는 것도 이 세 단계입니다. **크기만 다릅니다.**

두뇌를 만듭니다

```
직선두뇌 = keras.Sequential([
    keras.Input(shape=(1,)),
    keras.layers.Dense(1),
])
```

1

Sequential — 층을 차례로 쌓아라

대괄호 안에 위에서 아래로 층을 적습니다. 적은 순서대로 신호가 지나갑니다.

2

Input — 들어오는 것은 숫자 하나

섭씨 온도 하나가 들어온다고 알려 줍니다.

3

Dense(1) — 계산 칸 한 개

이 칸이 하는 일은 곱하고 더하기가 전부입니다.

이 세 줄이 두뇌 하나입니다. **층을 쌓는 것**이 두뇌를 만드는 일입니다.

숫자 두 개가 들어 있습니다

입력 ——(곱하기 **w**)——(더하기 **b**)——▶ 출력

방금 만든 두뇌 안에는 숫자가 **딱 두 개** 들어 있습니다. 하나는 곱하는 숫자, 하나는 더하는 숫자입니다.

지금

아무 값

만들자마자 아무 숫자나 들어 있습니다. 그래서 **아무 말이나 합니다.**

학습이란

이 두 숫자를 고쳐 나가는 일

그게 전부입니다. 더 복잡한 두뇌는 **이 숫자가 수천억 개**일 뿐입니다.

가르칩니다

직선두뇌.fit(셉씨, 화씨, epochs=500)

fit 이 「가르쳐라」입니다. 셉씨를 보여 주고 화씨가 정답이라고 알려 줍니다. **500번** 반복해서 봅니다.

01

답해 본다

지금 아는 대로 화씨를 계산해 봅니다.

02

얼마나 틀렸나

정답과 비교합니다.

03

어디로 고칠까

더 커야 하는지 작아야 하는지 봅니다.

04

조금 고친다

두 숫자를 아주 조금 움직입니다.

이 네 걸음을 **500번** 둡니다. 그게 학습입니다. 사람이 연습하는 것과 같습니다.

열어봅니다

```
w, b = 직선두뇌.layers[0].get_weights()
```

	곱하는 숫자	더하는 숫자
가르치기 전	아무 값	0
가르친 뒤	1.7979	31.9525
실제 정답	1.8	32

공식을 알려 준 적이 없습니다.

숫자 일곱 줄만 보여 줬는데 스스로 1.8과 32를 찾아냈습니다.

이 순간이 오늘 실습의 핵심입니다. 기계가 규칙을 찾아내는 것을 눈으로 보는 자리입니다.

층 하나로 안 되는 일이 있습니다

계산 칸 하나는 **직선**밖에 못 그립니다. 굽은 것은 못 그립니다.

층 하나

직선만

아무리 오래 가르쳐도 **굽힐** 수가 없습니다.

층 여러 겹

곡선도

층을 **세 겹**으로 쌓으면 굽은 모양도 그려냅니다.

```
깊은두뇌 = keras.Sequential([
    keras.Input(shape=(1,)),
    keras.layers.Dense(64, activation='relu'), < 한 겹 더
    keras.layers.Dense(64, activation='relu'), < 또 한 겹
    keras.layers.Dense(1),
])
```

층을 **깊게 쌓은 것**, 그것이 **딥러닝**입니다. 깊게(deep) 배운다(learning)는 말 그대로입니다.

activation='relu' 는 직선을 꺾을 수 있게 해 주는 장치입니다. 지금은 「층을 쌓을 때 같이 적는 것」 정도로만 아셔도 됩니다.

로컬AI의 기초 구조

① 두뇌를 만든다	층을 쌓는다
Sequential([...])	
② 가르친다	예시를 보여 준다
fit(문제, 정답)	
③ 쓴다 · 열어본다	답을 시키거나 안을 본다
predict / get_weights	

=

이것이 전부입니다

허깅페이스에 올라와 있는 수많은 모델들도 **이 세 단계로 만들어진 것**입니다. 누군가 층을 쌓고, 오래 가르치고, 그 결과를 올려놓은 것입니다.

→

그래서 우리가 할 수 있는 일

남이 오래 가르쳐 둔 두뇌를 **내려받아 쓰고**, 내 자료로 **조금 더 가르치고**, 여러 개를 **이어 붙여** 나만의 것을 만듭니다.

내 컴퓨터로 가져옵니다

오늘은 구글이 빌려주는 컴퓨터에서 만들어 봤습니다. 껏다 켜면 사라집니다.

다음

내 컴퓨터에 올리기

남이 몇 달 걸쳐 가르쳐 둔 큰 두뇌를 그대로 내려받습니다.

그다음

내 지식 넣기

내 자료를 두뇌에 붙여 나를 아는 AI로 만듭니다.

그다음

귀와 입 달기

듣고 말하게 합니다.

끝에

나만의 자비스

사용료 0원. 빌리는 것이 아니라 소유합니다.

강의 보러 가기 ↗

AI CITY BUILDERS · 나만의 인공지능 1강 교재 · 실습 코랩은 강의 페이지에서 열 수 있습니다