

● DAY 11 · → 20 · 2 부 여기까지 온 길 · 고급 총정리

2 부 총정리 — 여기까지 온 길

DAY 11-20 · 용어를 정의하고 그대로 쓰는 판 · 원리와 배경까지



시험 없음 — 읽고 넘어가는 복습 자료

뉴런 하나를 흉내 낸 첫 기계

1

1958 년 프랭크 로젠블랫이 퍼셉트론을 만들었어 .

2

들어온 신호마다 무게를 곱해 더하고 , 합이 문턱을 넘으면 1, 못 넘으면 0 을 내는 인공 뉴런 하나야 .

3

진짜 새로웠던 건 틀리면 스스로 무게를 고쳤다는 점이야 — 규칙을 안 써 줘도 예시만 보고 배운 첫 기계였어 .

왜 이게 출발점일까

1

지금의 거대한 AI 도 뜯어보면 이 뉴런을 아주 많이 쌓고 이어 붙인 거야 .

2

' 신호에 무게를 곱해 더한다 ' 와 ' 틀리면 무게를 고친다 ' , 이 두 가지는 지금도 그대로야 .

3

그보다 15 년 앞선 1943 년에 매컬러와 피츠가 뉴런을 논리 연산으로 적어 낸 글이 있었지만 , 스스로 배우는 기계를 실제로 만든 건 퍼셉트론이 처음이었어 .

용어로 다시 보면

1

퍼셉트론의 계산을 용어로 쓰면 , 입력마다 가중치 (weight) 를 곱해 더한 가중합에 편향 (bias) 을 더하고 , 그 값이 0 보다 크면 1 을 내는 계단 함수야 .

2

학습 규칙은 단순해 — 틀린 예시가 나오면 정답 쪽으로 가중치를 조금 옮긴다 .

3

이런 한 층짜리 모델은 데이터를 직선 하나로 가를 수 있을 때만 제대로 배우는데 , 이 조건을 선형 분리 가능 (linearly separable) 이라고 불러 — 바로 다음 날 이야기가 여기서 나와 .

기대가 실력보다 앞서 가면 겨울이 온다

1

1969 년 민스키와 페퍼트가 한 층짜리 뉴런으로는 못 푸는 문제가 있다는 걸 수학으로 증명하자 , 연구비도 사람도 빠져나갔어 — 첫 번째 겨울이야 .

2

1980 년대엔 규칙을 사람이 다 적어 주는 전문가 시스템이 붐이었는데 , 적어 둔 상황을 조금만 벗어나도 틀렸고 1987 년엔 그 전용 컴퓨터 시장이 무너졌어 — 두 번째 겨울이야 .

3

두 번 다 AI 가 못해서가 아니라 , 약속이 실력보다 앞서 갔기 때문이야 .

겨울을 녹인 한 편의 논문

1

첫 겨울의 원인은 한 층짜리의 한계였어 .

2

층을 여러 겹 쌓으면 되지만 , 여러 층일 때 어느 무게를 얼마나 고쳐야 할지 계산하는 방법이 널리 쓰이지 않았지 .

3

1986 년 네이처에 실린 역전파 논문이 층을 쌓아도 무게를 고칠 수 있다는 걸 널리 알리면서 겨울이 녹기 시작했어 .

XOR 과 역전파

1

한 층짜리가 못 푸는 대표 문제가 XOR 이야 — 두 입력 중 정확히 하나만 1 일 때 1 을 내는 규칙인데 , 네 점을 직선 하나로 가를 수 없어서 선형 분리가 안 돼 .

2

층을 하나만 더 끼워도 풀리지만 , 그러려면 중간 층의 가중치가 오차에 얼마나 책임이 있는지 계산해야 해 .

3

역전파 (backpropagation) 는 출력의 오차를 미분의 연쇄 법칙으로 뒤쪽 층부터 거꾸로 나눠 주는 방법이고 , 지금 신경망 학습은 거의 전부 이 방법 위에 서 있어 .

세 가지가 한자리에 모인 날

1 2012 년 사진 맞추기 대회 이미지넷에서 토론토대 팀의 신경망이 오류 15.3% 로 우승했어 .

2 2 위가 26.2% 였으니 10% 포인트가 넘는 차이였지 .

3 새로 발명한 건 거의 없었어 — 사진 수백만 장 , 게임용 그래픽카드 , 깊게 쌓은 신경망이 그해에 처음으로 한자리에 모인 거야 .

왜 하필 2012 년이었을까

1

이미지넷은 사진 수백만 장에 이름표를 붙여 둔 데이터베이스로 2009 년에 공개됐어 .

2

게임용 그래픽카드는 같은 계산을 한꺼번에 많이 하는 데 강해서 신경망 학습에 딱 맞았고 , 우승 팀은 그래픽카드 두 장으로 5~6 일을 학습시켰어 .

3

재료가 다 갖춰지자 오래된 아이디어가 드디어 실력을 보였고 , 이날 이후 사진 연구는 통째로 신경망으로 넘어갔어 .

무엇이 깊었나

1

우승한 모델은 격자로 된 입력에 강한 합성곱 신경망 (CNN) 이었어 — 작은 필터를 사진 전체에 미끄러뜨리며 모서리나 무늬 같은 특징을 찾아내는 구조야 .

2

층이 깊을수록 앞쪽은 선과 모서리를 , 뒤쪽은 눈이나 바퀴 같은 덩어리를 알아보는 식으로 특징이 쌓여 .

3

'top-5 오류 ' 는 모델이 내놓은 후보 다섯 개 안에 정답이 없을 비율이라 , 15.3% 는 사진 100 장 중 약 85 장에서 다섯 후보 안에 정답을 넣었다는 뜻이야 .

단어들이 서로를 쳐다보기 시작했다

1

2017 년 구글 연구팀의 논문 'Attention Is All You Need' 가 지금 AI 의 뼈대인 트랜스포머를 정했어 .

2

그 전엔 문장을 앞에서부터 한 단어씩 읽었는데 , 트랜스포머는 문장 속 모든 단어가 서로를 동시에 쳐다보게 했어 .

3

동시에 본다는 건 한꺼번에 계산할 수 있다는 뜻이라 , 그래픽카드와 딱 맞았지 .

쳐다본다는 게 뭘까

1

'동물이 길을 건너지 않았다, 그것은 너무 피곤했다'에서 '그것'은 동물이고, '너무 넓었다'로 바꾸면 길이야.

2

트랜스포머는 단어마다 문장 속 다른 단어를 얼마나 봐야 하는지 점수를 매겨서 이런 연결을 한 번에 찾아.

3

차례로 읽을 땐 멀리 떨어진 단어가 흐려졌는데, 서로 직접 쳐다보니 거리가 상관없어졌어.

셀프 어텐션의 속

1

이 구조를 셀프 어텐션 (self-attention) 이라고 불러 .

2

각 단어는 '내가 찾는 것 (쿼리)', '내가 가진 것 (키)', '넘겨줄 내용 (값)' 세 가지 벡터 — 숫자를 줄지어 놓은 목록 — 를 만들고 , 쿼리와 키가 잘 맞을수록 그 단어의 값을 많이 가져와 .

3

그 전의 순환 신경망 (RNN) 은 앞 단어 계산이 끝나야 다음으로 넘어가서 병렬로 돌리기 어려웠는데 , 셀프 어텐션은 모든 쌍을 행렬 곱셈으로 한꺼번에 계산해 — 그래픽카드가 가장 잘하는 일이야 .

키웠더니 예시만 보고 따라 했다

1 트랜스포머 뼈대는 그대로 두고 크게만 키웠더니 , 2020 년 숫자 1,750 억 개짜리 GPT-3 가 나왔어 .

2 놀라운 건 다시 훈련시키지 않아도 질문 안에 예시 두세 개만 넣으면 바로 따라 했다는 거야 .

3 이때부터 ' 뭐라고 시키느냐 ' , 즉 프롬프트가 중요해졌어 .

얼마나 커진 걸까

1

GPT-1 은 약 1 억 1,700 만 개 , GPT-2 는 약 15 억 개 , GPT-3 는 1,750 억 개였어 .

2

2018 년부터 2020 년 사이에 천 배 넘게 커진 셈이야 .

3

숫자를 키우면 성능이 얼마나 오를지 미리 계산해 보는 연구가 먼저 있었고 , GPT-3 는 그 계산을 실제로 끝까지 밀어붙인 결과였어 .

문맥 내 학습과 스케일링

1

GPT-3 가 보여 준 재주를 문맥 내 학습 (in-context learning) 이라고 불러 — 가중치는 하나도 바꾸지 않고 , 입력에 들어온 예시만 보고 그 자리에서 규칙을 따라 하는 거야 .

2

예시를 몇 개 주느냐에 따라 제로샷 (0 개) · 원샷 (1 개) · 퓨샷 (몇 개) 으로 나눠 부르고 , 논문 제목의 'Few-Shot' 이 바로 이거야 .

3

모델 크기 · 데이터 · 계산량을 늘릴 때 성능이 예측 가능한 곡선을 따라 좋아진다는 관찰을 스케일링 법칙이라고 부르는데 , 이게 ' 더 키우자 ' 는 투자의 근거가 됐어 .

세상을 바꾼 건 대화창이었다

1

2022 년 11 월 30 일 , GPT 계열 모델을 말을 주고받는 대화창에 얹어 누구나 쓰게 연 게 ChatGPT 야 .

2

사람들이 어떤 답을 더 좋아하는지 골라 준 걸로 다시 다듬어서 , 시키는 걸 제대로 알아듣게 만든 게 컸어 .

3

그래서 이날 바뀐 건 기술보다 쓰는 사람의 범위였어 .

작은 모델이 큰 모델을 이긴 실험

1

그 다듬기 방법을 정리한 2022 년 논문에서 , 사람 피드백으로 다듬은 13 억 개짜리 모델의 답을 사람들은 1,750 억 개짜리 GPT-3 의 답보다 더 좋아했어 .

2

크기만이 답이 아니라는 신호였지 .

3

여기에 예시를 붙일 줄 몰라도 그냥 말 걸면 되는 창이 더해지자 , 그제서야 모두가 AI 를 직접 만져 보게 됐어 .

RLHF 라는 다듬기

1

사람 피드백으로 다듬는 방법을 RLHF(사람 피드백을 통한 강화학습) 라고 불러 .

2

사람이 쓴 모범 답으로 먼저 한 번 다듬고 , 같은 질문에 대한 여러 답을 사람이 순위 매긴 데이터로 ' 어떤 답이 더 좋은지 ' 점수를 매기는 보상 모델을 만든 다음 , 그 점수가 높아지는 쪽으로 언어 모델을 다시 학습시켜 .

3

이렇게 모델이 사람의 뜻에 맞게 움직이도록 맞추는 일을 정렬 (alignment) 이라고 하는데 , 쓸모 있고 해롭지 않은 답을 내게 만드는 연구가 여기서 본격적으로 시작됐어 .

답하기 전에 풀이부터

1 어려운 문제에서 AI 가 확 잘해진 비결은 더 커진 게 아니라 , 답하기 전에 풀이 단계를 밟게 한 거야 .

2 처음엔 사람이 ' 단계별로 생각해 봐 ' 라고 시켜야 했는데 , 나중엔 스스로 그러도록 훈련받은 추론 모델이 나왔어 .

3 대신 생각이 길수록 답이 늦고 글자도 많이 써서 , 쉬운 질문엔 굳이 안 써 .

숫자로 보면

1

2022 년 한 논문은 질문 끝에 ' 단계별로 생각해 보자 ' 한 줄만 붙여도 수학 정답률이 크게 뛴다는 걸 보였어 .

2

2024 년엔 시키지 않아도 스스로 풀이를 쓰도록 훈련받은 모델이 나왔는데 , 같은 미국 고교 수학 대회 문제에서 바로 답하던 모델은 12% , 이 모델은 74% 를 맞혔어 .

3

뼈대를 바꾼 게 아니라 답하는 차례를 바꾼 결과야 .

사고의 사슬과 생각하는 데 드는 값

1

풀이를 먼저 쓰게 하는 방식을 사고의 사슬 (chain-of-thought) 이라고 불러 .

2

언어 모델은 한 조각씩 이어 쓰니까 , 중간 결과를 글로 적어 두면 다음 조각을 고를 때 그걸 다시 볼 수 있어 — 암산 대신 연습장을 쓰는 셈이지 .

3

추론 모델은 답하기 전에 쓰는 생각을 늘릴수록 어려운 문제의 정답률이 오르는 경향을 보이는데 , 이렇게 답하는 순간에 쓰는 계산을 추론 시점 계산 (test-time compute) 이라고 하고 , 모델을 키우는 것과는 다른 두 번째 손잡이가 됐어 .

모를 때 계산기를 잡는 것

1 AI 는 다음 낱말을 잇는 게 본업이라 , 큰 곱셈이나 오늘 날씨처럼 정확히 알아야 하는 것에 약해 .

2 그래서 계산기 · 검색 · 코드 실행 같은 도구를 붙여 줬고 , 2023 년부터는 언제 집을지 모델이 스스로 고르게 됐어 .

3 잘하는 AI 는 다 아는 AI 가 아니라 , 모른다는 걸 알아채고 도구를 잡는 AI 야 .

도구를 집는 네 걸음

1

질문을 받고 , 도구가 필요한지 판단하고 , 정해진 형식으로 도구를 부르고 , 받은 결과로 답을 마무리해 .

2

핵심은 두 번째 걸음이야 — 예전엔 사람이 ' 검색해 봐 ' 라고 시켜야 했는데 , 지금은 모델이 스스로 정해 .

3

2023 년 논문 톨포머는 계산기 · 검색 · 달력을 언제 부를지 모델이 스스로 배우게 했고 , 같은 해 모델이 도구 이름과 값을 정해진 형식으로 말하는 기능이 공식으로 열렸어 .

함수 호출과 에이전트

1

모델이 도구를 부르는 이 형식을 함수 호출 (function calling) 이라고 불러 .

2

모델은 '어떤 함수를 , 어떤 값으로 ' 를 미리 정한 구조의 글로 내놓을 뿐이고 , 실제 실행은 바깥 프로그램이 한 뒤 결과를 다시 모델에게 넣어 줘 — 모델이 직접 계산기를 누르는 게 아니야 .

3

이 걸음을 여러 번 이어 가며 스스로 계획하고 도구를 부르고 결과를 확인하는 시스템을 에이전트라고 부르는데 , 오늘날 ' 일을 맡기는 AI' 는 대부분 이 구조야 .

무엇이든 조각으로 쪼개 한 줄로

1

AI 가 사진을 보고 소리를 듣게 된 건 눈과 귀를 단 게 아니야 .

2

사진은 격자로 자른 네모 조각으로 , 소리는 아주 짧은 시간 토막으로 잘라 글자 조각과 같은 줄에 세운 거야 .

3

그러면 뼈대를 새로 만들 필요 없이 , 조각들이 서로 쳐다보게 하는 트랜스포머에 그대로 넣으면 돼 .

같이 배우면 뭐가 좋을까

1

처음엔 사진을 따로 글로 옮겨 적은 뒤 언어 모델에 넘기는 식이었어 .

2

처음부터 글 · 사진 · 소리를 한 뼈대로 함께 배운 모델은 그 중간 단계를 건너뛰어서 더 빠르고 정확해 .

3

2023 년엔 사진을 읽는 기능이 널리 열렸고 , 2024 년엔 세 가지를 함께 배운 모델이 나와 말 대답이 평균 0.32 초까지 빨라졌어 .

패치와 멀티모달

1

사진을 격자로 자른 네모 조각을 패치 (patch) 라고 불러 .

2

조각 하나하나를 숫자 목록으로 바꾸고 어디에 있던 조각인지 위치 정보를 붙이면 , 트랜스포머 입장에선 낱말 토큰과 다를 게 없어 .

3

이렇게 글 · 사진 · 소리처럼 서로 다른 종류의 입력을 함께 다루는 걸 멀티모달 (multimodal) 이라고 하고 , 처음부터 함께 배웠는지 따로 만든 부품을 이어 붙였는지에 따라 속도와 정확도가 갈려 .

쓰는 것에서 고르는 것으로

1

2022 년 말엔 쓸 만한 대화형 AI 가 손에 꼽을 정도였는데 , 지금은 미국 · 유럽 · 중국 · 한국까지 수십 곳이 저마다 모델을 내놓 .

2

가장 크게 바뀐 건 모델의 숫자 문치를 통째로 풀어 , 누구나 내려받아 자기 장비에서 돌릴 수 있게 한 열린 모델이야 .

3

그래서 지금은 하나를 쓰는 게 아니라 값 · 속도 · 잘하는 분야를 따져 골라 쓰는 시대야 .

달한 문이 열리기까지

1

누구나 내려받을 수 있던 모델의 크기가 2023 년 2 월엔 650 억 , 그해 7 월엔 700 억 , 2024 년 7 월엔 4,050 억이 됐어 .

2

1 년 반 만에 여섯 배 , GPT-3 의 1,750 억보다 큰 모델을 아무나 받게 된 거야 .

3

2025 년엔 중국 연구팀도 추론 모델을 열어 공개했어 — 열린 모델 경쟁이 한 나라의 일이 아니라는 뜻이야 .

열린 가중치가 여는 것과 안 여는 것

1

여기서 '숫자 뭉치'는 학습으로 정해진 가중치 전체고, 이걸 공개한 모델을 열린 가중치 (open-weight) 모델이라고 불러.

2

가중치가 열려 있으면 누구나 자기 데이터로 추가 학습 (파인튜닝) 을 하거나, 작게 줄여 노트북에서 돌릴 수 있어.

3

다만 가중치를 연다고 학습 데이터나 코드까지 전부 여는 건 아니어서, '오픈' 이 어디까지 여는지는 모델마다 라이선스를 봐야 해.

**열흘은 하나의 길이었다 .
뉴런 하나에서 모두의 모델까지 .**



3 부에서는 이걸로 뭘 할 수 있는지 — 요약 · 번역부터 하나씩 직접 해 본다 .