Mixture of Expert (MoE)

Weilin Cai, et. al, “A Survey on Mixture of Experts in Large Language Models” 2025

1. 서론: AI 스케일링의 새로운 패러다임, MoE

- 대규모 언어 모델(LLM)의 급격한 발전은 인공지능 분야에 전례 없는 혁신을 가져왔지만, 동시에 모델 역량과 계산 비용 사이의 상충 관계라는 근본적인 과제를 부각시켰습니다. 이 문제에 대한 패러다임 전환적 해법으로 '**전문가 혼합(Mixture of Experts, MoE)**' 아키텍처가 부상하며, AI 모델의 성능과 계산 비용 간의 **파레토 최적 경계(Pareto frontier)**를 재정의하고 있습니다.
- AI 분야의 '스케일링 법칙(scaling law)'은 모델 성능이 모델 크기, 데이터셋 규모, 그리고 훈련에 투입되는 계산 능력에 비례하여 향상됨을 명확히 보여주었습니다. 이는 더 크고 강력한 모델을 향한 경쟁을 촉발했지만, **기하급수적으로 증가하는 계산 비용은 지속 가능한 성장에 중대한 장벽**이 되었습니다. 이 문제를 해결하기 위해, 모델 용량을 확장하면서도 계산 효율성을 유지할 수 있는 혁신적인 방법론의 필요성이 절실해졌습니다.
- MoE는 이러한 요구에 부응하는 핵심 아이디어, 즉 '조건부 계산(conditional computation)' 패러다임을 제시합니다. 이는 **모델 전체를 모든 입력에 대해 활성화하는 대신, 모델을 여러 개의 '전문가(expert)' 네트워크로 분할하고, 주어진 입력에 가장 관련성이 높은 전문가들만 선택적으로 활성화하는 방식**입니다. 게이팅 네트워크(gating network)라는 작은 모듈이 어떤 전문가를 활성화할지 결정함으로써, 전체 파라미터 수는 크게 늘리면서도 실제 계산량은 합리적인 수준으로 제어할 수 있습니다.
- 이러한 접근법은 **Mixtral-8x7B, Grok-1, DeepSeek-V2**와 같은 최신 고성능 모델들이 등장하면서 학계와 산업계에서 다시금 활력을 얻었습니다. 이 모델들은 MoE 아키텍처를 통해 수천억 개에서 수조 개에 이르는 파라미터를 보유하면서도, 훨씬 작은 규모의 밀집 모델(dense model)과 유사한 계산 비용으로 우수한 성능을 달성했음을 입증했습니다. 이는 MoE가 단순한 이론적 개념을 넘어, 실제 AI 서비스의 품질과 효율성을 결정하는 핵심 성장 동력이 되었음을 시사합니다.

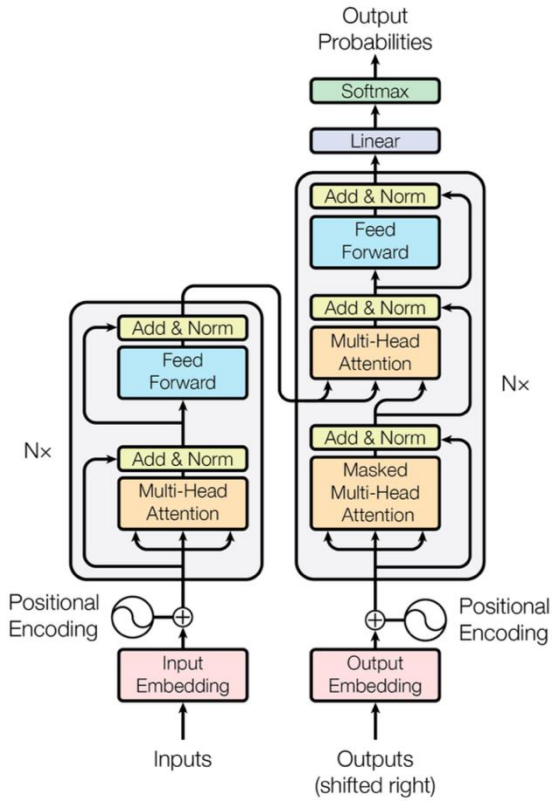


Figure 1: The Transformer - model architecture.

1. 서론: MoE의 장점

구분	설명
효율적 확장 (Efficient Scaling)	예: 100B 파라미터 모델이 있지만, 한 번의 토큰 처리 시 10B만 활성화된다면, **계산량은 10B 모델 수준**이지만 **표현력은 100B 모델 수준**
계산 속도 향상	모든 Expert를 동시에 쓰지 않으므로 **실제 per-token 연산량 감소** (병렬 GPU 환경에서 throughput 증가)
전문화(Specialization)	서로 다른 Expert들이 입력 데이터의 **특정 유형(문법, 수학, 코드, 언어 등)**에 특화되어 학습됨 → **모델 해석성 및 성능 향상**
모델 용량 증가 (Parameter Capacity)	Dense 모델과 같은 FLOPs로 더 큰 파라미터 공간을 확보 가능 → **capacity per FLOP ratio 향상**
확장 유연성 (Scalability)	Expert 수를 늘려도 inference 시 Top-k만 쓰므로 계산량은 일정 → **모델 크기 증가 시 선형 확장 가능**
모듈화된 학습 (Compositionality)	Expert를 독립적으로 학습·교체·재활용할 수 있어 **continual learning / transfer learning**에 유리

모델	공개 구조 정보	MoE 사용 여부	설명
OpenAI GPT-4 / GPT-4o	비공개 (2024~)	✅ **MoE 추정**	OpenAI가 "sparse activation" 언급. GPT-4 논문 유출에서도 16–32 Expert 구조 추정.
Anthropic Claude 3 시리즈	비공개	✅ **MoE 추정**	parameter-to-FLOP 비율이 Dense로는 설명 불가. Expert routing 관련 patent filing 존재.
Google Gemini 1.5 / Bard	일부 공개	✅ **MoE 기반**	GLaM, Switch Transformer의 후속. Google 내부 MoE 전통 계승.
Mistral Mixtral 8×7B (2024)	완전 공개	✅ **명시적 MoE (Top-2 gating)**	상용 API로도 제공. Mixtral 8×7B = 56B parameter, 12B FLOPs 수준 성능.
DeepSeek-V2 (2024, 중국)	논문 공개	✅ **Top-4 MoE 구조**	compute 효율로 유명. "8×MoE experts" 구조.
Llama 3 (Meta, 2024)	공개	❌ **Dense 구조**	8B/70B 모두 Dense. 효율성보다 단순성 우선.
Gemma, Phi-3 (Google/MS)	공개	❌ **Dense 구조**	연구·소형 모델 용도. 추론 효율 중시 아님.

2. MoE의 주요 아키텍처

- MoE 모델의 잠재력을 이해하기 위해서는 그 구조를 구성하는 기본적인 빌딩 블록을 파악하는 것이 중요합니다. 이 섹션에서는 MoE 레이어의 주요 구성 요소, 전문가 네트워크의 역할, 그리고 입력 데이터를 적절한 전문가에게 할당하는 게이팅 메커니즘을 심층적으로 분석합니다. 이를 통해 MoE가 어떻게 계산 효율성과 모델 용량 사이의 균형을 달성하는지 명확히 이해할 수 있습니다.
- **MoE 레이어의 기본 구조**
- Transformer 아키텍처 내에서 MoE 레이어는 일반적으로 자기 주의(self-attention) 하위 레이어 다음에 위치하는 **순방향 신경망(Feed-Forward Network, FFN)을 대체하는 형태로 전략적으로 배치**됩니다. 이러한 배치는 **FFN 계층이 모델 스케일링 시 계산량과 파라미터 수에서 가장 큰 비중을 차지**하기 때문입니다. 예를 들어, 540B 파라미터를 가진 PaLM 모델의 경우, 전체 파라미터의 90%가 FFN 계층에 집중되어 있습니다. MoE는 이 계산 집약적인 부분을 조건부로 활성화함으로써 모델 확장의 병목 현상을 해결합니다.
- MoE 레이어는 두 가지 핵심 요소로 구성됩니다.
    - 1. **전문가 네트워크(Expert Networks)**: 일반적으로 FFN과 동일한 구조를 가진 여러 개의 독립적인 하위 네트워크 집합입니다. 각 전문가는 데이터의 특정 측면이나 패턴을 학습하도록 특화될 수 있습니다.
    - 2. **게이팅 네트워크(Gating Network)**: 입력 토큰을 받아 어떤 전문가 네트워크를 활성화할지 결정하는 작은 신경망입니다. 이 네트워크는 각 전문가에 대한 가중치 또는 확률을 출력하여, 입력 데이터가 어떤 전문가에게 전달될지를 제어하는 '라우터(router)' 역할을 수행합니다.

2. MoE의 주요 아키텍처

- 전문가 활성화 방식: 희소 MoE와 밀집 MoE
- 전문가를 활성화하는 방식에 따라 MoE는 크게 희소(Sparse) 모델과 밀집(Dense) 모델로 나뉩니다. 두 방식은 계산 효율성과 성능 사이에서 서로 다른 트레이드오프를 가집니다.

구분	희소 MoE (Sparse MoE)	밀집 MoE (Dense MoE)
정의	게이팅 네트워크가 입력에 대해 상위 k개의 전문가만 선택하여 활성화하는 방식입니다.	모든 전문가가 각 입력에 대해 항상 활성화되는 방식입니다.
장점	계산 비용을 제어하면서 모델 파라미터 수를 크게 확장할 수 있습니다. 현재 MoE 연구의 주류입니다.	모든 전문가의 지식을 활용하므로 잠재적으로 더 높은 예측 정확도를 달성할 수 있습니다.
단점	게이팅 결정의 이산성(discreteness)으로 인해 최적화가 어렵고, 전문가 용량(expert capacity) 한계로 인해 토큰이 처리되지 않고 버려질(dropped) 수 있음.	모델의 파라미터 수에 비례하여 계산 비용이 직접적으로 증가하므로 대규모 확장에 한계가 있습니다.

2. MoE의 주요 아키텍처

- 밀집 MoE (Dense MoE): 밀집 MoE는 각 입력에 대해 모든 전문가를 활성화하고, 게이팅 네트워크가 산출한 가중치를 각 전문가의 출력에 곱하여 합산합니다. 이 방식은 모든 전문가의 지식을 종합하므로 높은 정확도를 기대할 수 있지만, 전문가 수가 늘어날수록 계산 비용이 선형적으로 증가하여 대규모 모델에는 비실용적입니다.
- 희소 MoE (Sparse MoE): 희소 MoE는 현대 대규모 AI 모델의 핵심이 되는 방식으로, 게이팅 네트워크가 가장 관련성 높은 상위 k개의 전문가만 선택하여 활성화함으로써 계산 효율성을 전략적으로 확보합니다. 이를 통해 모델의 총 파라미터 수는 수조 개까지 확장하면서도, 실제 연산에 참여하는 활성 파라미터 수는 관리 가능한 수준으로 유지합니다.

$$\mathcal{F}^{\text{MoE}}_{\text{dense}}(\mathbf{x};\Theta,\{\mathbf{W}_i\}_{i=1}^{N}) = \sum_{i=1}^{\cdot\cdot} \mathcal{G}(\mathbf{x};\Theta)_i f_i(\mathbf{x};\mathbf{W}_i), \quad (1)$$

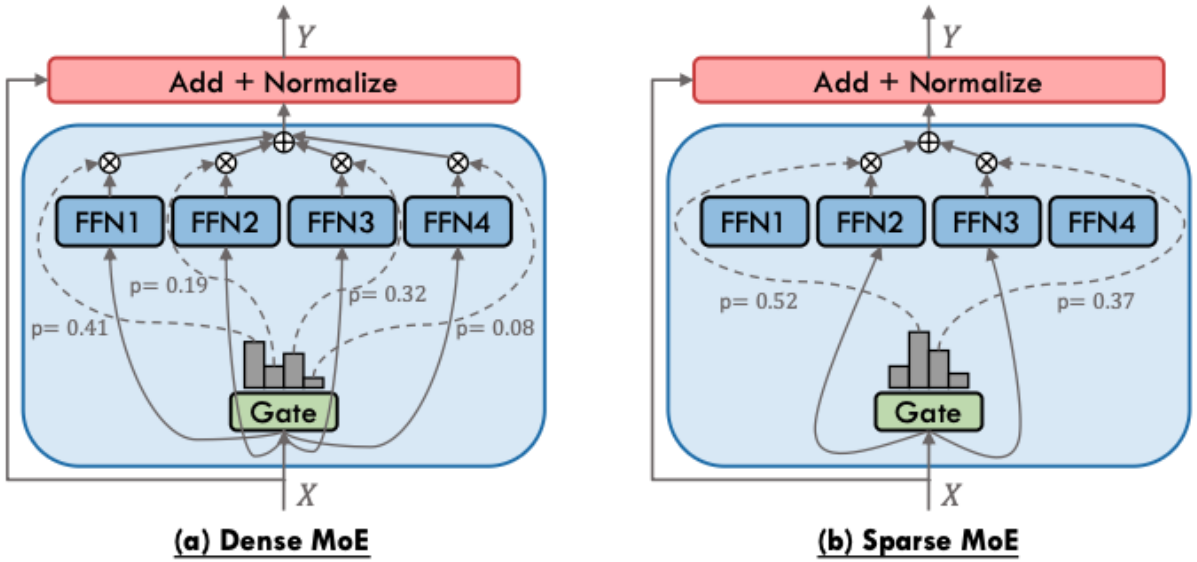
$$\mathcal{G}(\mathbf{x};\Theta)_i = \text{softmax}(g(\mathbf{x};\Theta))_i = \frac{\exp(g(\mathbf{x};\Theta)_i)}{\sum_{j=1}^{N}\exp(g(\mathbf{x};\Theta)_j)}, \quad (2)$$

where $g(\mathbf{x};\Theta)$ represents the gating value prior to the softmax operation.

$$\mathcal{G}(\mathbf{x};\Theta)_i = \text{softmax}(\text{TopK}(g(\mathbf{x};\Theta) + \mathcal{R}_{\text{noise}}, k))_i, \quad (3)$$

$$\text{TopK}(g(\mathbf{x};\Theta),k)_i = \begin{cases} g(\mathbf{x};\Theta)_i, & \text{condition}, \\ -\infty, & \text{otherwise}. \end{cases}, \quad (4)$$

condition : if $g(\mathbf{x};\Theta)_i$ is in the top-k elements of $g(\mathbf{x};\Theta)$. (5)



3. MoE 모델의 알고리즘 설계

- MoE 모델의 성능은 아키텍처에 의해 미리 정해지는 것이 아니라, 알고리즘 설계의 세부 사항 속에서 훈련됩니다. 특히 게이팅 함수는 계산적 순수성(소프트/밀집 방식)과 **압도적인 효율성(희소 방식)** 사이의 근본적인 트레이드오프를 나타내며, 후자는 상당한 최적화 과제에도 불구하고 대규모 모델을 지배하고 있습니다.
- 게이팅 함수(Gating Function) 설계: 게이팅 함수는 어떤 전문가가 특정 입력을 처리할지 결정하고 그 결과를 종합하는 구성 요소입니다. 게이팅 메커니즘은 처리 방법론에 따라 크게 희소(Sparse), 밀집(Dense), 소프트(Soft) 세 가지 유형으로 분류할 수 있습니다.
- **희소 게이팅 (Sparse Gating) :** 희소 게이팅은 MoE의 핵심인 조건부 계산을 구현하는 가장 일반적인 방식으로, 입력 토큰별로 소수의 전문가만 선택하여 활성화합니다.
- (1) **토큰 선택 게이팅 (Token-Choice Gating):** 가장 널리 사용되는 방식으로, 각 입력 토큰이 자신을 처리할 상위 k개의 전문가를 선택합니다. 개념적으로는 간단하지만 특정 전문가에게 토큰이 몰리는 심각한 로드 밸런싱 문제를 야기할 수 있습니다.
- (2) **전문가 선택 게이팅 (Expert-Choice Gating)**: 토큰 선택 게이팅의 로드 밸런싱 문제에 대한 직접적인 아키텍처적 대응으로, 기존 패러다임을 역전시켜 각 전문가가 자신이 처리할 상위 k개의 토큰을 선택합니다. 이 접근법은 전문가별로 처리하는 토큰 수가 고정되므로, 복잡한 보조 손실 함수 없이도 자연스럽게 부하 균형을 맞출 수 있습니다. 그러나 일부 토큰은 어떤 전문가에게도 선택받지 못하고 처리되지 않을 수 있다는 단점이 있습니다.
- (3) **비훈련 토큰 선택 게이팅 (Non-trainable Token-Choice Gating)**: 게이팅 네트워크를 훈련시키지 않고 결정론적인 규칙에 따라 전문가를 할당하는 방식입니다. 예를 들어, 해시 함수를 사용하여 토큰을 전문가에게 무작위로 매핑하거나(Hash Layer), 언어 ID와 같은 사전 정의된 도메인에 따라 전문가를 고정적으로 할당(DEMix)할 수 있습니다. 이는 훈련 복잡도를 낮추지만, 데이터의 미묘한 특성을 학습하는 유연성은 떨어질 수 있습니다.

3. MoE 모델의 알고리즘 설계

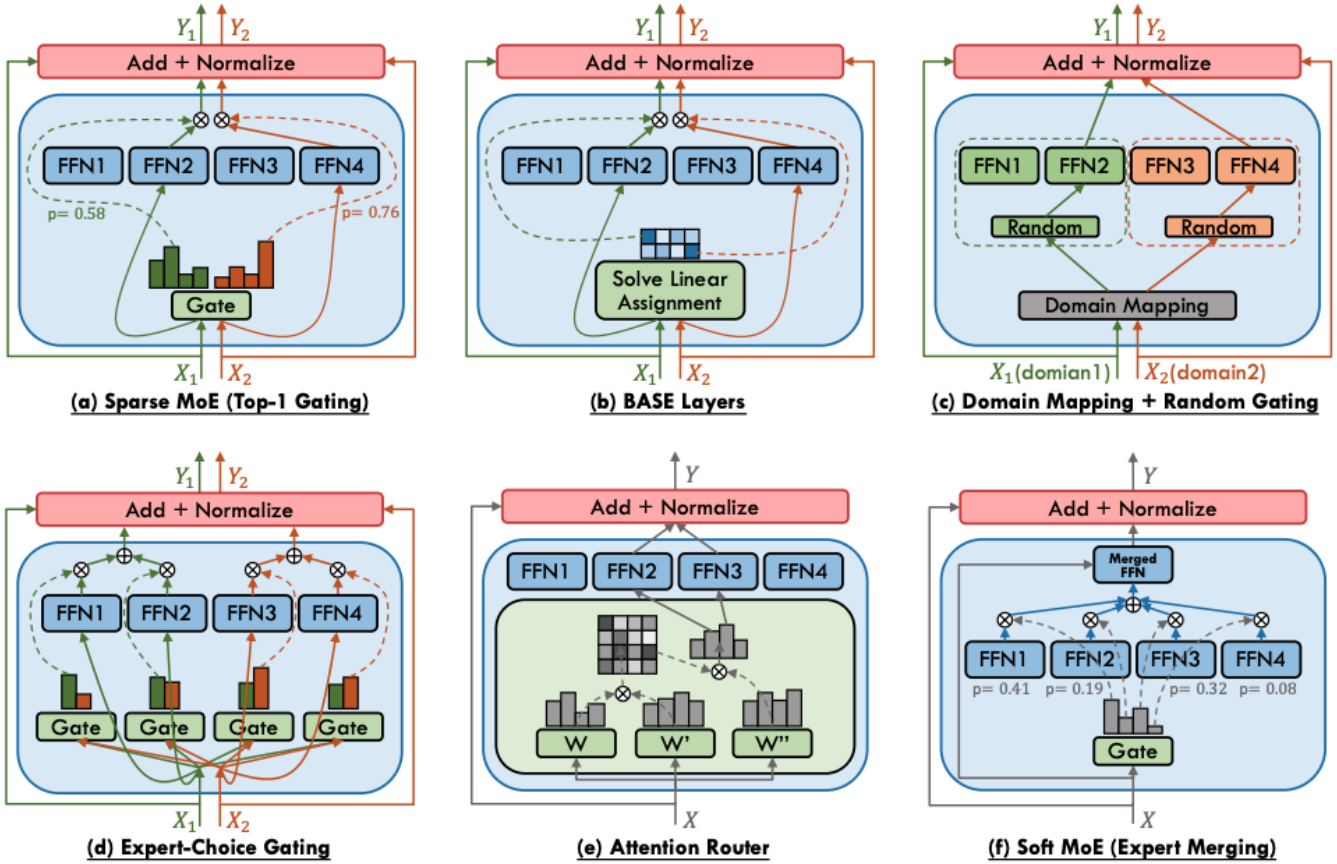
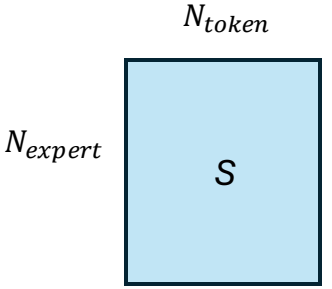


Fig. 4. The illustration of various gating functions employed in MoE models, including (a) sparse MoE with top-1 gating [34], (b) BASE layers [72], (c) the combination of grouped domain mapping and random gating [91], (d) expert-choice gating [92], (e) attention router [82], and (f) soft MoE with expert merging [38].

26페이지로 이동



- Token-Choice Gating :
  - Token이 자신이 처리될 top-k Expert를 선정
  - Score matrix (FFN) 에서 column 별 top-K 선정
- Expert-Choice Gating: Expert:
  - Expert가 내가 가장 잘 처리할 top-k 토큰만 선택
  - Score matrix (FFN) 에서 Row 별 top-K 선정

3. MoE 모델의 알고리즘 설계

- 보조 손실 함수 (Auxiliary Loss Functions): 토큰 선택 게이팅에서 발생하는 로드 불균형 문제를 해결하기 위해 보조 손실 함수가 필수적입니다. 이 함수는 모든 전문가가 비슷한 수의 토큰을 처리하도록 유도하는 정규화(regularization) 역할을 하여 훈련 안정성과 시스템 효율성을 높입니다.

보조 손실 함수	목적	주요 사용 사례
L_importance & L_load	전문가의 중요도(총 게이트 값)와 실제 할당된 토큰 수의 균형을 맞춥니다.	초기 MoE 모델에서 제안되었습니다 (Shazeer et al.).
L_aux	배치 내에서 전문가별로 할당된 토큰의 비율과 게이트 값의 비율 간의 내적을 최소화하여 로드 밸런싱을 유도합니다.	**가장 널리 사용되는 방식**으로, GShard, Switch Transformers, Mixtral 등 다수의 현대 MoE 모델에서 채택되었습니다.
L_z	게이팅 네트워크에 입력되는 로짓(logit) 값의 크기를 제한하여 훈련 안정성을 높입니다.	`L_aux`의 한계를 보완하기 위해 ST-MoE에서 제안되었습니다.
L_MI	전문가와 작업(task) 간의 상호 정보량(Mutual Information)을 최대화하여 전문가 전문화를 촉진합니다.	멀티태스크 학습 환경에서 작업 간 간섭을 줄이기 위해 Mod-Squad에서 사용됩니다.

3. MoE 모델의 알고리즘 설계

- **밀집 게이팅 (Dense Gating)**
- 모든 전문가를 활성화하는 밀집 게이팅 방식은 높은 계산 비용 때문에 대규모 MoE에서는 드물게 사용됩니다. 하지만 LoRA와 같은 PEFT(Parameter-Efficient Fine-Tuning) 기법을 전문가로 활용하는 경우, 각 전문가의 계산 오버헤드가 매우 낮기 때문에 밀집 게이팅이 여전히 유용할 수 있습니다. 이 경우, 여러 LoRA 전문가의 지식을 유연하게 조합하여 특정 다운스트림 작업에 대한 성능을 극대화할 수 있습니다.
- **소프트 게이팅 (Soft Gating)**
- 희소 게이팅은 전문가를 이산적으로 선택하기 때문에 미분이 불가능하여 훈련 시 불안정성을 야기할 수 있습니다. 소프트 게이팅은 이 문제를 해결하기 위해 모든 전문가를 활용하되, 계산 부담을 완화하는 미분 가능한 방식을 사용합니다.
- 토큰 병합 (Token Merging): Soft MoE와 같은 모델에서 사용되는 방식으로, 각 전문가에 대해 모든 토큰의 가중 평균을 계산하여 하나의 '가상 토큰'을 만듭니다. 이 가상 토큰을 각 전문가가 처리하므로, 훈련 안정성이 높고 로드 밸런싱 문제가 자연스럽게 해결됩니다. 하지만 자기회귀적 디코딩 시 미래 토큰에 접근할 수 없어 추론 적용이 복잡하다는 단점이 있습니다.
- 전문가 병합 (Expert Merging): SMEAR, Lory와 같은 모델에서 제안된 방식으로, 게이팅 가중치를 사용하여 모든 전문가의 파라미터를 가중 평균하여 하나의 '가상 전문가'를 동적으로 생성합니다. 이후 모든 토큰이 이 가상 전문가에 의해 처리됩니다. 이 접근법은 완전한 미분 가능성을 유지하면서도 계산 비용 증가를 억제하여 훈련 안정성과 효율성을 동시에 달성합니다. 특히 Lory는 언어 모델의 자기회귀적 특성을 유지하면서 세그먼트 수준에서 전문가 병합을 수행하여, 이 기법을 대규모 언어 모델 사전 훈련에 성공적으로 적용하는 핵심적인 혁신을 이루었습니다.

3. MoE 모델의 알고리즘 설계

- **전문가 네트워크(Expert Network) 설계:**
- **네트워크 유형**:
  - **FFN (Feed-Forward Network):** 전문가로 가장 보편적으로 사용됩니다. FFN은 본질적으로 뉴런 활성화 패턴이 특정 작업과 강한 연관성을 보이는 '모듈성' 현상이 관찰되었습니다. 이는 FFN이 MoE의 '전문가' 역할에 자연스럽게 부합함을 시사합니다.
  - **어텐션 (Attention)**: 어텐션 헤드를 전문가로 사용하는 MoA(Mixture of Attention Heads)와 같은 접근법도 있습니다. 이는 어텐션 메커니즘 자체에 조건부 계산을 도입하여 표현력을 높이려는 시도입니다.
  - **CNN (Convolutional Neural Network)**: 컴퓨터 비전 분야에서는 CNN을 전문가로 사용하여 이미지의 지역적 특징을 전문적으로 처리하는 연구도 진행되고 있습니다.
- **주요 하이퍼파라미터**: 산업계는 전문가 수에 대해 '더 적을수록 좋다'(일반적으로 8-64개)는 접근법으로 수렴하고 있으며, 동시에 MoE 레이어의 빈도를 높이는(종종 모든 FFN을 대체하는) 경향을 보입니다. 이는 Mixtral-8x7B가 대중화한 설계 패턴으로, 얕지만 더 빈번한 라우팅 결정을 통해 파라미터 수를 극대화하면서도 아키텍처의 균일성을 유지하는 전략적 전환을 반영합니다.
  - **전문가 수 (Number of Experts):** 최근 모델들은 8개에서 64개 사이의 비교적 적은 수의 전문가를 사용하는 경향이 있습니다.
  - **전문가 크기 (Expert Size)**: 각 전문가의 FFN 중간층 차원 크기입니다. 최근에는 FFN의 중간 차원을 더 작게 나누어 전문가 수를 늘리는 "세분화된 전문가 분할(fine-grained expert segmentation)" 전략이 채택되고 있습니다. 이 전략은 전문가 간 지식의 분해능을 높이고 전문가 활성화 조합의 유연성을 강화하여 더 정교한 학습을 가능하게 합니다.
  - **MoE 레이어 배치 빈도 (Frequency of MoE Layers)**: Transformer 블록 전체에 MoE 레이어를 배치할지, 아니면 특정 간격으로 배치할지를 결정합니다. Mixtral-8x7B 이후 모든 FFN 레이어를 MoE 레이어로 대체하는 방식이 널리 채택되는 추세입니다.
  - **공유 전문가 (Shared Expert):** DeepSpeed-MoE에서 제안된 개념으로, 게이팅을 통해 선택된 희소 전문가와 함께 모든 토큰을 항상 처리하는 고정된 전문가(공유 전문가)를 두는 방식입니다. 이 아키텍처는 공유 전문가가 공통적이고 작업에 구애받지 않는 지식을 학습하게 하고, 희소하게 활성화되는 전문가들은 더 미묘한 패턴에 특화될 수 있도록 하여, 모델 내에서 효과적인 역할 분담을 창출합니다. 이는 Top-1 게이팅의 통신 비용으로 Top-2 게이팅과 유사한 효과를 내는 매우 효율적인 방법으로, DeepSeekMoE, Qwen1.5-MoE 등 다수의 최신 모델에서 주류 구성으로 자리 잡았습니다.

3. MoE 모델의 알고리즘 설계

- **훈련 및 추론 전략**
- MoE 모델의 효율성과 배포 용이성을 최적화하기 위해, 모델의 구조를 훈련과 추론 단계에서 동적으로 변환하는 다양한 전략이 연구되고 있습니다.
- **밀집-희소 변환 (Dense-to-Sparse)**: 이 전략은 **'희소 업사이클링(sparse upcycling)'**으로 알려져 있으며, 사전 훈련된 고품질의 밀집 모델 체크포인트에서 시작하여 희소 MoE 모델을 초기화합니다. 즉, 밀집 모델의 FFN 가중치를 복제하여 여러 전문가를 만들고, 게이팅 네트워크만 무작위로 초기화한 후 추가 훈련을 진행합니다. 이 방식은 MoE 모델을 처음부터 훈련하는 데 드는 막대한 비용을 절감할 수 있지만, 일반적으로 처음부터 희소 모델로 훈련한 경우보다는 성능이 다소 낮은 경향을 보입니다.
- **희소-밀집 변환 (Sparse-to-Dense):** 대규모 희소 MoE 모델은 추론 및 배포에 부담이 될 수 있습니다. 이를 해결하기 위해, 훈련된 대규모 희소 모델의 지식을 더 작고 배포가 용이한 밀집 모델로 **'증류(distillation)'**하는 방법이 있습니다. 또 다른 접근법은 추론 시에 모든 전문가의 가중치를 평균 내어 단일 FFN으로 변환하는 것입니다. 이는 모델 구조를 단순화하여 하드웨어 호환성을 높이지만, 전문가의 특화된 지식이 일부 희석될 수 있습니다.
- **전문가 모델 병합 (Expert Models Merging)**: 이 접근법은 각기 다른 도메인(예: 과학, 법률 텍스트)에 특화된 여러 사전 훈련된 밀집 모델들을 하나의 MoE 모델로 통합하는 방식입니다. BTX(Branch-Train-MiX)가 대표적인 예로, 각 모델의 FFN을 전문가로 가져오고 어텐션과 같은 다른 계층의 가중치는 평균을 내어 결합합니다. 이후 통합된 데이터셋으로 게이팅 네트워크를 파인튜닝하여, 여러 전문가의 지식을 효과적으로 혼합하는 단일 모델을 만듭니다.

3. MoE 모델의 알고리즘 설계

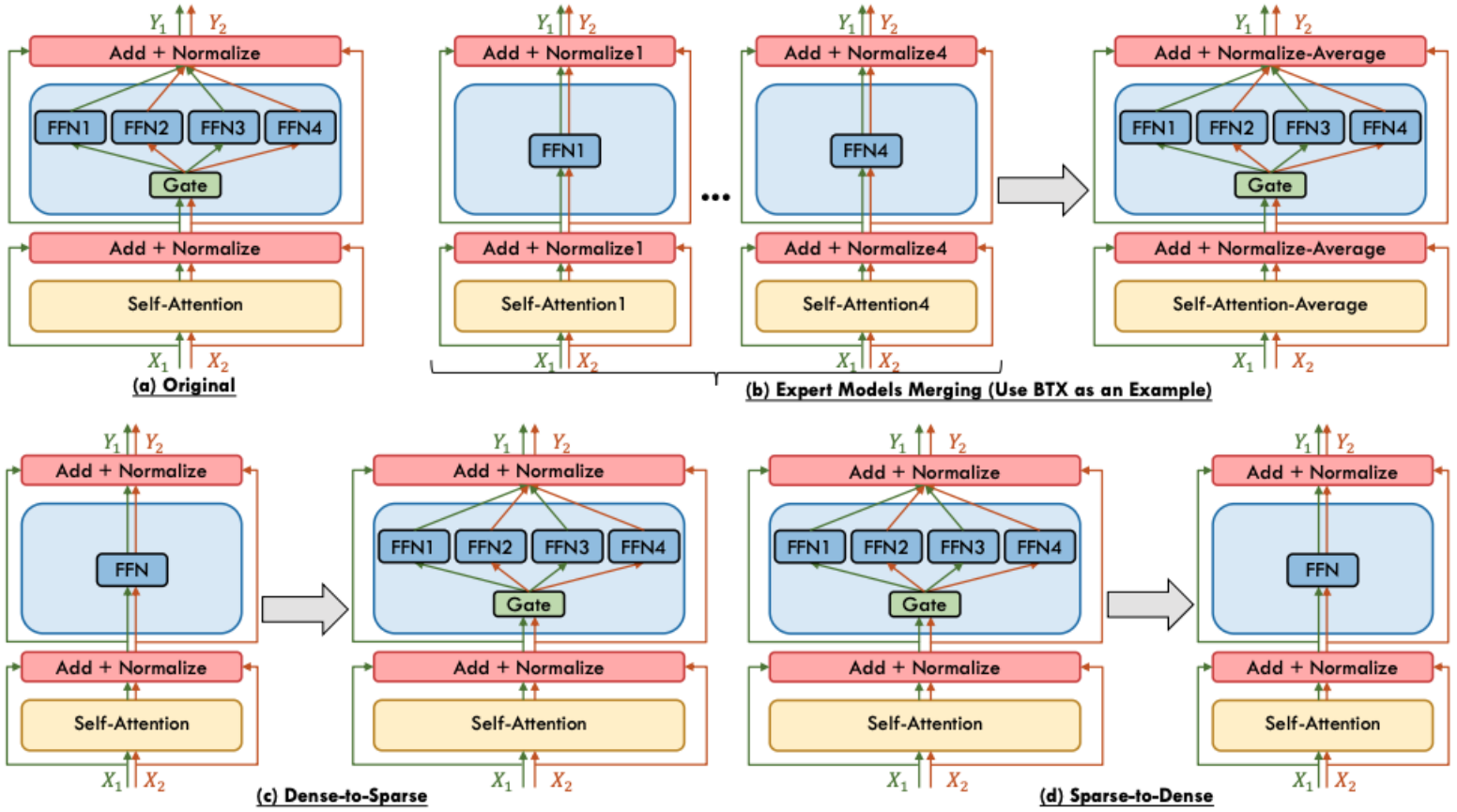


Fig. 7. Schematic representation of training and inference schemes related to MoE. It provides an abstracted view of model transition, without focusing specific model states during training or inference. Subfigure (a) depicts the original scheme without architectural transformation. Subfigure (b) depicts the merging of distinct expert models, exemplified by BTX [52]. Subfigure (c) depicts the transition from a dense model to a sparse model. Subfigure (d) depicts the inverse process, where a sparse model is converted to a dense model.

4. MoE 시스템 최적화

- MoE 모델은 희소하고 동적인 계산 워크로드라는 고유한 특성 때문에 새로운 과제를 제기합니다. 전문가를 여러 장치에 분산시키는 과정에서 발생하는 통신 병목 현상, 전문가별 작업량 불균형으로 인한 계산 비효율성, 그리고 막대한 파라미터로 인한 저장 공간 제약 등이 대표적인 문제입니다.
- **병렬 처리 전략**
- 대규모 MoE 모델을 훈련하기 위해서는 여러 계산 장치(GPU 등)에 모델을 분산시키는 병렬 처리 기술이 필수적입니다.
- **전문가 병렬 처리 (Expert Parallelism)**: 이는 데이터 병렬 처리의 확장된 형태로, MoE를 위한 가장 기본적인 병렬화 전략입니다. 각 전문가는 서로 다른 장치에 할당되고, 어텐션과 같은 비전문가 레이어는 모든 장치에 복제됩니다. 훈련 과정에서 게이팅 네트워크는 각 토큰을 어떤 전문가(즉, 어떤 장치)로 보낼지 결정하고, All-to-All 통신을 통해 토큰들을 해당 장치로 재분배합니다. 전문가 계산이 완료되면, 다시 All-to-All 통신을 통해 결과를 원래 토큰 순서대로 재결합합니다.
- **하이브리드 병렬 처리 (Hybrid Parallelism)**: 전문가 병렬 처리는 단독으로 사용되기보다는 텐서 병렬, 파이프라인 병렬 등 다른 병렬 처리 전략과 결합되어 사용됩니다. 예를 들어, 단일 전문가의 크기가 너무 커서 하나의 장치 메모리에 담을 수 없을 때는 텐서 병렬 처리를 함께 사용하여 전문가 내부를 여러 장치에 분할합니다. 이러한 하이브리드 접근법은 대규모 분산 환경에서 MoE 모델의 확장성과 효율성을 극대화하는 데 필수적입니다.

4. MoE 시스템 최적화

- MoE가 야기하는 주요 시스템 병목 현상은 계산(Computation), 통신(Communication), 저장소(Storage) 세 가지 측면이 있습니다.
- **계산 최적화**: 전문가 간 워크로드 불균형(load imbalance)은 주요한 계산 비효율성의 원인입니다. 특정 전문가에게 토큰이 몰리면 해당 전문가를 할당받은 장치는 과부하가 걸리고, 다른 장치들은 유휴 상태로 대기하게 되어 전체 시스템의 처리량을 저하시키는 동기화 오버헤드가 발생합니다.
  - 해결 방안: 동적 전문가 배치 및 복제, 즉 워크로드 불균형을 완화하기 위해 전문가를 동적으로 재배치하거나, 부하가 높은 전문가를 여러 장치에 복제하여(FasterMoE) 요청을 분산시킬 수 있습니다.
- **통신 최적화**: 전문가 병렬 처리에서 발생하는 All-to-All 통신은 MoE 시스템의 가장 큰 성능 병목 현상으로 꼽힙니다. 이 병목 현상은 현대 LLM에서 널리 사용되는 희소 토큰 선택 게이팅 메커니즘의 직접적인 결과이므로, 확장 가능한 성능을 위해서는 알고리즘-시스템 공동 설계가 필수적입니다.
  - 해결 방안: 계층적 통신 (Hierarchical Communication): 최신 분산 시스템은 노드 내(e.g., NVLink) 통신 대역폭이 노드 간(e.g., Ethernet) 대역폭보다 훨씬 높습니다. 이 특성을 활용하여 All-to-All 통신을 두 단계로 나누어, 노드 내에서는 고대역폭 통신을 최대한 활용하고 비싼 노드 간 통신량을 최소화하는 계층적 통신 전략을 사용합니다.
- **저장소 최적화**: MoE 모델은 수조 개에 달하는 파라미터를 가질 수 있어, GPU의 고대역폭 메모리(HBM) 용량을 쉽게 초과합니다. 이는 모델을 메모리에 로드하는 것 자체를 어렵게 만듭니다.
  - 해결 방안: 오프로딩 (Offloading): 현재 계산에 필요하지 않은 비활성 전문가 파라미터를 GPU 메모리에서 더 용량이 큰 CPU 메모리나 SSD와 같은 보조 저장 장치로 오프로딩하는 전략입니다. SE-MoE, EdgeMoE 등의 시스템은 이 방식을 사용합니다.

5. 다양한 머신러닝 패러다임에서의 MoE 활용

- MoE 아키텍처는 단순히 대규모 언어 모델을 효율적으로 확장하는 기술을 넘어, 다양한 머신러닝 패러다임이 직면한 근본적인 문제들을 해결하는 유연한 프레임워크로 활용될 수 있습니다. 각 전문가가 특정 지식이나 기술을 전담하는 '모듈성'이라는 MoE의 본질적인 특성은 연속 학습, 메타 학습, 멀티태스크 학습, 강화 학습과 같은 고급 머신러닝 분야에서 발생하는 핵심 과제에 대한 효과적인 해결책을 제시합니다.
- **연속 학습 (Continual Learning)** : **파국적 망각 (Catastrophic Forgetting)** 연속 학습은 새로운 작업을 순차적으로 학습하면서 이전에 배운 지식을 잊어버리는 '파국적 망각' 문제를 해결하는 것을 목표로 합니다. 단일 모델의 파라미터를 새로운 작업에 맞게 계속 수정하면 과거 작업에 대한 성능이 급격히 저하되기 때문입니다.
    - MoE의 해결 방안: 지식의 분리 및 보존 MoE는 이 문제를 완화하는 자연스러운 해결책을 제공합니다. 새로운 작업이나 데이터 분포가 등장할 때마다, 기존 전문가의 가중치를 수정하는 대신 새로운 전문가를 추가하거나 특정 전문가를 새 작업에 할당할 수 있습니다. 게이팅 네트워크는 입력 데이터의 특성에 따라 적절한 전문가(오래된 작업 또는 새로운 작업)를 선택하므로, 기존에 학습된 지식은 거의 영향을 받지 않고 보존됩니다.
- **메타 학습 (Meta-Learning)**: **빠른 적응 (Fast Adaptation**) 메타 학습, 즉 '학습하는 방법을 학습'하는 것은 적은 양의 데이터만으로 새로운 작업에 빠르게 적응하는 능력을 모델에 부여하는 것을 목표로 합니다. 이를 위해서는 다양한 작업들 간의 공통적인 구조나 지식을 효과적으로 추출하는 것이 중요합니다.
    - MoE의 해결 방안: 작업 간 관계 모델링 MoE 아키텍처에서 각기 다른 전문가는 다양한 작업이나 도메인에 내재된 핵심 특징이나 하위 기술(sub-skill)을 학습할 수 있습니다. 게이팅 네트워크는 새로운 작업이 주어졌을 때, 이전에 학습된 전문가들을 적절히 조합하여 해당 작업을 해결하는 방법을 빠르게 찾아냅니다. 이는 모델이 작업 간의 관계를 더 효과적으로 포착하고, 새로운 작업에 대한 적응을 촉진하는 원리로 작용합니다. MoE-NPs와 같은 연구는 MoE를 통해 복잡한 작업 분포를 모델링하고 새로운 작업에 대한 일반화 성능을 높이는 가능성을 보여줍니다.

5. 다양한 머신러닝 패러다임에서의 MoE 활용

- **멀티태스크 학습 (Multi-task Learning)**: 부정적 전이 (Negative Transfer) 여러 작업을 동시에 학습하는 멀티태스크 학습은 작업 간 지식 공유를 통해 성능 향상을 꾀하지만, 관련성이 낮거나 상충하는 작업들 사이에서는 오히려 성능이 저하되는 '부정적 전이' 현상이 발생할 수 있습니다.
    - MoE의 해결 방안: 작업별 전문가 조합 최적화 MMoE(Multi-gate Mixture-of-Experts) 아키텍처는 이 문제에 대한 대표적인 해결책입니다. MMoE는 공유된 전문가 풀(pool)을 두되, 각 작업별로 별도의 게이팅 네트워크를 둡니다. 이를 통해 각 작업은 자신에게 가장 유용한 전문가들의 조합을 독립적으로 학습할 수 있습니다. 관련성이 높은 작업들은 비슷한 전문가 조합을 공유하게 되고, 상충하는 작업들은 서로 다른 전문가 조합을 활용하게 되어 작업 간 간섭을 최소화하고 긍정적 지식 공유를 최적화합니다. 추천 시스템에서 클릭률 예측과 구매 전환율 예측을 동시에 수행하기 위해 개발된 PLE(Progressive Layered Extraction) 모델 역시 이러한 MMoE의 아이디어를 확장한 성공적인 사례입니다.
- **강화 학습 (Reinforcement Learning) :** 핵심 과제: 계산 비효율성 및 적응성 한계 복잡한 환경에서 정책(policy)이나 가치 함수(value function)를 학습하는 강화 학습은 종종 방대한 상태 공간으로 인해 계산적으로 비효율적이며, 환경이 변할 때 유연하게 대처하는 데 한계를 보입니다.
    - MoE의 해결 방안: 모듈형 정책 학습 (Modular RL) MoE는 복잡한 정책이나 가치 함수를 여러 개의 간단한 전문가 모듈로 분해하여 학습하는 '모듈형 강화학습' 접근법을 지원합니다. 각 전문가는 상태 공간의 특정 영역(e.g., 특정 지형)이나 특정 기술(e.g., 점프하기, 잡기)을 담당할 수 있습니다. 게이팅 네트워크는 현재 상태(state)를 입력으로 받아 가장 적절한 전문가(또는 전문가 조합)를 활성화하여 행동(action)을 결정합니다. MACE와 같은 아키텍처는 이를 통해 에이전트가 다양한 환경 변화에 더 유연하고 효율적으로 대응할 수 있도록 돕습니다.

6. 주요 응용 분야

- MoE는 특히 모델의 규모가 성능과 직결되는 분야에서 그 진가를 발휘하며, 기존의 한계를 뛰어넘는 새로운 가능성을 열고 있습니다.
- **자연어 처리 (Natural Language Processing) :** MoE는 LLM 분야에서 가장 성공적으로 적용된 기술 중 하나입니다. GShard, Switch Transformer, GLaM과 같은 선구적인 모델들은 MoE를 통해 모델의 파라미터 수를 수조 개까지 확장하면서도, 훈련 및 추론 비용을 기존 밀집 모델 대비 훨씬 낮은 수준으로 유지할 수 있음을 입증했습니다. 최근 등장한 Mixtral-8x7B 모델은 총 47B개의 파라미터를 가지면서도, 추론 시에는 13B 파라미터만 활성화하여 Llama 2 70B와 같은 훨씬 큰 밀집 모델과 동등하거나 우수한 성능을 보여주었습니다.
- **컴퓨터 비전 (Computer Vision):** 이미지 분류 (Image Classification): V-MoE(Vision MoE)는 Vision Transformer(ViT)의 MLP(FFN) 블록을 MoE 레이어로 대체하여 컴퓨터 비전 분야에 MoE를 성공적으로 도입한 대표적인 사례입니다. V-MoE는 이미지의 각 패치(patch)를 다른 전문가에게 라우팅함으로써, 이미지 내의 다양한 객체나 질감을 전문적으로 처리할 수 있습니다. 그 결과, 훨씬 적은 계산량으로 기존의 최첨단(SOTA) 밀집 모델과 동등하거나 더 나은 이미지 분류 성능을 달성했습니다.
- **추천 시스템 (Recommender Systems) :** 멀티태스크 학습 문제 해결: 추천 시스템은 사용자의 클릭률(CTR), 구매 전환율(CVR), 시청 시간 등 여러 목표를 동시에 최적화해야 하는 대표적인 멀티태스크 학습 문제입니다. 이러한 목표들은 서로 관련이 있지만 때로는 상충 관계에 있기도 합니다. MMoE(Multi-gate Mixture-of-Experts)와 이를 발전시킨 PLE(Progressive Layered Extraction)는 추천 시스템 분야에서 MoE가 어떻게 활용되는지를 보여주는 핵심 아키텍처입니다. 이 모델들은 공유된 전문가 풀과 함께 각 작업(task)마다 별도의 게이팅 네트워크를 둡니다. 이를 통해 각 작업은 자신에게 가장 유용한 전문가들의 조합을 독립적으로 학습하여, 작업 간 긍정적 지식 전이(positive transfer)는 극대화하고 부정적 간섭(negative transfer)은 최소화합니다. 이 구조는 공유 지식과 작업별 특화 지식을 효과적으로 분리하여 추천 성능을 크게 향상시킵니다.