0

[논문 정리] Outrageously Large Neural Networks : The Sparely-Gated Mixture-of-Experts Layer

ijnuyh · 2025년 4월 19일

통계 수정 삭제

TIL

▾ 목록 보기 11/14

Abstract

정보를 흡수하는 신경망 능력 - 매개 변수의 수에 의해 제한.

네트워크의 일부가 예제 별로 활성화되는 조건부 계산 - 계산의 비례적인 증가 없이 모델 용량을 극적으로 증가시키는 방법으로 이론적으로 제안됨.

But, 실제로는 상당한 알고리즘 및 성능 문제 존재.

결론 :
위 문제를 해결하고

- 조건부 계산의 약속을 실현하여, 현대 GPU 클러스터에서 계산 효율성의 사소한 손실만으로 모델 용량에서 1000배 이상의 개선 달성.
- 최대 수천 개의 feed-forward sub-network로 구성된 a Sparsely-Gated Mixture-of-Experts layer(MoE) 도입.
- Trainable한 gating network는 각 예제에 사용될 MoE 조합을 결정.

MoE - 언어 모델링 및 기계 번역 작업에 적용.

- 여기서 모델 용량은 훈련 코퍼스에서 사용할 수 있는 방대한 양의 지식을 흡수하는 데 중요.
- 최대 1,370억 개의 매개 변수를 가진 MoE가 누적된 LSTM 레이어 사이에 Conv로 적용되는 모델 아키텍처 제시.
- 대규모 언어 모델링 및 기계 번역 벤치마크 성능 - 더 낮은 계산비용, 더 나은 결과 달성.

---

1. Introduction and Related Work

1-1. 조건부 계산

원래 조건부 계산할 때, parameter와 accuracy가 비례함.

컴퓨팅파워와 분산컴퓨팅은 이 수요를 감당할 수 없음.

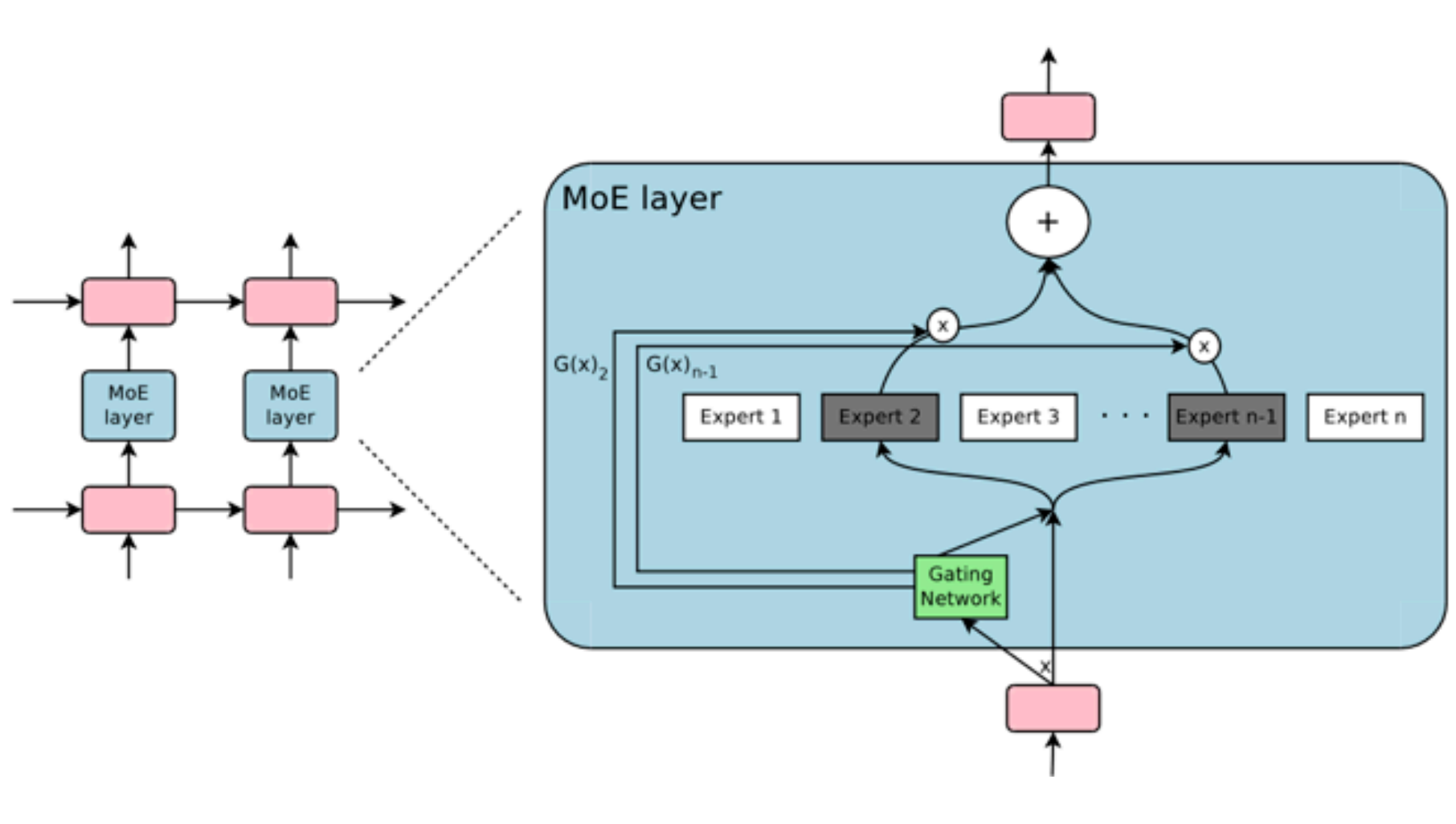

Figure 1: 반복 언어 모델 내에 포함된 MoE 계층.

The sparse gating func. - 계산을 수행할 두 가지의 전문가를 선택. 그들의 출력은 Gating network output에 의해 변조된다.

위 아이디어는 이론적으로는 가능하지만, 실제로 현재까지 모델 용량, 훈련 시간 또는 모델 품질에서 대규모 개선을 입증한 작업은 아직 없음. 여러가지 난제가 있었음.

1-2. Our Approach : The Sparsely-Gated MoE Layer

MoE : number of experts 로 구성, 각각은 간단한 feed-forward 신경망 & input의 처리를 위한 MoE 조합을 선택하는 trainable Gating network로 구성됨.

각 단계의 Network들은 모두 back-propagation에 의해 공동으로 train됨.

"중심 Task : 언어 모델링 & 기계 번역 "

- 쌓여진 LSTM layer 사이에 MoE Conv 를 적용.
- MoE - text의 각 위치에 대해 1번 호출.
- 각 위치에서 potential한 다른 expert 조합 선택.
- 이 potential 한 expeort는 구문과 의미론에 따라 고도로 전문화되는 경향을 가짐.

두 가지 task 벤치마크 모두, 연산은 더 적고, 성능은 더 나음.

1-3. Related work on Mixtrues of Experts

Jacobs et al., 1991; Jordan & Jacobs, 1994으로부터, 20년 전 도입된 동안 expert 접근 방식은 많은 연구의 주제.

SVMs (Collobert et al., 2002), Gaussian Processs (Tresp, 2001; Theis & Bethge, 2015; Deisenroth & Ng, 2015), Dirichlet Processs (Shahbaba & Neal, 2009) 같은 Deep Network들의 다양한 expert architecture 가 제안됨.

다른 작업으로는 *계층 구조(Yao et al., 2009), 무한한 수의 전문가(Rasmussen & Ghahramani, 2002), 순차적 전문가 추가(Aljundi et al., 2016)*와 같은 다양한 expert 구성에 초점을 맞춤.

Garmash & Monz (2016) : 기계 번역을 위한 전문가들의 혼합 형식의 앙상블 모델 제안.

Gating Network : Pre-trained 앙상블 NMT 모델에서 훈련.

- NMT : ***End-to-End 방식의 신경망 기반 기계번역 model***(Neural Machine Translation)

위의 논문들 - Top-Level MoE에 대한 것들. MoE = 전체 모델.

Idea 1. Eigen et al.(2013)

- Multiple MoE 사용 & Deep model의 일부로 자체 gating network를 사용함.
- 복잡한 문제들은 각각 다른 expert를 필요로 하는 하위의 많은 문제들을 포함할 수도 있어서 이 접근방식이 더 강력함.
- 결론에서 희소성을 도입할 것을 암시하며, MoE = 계산을 위한 수단으로 바꿈.

이 논문에서 Genral Neural Network 구성요소로서, MoEs의 이러한 사용을 기반으로 함.

Idea 1의 논문은 두 set의 gating decision을 허용하는 두개의 stacked MoEs를 사용함.

반면, 우리의 MoE의 Conv적용은 MoE가 다른 gating decision을 텍스트의 각 위치에서 허용함.

또한, Sparse gating을 실현하고, model capacity를 게 늘리는 실용적인 방법으로 시연.

---

2. The Structure of the Mixture-of-Experts Layer

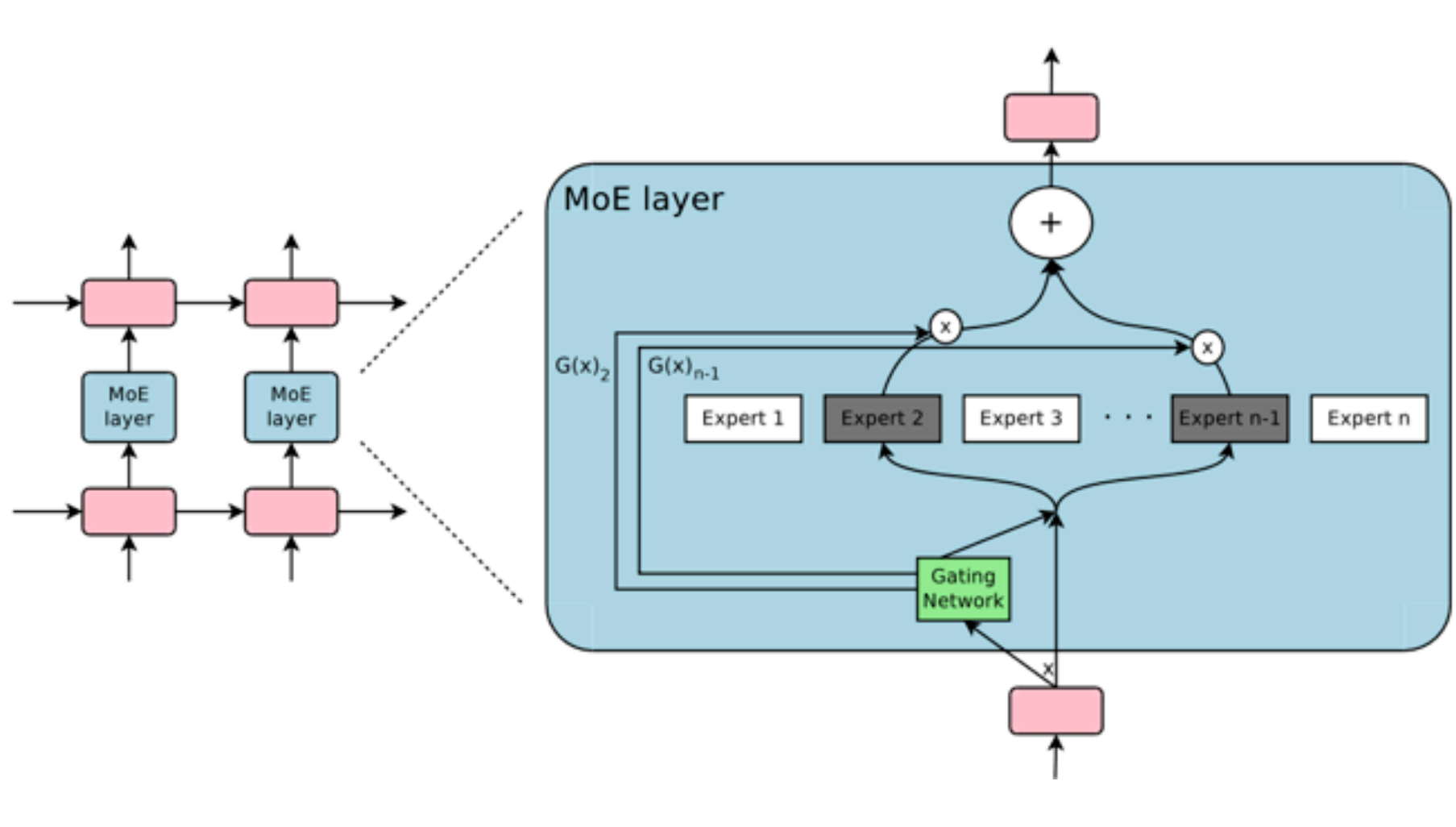

Figure1. MoE module의 개요.

MoE layer : n개의 expert networks, (E1, … , En) , Gating network G (output이 흩어져있는 n차원 벡터) 로 구성.

Expert들은 그 자신이 신경망이고, 그들은 각각 자신만의 parameter들을 가지고 있음.

원칙적으로, expert들이 같은 크기의 input을 받아들이고, 같은 크기의 output을 생성하도록 요구함.

초기 조사에서 우리는 별도의 매개변수를 가졌지만, 동일한 구조를 가진 feed-forward network의 모델로 제한.

1. **Experts Set**
    - MoE 레이어는 n개의 "expert network" 로 구성. (E1,...,En)
    - 각 expert는 동일한 구조(주로 feed-forward 네트워크)를 가지지만, 파라미터는 서로 독립.
    - 모든 expert는 동일한 입력 크기와 출력 크기를 갖도록 설계됩니다.
2. **Gating Network**
    - MoE 레이어에는 추가로 하나의 "gating network" 가 존재합니다.
    - 입력에 대한 차원 벡터 출력.
    - 해당 벡터는 각 expert에 할당되는 가중치(중요도)를 나타냄.
      대부분의 값은 0이 되도록 sparse하게 만듦.
    - 소수의 expert를 선택하고 해당 출력을 가중합 하여 output
    - **입력마다 소수의 expert만 활성화, 나머지 expert는 연산에서 제외되어 효율 극대화.**
3. **MoE layer의 동작 방식**
    - input: x → output : Gating Network : G(x)
    - 각 expert 는 x를 받아 Ei(x) 계산.
    - MoE 레이어의 최종 출력 :

image.png

- G(x)가 0인 expert는 아예 계산하지 않아 연산량을 크게 줄일 수 있음. → 실제로는 수천 개의 expert 중 오직 몇 개(예: 2~4개)만 활성화.

1. **Gating Function & Sparsity 구현**

- Gating Function : Softmax 함수와 noise를 결합한 Noisy Top-k Gating 방식 위주.
  → input 마다 상위 K개의 expert만 선택하여, sparsity를 강제함
  sparse : 신경망의 가중치 행렬에서 대부분의 값이 0이거나 거의 0에 가까운 값일 때.
- 해당 방식은 연산 효율 및 Expert간의 load balancing에도 기여.
- Gating network가 특정 expert만 반복적으로 선택하는 현상을 방지하기 위해, expert 간 중요도와 부하 균형을 위한 추가 loss term을 도입.

2. **Hierarchical MoE 확장 구조**

- Expert가 너무 많을 경우, 2개의 계층으로 확장가능
- 1차 gating network가 그룹을 선택하고, 각 그룹에서 2차 gating network가 세부 expert를 선택하는 방식.

3. **Train 및 활용**

- MoE layer - 전체 Network와 함께 end-to-end 방식으로 사용
- gating network와 expert 모두 backpropagation으로 update.
- MoE 구조 ⇒ Transformer, LSTM 등 다양한 신경망 구조에 삽입해 사용가능.

---

3. Addressing Performance Challenges

기존 - 매개변수 load 및 update의 overhead 때문에 큰 Batch size 가 필요함.

Gating Network가 각 예제에 대해 n명의 전문가 중, K명을 선택하면 , b개의 예시 batch에 대해 kb/n<<B 와 같은 크기의 훨씬 적은 batch를 할당.

Expert 수의 증가에 따라 Navie MoE 구현이 비효율적이게 됨.

batch 축소에 대한 solution → original batch size 🔼

Batch size : forward & backward pass사이의 activation을 저장하는데에 있어 메모리 제한 존재.

→ Batch size를 늘리기 위한 본 논문에서의 제안

1. Mixing Data Parallelism and Model Parallelism:
   데이터와 모델의 병렬성
2. Taking Advantage of Convolutionality:
   conv. layer 활용하기
3. Increasing Batch Size for a Recurrent MoE:
   최근의 MoE에서 batch size 증강

기존 분산 학습 방식:

분산 학습에서는 여러 디바이스(서버, GPU 등)에 모델 복사본을 두고, 각 디바이스가 서로 다른 데이터 배치를 비동기적으로 처리.
각 디바이스의 파라미터는 파라미터 서버를 통해 동기화.

Mixing Data Parallelism and Model Parallelism

데이터 병렬성:
여러 디바이스에 모델 복사본을 두고 각기 다른 데이터 배치를 처리.
모델 병렬성:
모델의 expert를 여러 디바이스에 나누어 배치.

- 모델이 d개의 디바이스에 분산되고, 각 디바이스가 b 크기의 배치를 처리하면, 각 expert는 약 kdb/n개의 예시를 한 번에 받게 됨.
  (k: 각 입력당 활성화되는 expert 수, n: 전체 expert 수)
- expert별 배치 크기를 d배로 늘릴 수 있음.
- **계층적 MoE의 경우,**
    - 1차 gating network - 데이터 병렬
    - 2차 MoE - model 병렬
- MoE : 일반 layer & Gating Network ~ 데이터 병렬로 처리 & expert는 각 디바이스에 1개씩, 모델 병렬로 처리.
- 여러 데이터 병렬 배치에서 MoE 레이어로 들어오는 예시들을 합쳐서 각 expert가 한 번에 더 큰 배치를 처리.
    - 연산 효율🔼, 확장성🔼
- 디바이스 수를 늘리면, expert 수도 늘릴 수 있음
    - 모델 크기를 하드웨어에 맞게 확장가능.

Taking Advantage of Convolutionality:

MoE layer 적용 시, conv 구조의 활용 장점
(시점별로 같은 연산을 적용하는 구조)

확장성:
디바이스 수를 늘리면 expert 수도 비례해 늘릴 수 있어, 모델 파라미터를 크게 확장할 수 있습니다.
전체 배치 크기는 늘어나지만, expert별 배치 크기, 디바이스별 메모리/대역폭 요구량, 학습 속도는 일정하게 유지됩니다.

- In 언어 모델, **MoE 레이어를 이전 레이어의 각 시점(time step) 출력에 동일하게 적용**.
- 이전 레이어가 끝난 뒤, 모든 시점의 출력을 **하나의 큰 Batch**로 묶어 한꺼번에 MoE 레이어에 입력으로 사용 시, input batch → time step 수만큼 커짐.
    - 연산 효율과 하드웨어 활용도 🔼

Increasing Batch Size for a Recurrent MoE

순환신경망(RNN, LSTM)에서의 적용.

- **한계** :
  RNN/LSTM 가중치행렬 - MoE로 대체 시, 각 시점의 MoE입력이 이전 시점의 MoE 출력에 크게 의존
    - Conv적 활용이 불가능함.
      (all time step output = 1 batch)
    - In 순환구조, MoE batch size 증강의 어려움 & 메모리 효율성 🔽

3.2. Network Bandwidth)

"주요 성능 이슈 1: 네트워크 대역폭"

- 이유 : 계산 효율 유지를 위해 expert의 연산량 대비 in/output 크기 비율이 계산 성능 대비 네트워크 대역폭 보다 커야함. (GPU-수천:1)
- MoE 구조에서 네트워크 통신 대부분은 expert의 입력/출력 데이터 전송에 사용됨.
- 연산 효율 확보를 위해 expert의 연산량 대비 입출력 데이터 크기 비율이 커야 함.
- hidden layer 크기🔼 / 개수🔼 → 연산 효율🔼, 네트워크 병목🔽

4.Balancing Expert Utilization

expert 활용 불균형

- gating networ : 소수 expert에만 집중 → 불균형 발생
- **해결책:** 각 expert의 중요도(게이트 값 합계) 변동성을 최소화하는 추가 손실(L_importance) 도입
    - L_importance : 모든 expert의 중요도가 비슷해지도록 유도.
- L_importance = importance coefficient of variation^2 × scaling fator

로드 밸런싱(Load Balancing)

- 위의 중요도 균형만으로는 실제 예시 수 불균형 가능
- **해결책:** expert별 예시 수를 균등하게 하는 추가 손실(L_load) 도입

5. BenchMark

- MoE layer 대규모 언어 모델링 & 기계 번역 task 적용.
- 최대 1370억(137 billion) parameter의 MoE layer -> LSTM stack 사이에 Conv 방식으로 삽입 후 학습.
- 기존 최고 성능(state-of-the-art) 모델 대비 현저히 더 우수한 결과 달성.
- 연산량(계산 비용) : 기존 모델과 비슷하거나 더 적으면서도, 모델 용량(파라미터 수) 1000배 이상 증가
- 언어 모델링 및 번역 task : 주요 지표에서 기존 대비 큰 폭의 개선을 보임.

6. Conclusion

- 조건부 연산(Conditional Computation) 실제 구현
    - 입력별로 네트워크 일부만 활성화
    - 모델 용량 1000배 이상 증가, 연산 비용 소폭 증가
    - 대규모 언어 모델링, 번역 벤치마크에서 SOTA 성능 달성
- 모델 용량과 성능의 비례 관계 강화
    - 데이터 크기·문제 복잡도 증가 시 모델 용량이 성능 향상 핵심
    - 수천 개 feed-forward expert 활용, 대규모 지식 효과적 흡수
- 대규모 분산 학습 확장성
    - 데이터 병렬+모델 병렬 혼합 전략
    - 하드웨어 확장 시 트릴리언(1조) 파라미터 모델 학습 가능성
- 실용적·범용 신경망 컴포넌트
    - LSTM, Transformer 등 다양한 아키텍처 적용 가능
    - 자연어 처리 외 다양한 분야 활용 가능

ijnuyh
ad astra per aspera

이전 포스트
[TIL] vscode JAVA 기본 세팅

다음 포스트
[TIL] npm error - log

0개의 댓글

댓글을 작성하세요

댓글 작성