귀납 편향 제공을 위한 색채 어텐션 학습

박지원[O1] 이호성[1] 박재현[2] 김선동[2ψ]
광주과학기술원 전자전기컴퓨터공학부[1] 광주과학기술원 AI대학원[2]
parkjohn58@gm.gist.ac.kr, gitpush-force@gm.gist.ac.kr, white314@gm.gist.ac.kr, sundong@gist.ac.kr

Color Attention for Inductive Bias Provision

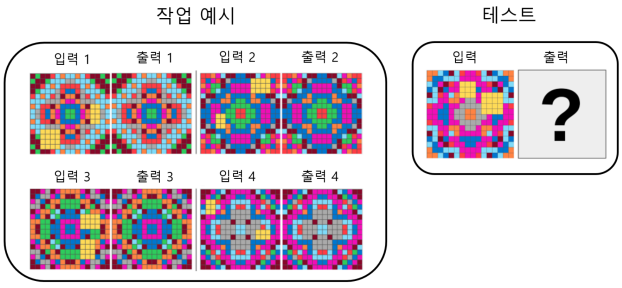
Jiwon Park[1] Hosung Lee[1] Jaehyun Park[2] Sundong Kim[2ψ]
GIST EECS[1] GIST AI[2]

요 약

ARC를 풀기 위해 필요한 사전 지식은 다양하다. 그렇기에 트랜스포머 계열 모델이 이를 학습하기 위해서는 사전 지식이 귀납 편향의 형태로 제공되어야 한다. LatFormer는 '그리드 변환'에 대한 사전 지식을 트랜스포머 모델에 귀납 편향으로 넣어 ARC 문제를 학습한 모델이다. 본 논문에서는 LatFormer에 색채 정보를 귀납 편향으로 제공해 색채 변환을 인식할 수 있게 하는 색채 어텐션을 개발해 도입했다. 색채 어텐션은 마스크 자기-어텐션이 이루어지기 전 입력과 색채를 어텐션하여 색채 변환을 어느 정도 반영할 것인지 계산하는 방식으로 작동한다. 원본 LatFormer와 색채 어텐션이 포함된 LatFormer를 대상으로 색채에 관한 ARC 문제를 규칙에 맞게 증강된 데이터셋을 학습하고 성능을 비교하는 실험을 수행했다.

1. 서 론[1]

ARC(Abstract and Reasoning Corpus)[1]는 Francois Chollet가 인공지능의 일반화 능력을 측정하기 위해 고안한 데이터셋이다. 이 데이터셋에서는 각 문제에 따라 입력과 출력 사이에 특정한 규칙이 존재한다 [그림1]. ARC 데이터셋에서 한 문제를 해결한다는 것은 해당 문제의 작업 예시 쌍에 나타나는 규칙을 활용하여 테스트에서 제시된 입력으로부터 적절한 출력을 도출하는 과정을 의미한다. 문제마다 서로 다른 규칙을 사용하기 때문에 ARC의 모든 문제를 해결하기 위해서는 다양한 범주의 문제 해결 능력이 필요하다. 또한 ARC 데이터셋은 한 문제당 2개에서 5개 남짓의 작업 예시를 제공한다. 그렇기 때문에 학습 과정에서 충분한 데이터의 양이 필요한 모델은 ARC 데이터셋의 문제들을 전부 해결할 수 없다.



[그림 1] ARC 문제의 한 예시. ARC는 작업 예시의 입력과 출력을 바탕으로 규칙을 유추한 후 테스트의 입력에 대한 출력을 예측하는 데이터셋이다.

LatFormer[2]는 ARC 문제 해결을 위해 제안된 모델로 트랜스포머[3]의 학습 과정에서 '그리드 변환'이라는 특정 범주의 도메인 특화 언어(domain specific language)에 대한 귀납 편향(inductive bias)을 제공한다. LatFormer는 도메인 특화 언어로 이동, 회전, 뒤집기를 사용한다. 그러나 ARC 데이터셋을 완벽히 학습하기 위해서는 데이터셋 외부에서 얻어야 하는 추가적인 귀납 편향이 필요하다.

본 논문에서 우리는 LatFormer에 색채 어텐션을 도입하여 색채 정보를 귀납 편향으로 모델에 제공했다. 색채 변환이 필요한 픽셀 위치에 색채와의 어텐션을 적용해 귀납 편향을 제공한다. 본 연구는 트랜스포머 모델에 색채 어텐션을 도입하여 단순히 기본 색채 정보와 프롬프트에 의존하는 기존 트랜스포머의 학습 방식[4]에서 더 나아가 새로운 방법을 시도해 본 것에 의미가 있다. 이러한 색채 어텐션의 작동 방식은 컴퓨터 비전에서 수행하는 작업에도 응용할 수 있다.

2. LatFormer 구조

2.1 마스크 자기-어텐션으로 그리드 변환 사전 지식 주입

직교좌표계 평면 위의 네 개의 정수 좌표 격자점 (1, 1), (3, 2), (2, 3), (5, 5)를 예로 들면, 네 격자점을 동시에 x방향 및 y방향으로 각각 3만큼 이동하면 네 격자점은 (4, 4), (6, 5), (5, 6), (8, 8)로 이동된다. 이와 마찬가지로 격자점들을 90도의 배수만큼 회전 이동하거나 x축, y축으로 거울 대칭 이동해도 격자점들은 정수 좌표에 존재한다. 이처럼 어떠한 변환 후에도 격자점들이 좌표가 모두 정수가 되도록 하는 이동, 회전, 반전들의 모임을 '그리드 변환'이라고 한다.

ARC는 최대 30x30 크기를 가진 정사각 격자의 격자점에 칠해진 색을 추론해 내는 문제로 볼 수 있다. 그중에는 [그림 1]에서 회전 및 반전을 사용하는 것과 같이 그리드 변환을 활용해야 하는 ARC 문제들이 존재한다. 즉, 모델 설계 시 그리드 변환을 귀납 편향으로 사용하면 [그림 1]과 같은 문제를 해결할 수 있다. LatFormer[2]는 마스크 자기-어텐션의 마스크로 이러한 그리드 변환을 귀납 편향으로 활용한다. 각 마스크는 트랜스포머 블록의 입력에 따라 생성되며, 이에 대한 설명은 2.2에서 이어가겠다. 출력된 마스크는 이동, 회전, 반전의 그리드 변환을 나타낸다.

마스크가 적용되는 과정을 이해하기 위해, 스케일드 닷 프로덕트 어텐션(Scaled dot-product Attention)[3]을 살펴보자.

$$MaskAtt(Q,K,V;M) = softmax(\frac{QK^{\top}}{\sqrt{d}} + M)V \quad (1)$$

여기서 Q는 질의, K는 키, V는 값이고 M은 0 또는 $-\infty$(마스킹 값)이다. 이때, M을 마스크 출력을 위해 사용할 경우, $-\infty$으로

[1] 이 논문은 과학기술정보통신부의 재원으로 한국연구재단과 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2023-00240062, RS-2023-00216011, 2019-0-01842)

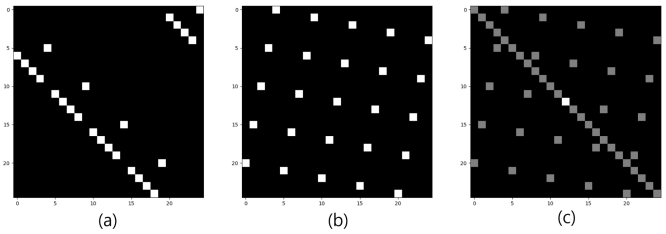
인해 역전파로 최적화할 수 없다. 대신, 소프트맥스 이후에 행렬 원소별 곱으로 마스킹하는 다음 식을 사용하여 M이 미분 가능하도록 했다.

$$MaskAtt(Q,K,V;M) = scale(softmax(\frac{QK^{\top}}{\sqrt{d}}) \odot M)V \quad (2)$$

여기서 $\odot$은 원소별 곱, M은 0 이상 1 이하의 값이며 $scale(\cdot)$은 마스킹으로 인해 변화한 각 행의 합을 1로 수정 하는 스케일 함수이다. 이해를 돕기 위해 $Q = K = V = X$의 열벡터라고 하고 M의 각 행이 단 하나의 1만을 가지되 나머지는 0을 가지는 행렬이라고 하자. 대입하고 식을 정리하면

$$MaskAtt(X;M) = MX \quad (3)$$

이므로, M이 이동, 반전, 회전을 나타내는 행렬이라면 마스크 자기 어텐션의 결과는 X를 이동, 반전, 회전한 벡터가 된다. 5x5의 입력 그리드를 그리드 변환에 따라 변환하는 M의 예시는 [그림 2]에 표현되어 있다. 5x5를 25x1 벡터로 바꾸어 [그림 2]의 마스크에 수식 (3)과 같이 곱해진다.



[그림 2] 5x5 그리드를 회전하는 마스크 예시. 흰색은 1, 회색은 0.5, 검은색은 0을 나타낸다. 각각 (a) 가로와 세로로 1칸 이동, (b) 반시계 90도 회전, (c) 0도 회전과 90도 회전의 혼합을 나타내는 마스크이다.

2.2 Lattice Mask Expert의 마스크 합성 방식

LatFormer의 마스크는 각 트랜스포머 블록에 존재하는 Lattice Mask Expert에 의해 생성된다. Expert의 역할은 각 블록의 입력 데이터를 전달받아 마스크 자기-어텐션에 사용될 마스크를 생성한다. 출력된 마스크는 [그림 2]와 같이 주어진 문제를 푸는 데 필요한 그리드 변환의 종류(이동, 회전, 반전)와 이동 변위, 회전 횟수, 반전 방향에 대한 상세 정보를 포함한다. 그리드 변환의 상세 정보는 다음 과정을 통해 결정된다.

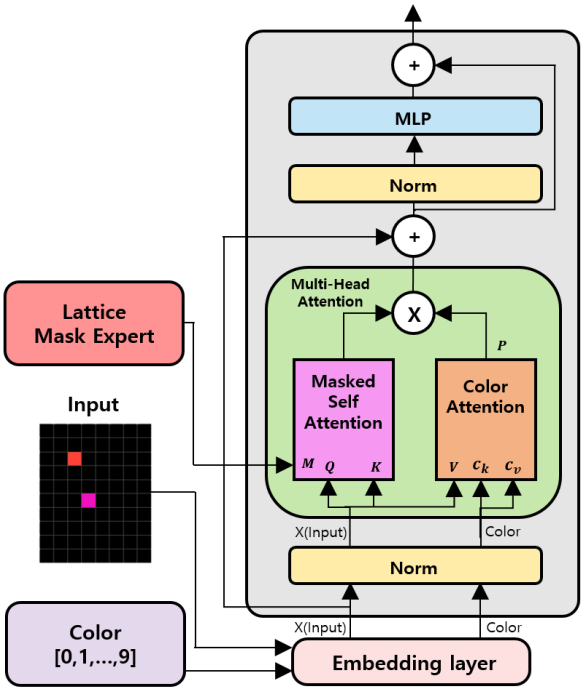
$$M_{t+1} = \alpha f(M_t) + (1-\alpha) M_t \quad (4)$$

M_t는 이전 마스크, $f(M_t)$는 이전 마스크에 특정 그리드 변환을 추가한 마스크이며 M_{t+1}는 출력 마스크이다. 0과 1 사이의 실수 α를 조절하여 그리드 변환의 혼합 정도를 선택할 수 있으며 이는 각 블록의 입력을 전달받는 순방향 신경망을 통해 출력된다. 예시로, $f(\cdot)$가 90도 회전 그리드 변환을 마스크에 추가한다고 하자. M_t가 0도 회전하는 마스크하고 하면 $f(M_t)$는 90도 회전하는 마스크이며, M_{t+1}는 α에 의해 0도(M_t)와 90도($f(M_t)$)를 혼합한 마스크이다. M_{t+2}, M_{t+3} 까지 반복하면 0회~3회 회전하는 마스크를 생성 가능하다. 이 경우, 순방향 신경망은 총 세 개의 α를 출력하여 입력 행렬에서 필요한 회전 횟수를 결정한다. 이를 이동, 반전에 대해서도 적용할 수 있으며, 수식 (4)와 유사한 가중 합 방식으로 각 대칭 요소를 사용하는 마스크를 혼합하면 마스크 자기 어텐션에 사용할 마스크가 완성된다. 실제 구현에선 α를 0과 1 사이의 실수로 출력하도록 하여 [그림 2] (c)와 같이 입력을 회전한 결과와 회전하지 않은 결과를 혼합할 수 있다.

3. 색채 어텐션

3.1 색채 어텐션의 도입 이유

LatFormer는 그리드 변환에 대한 귀납 편향을 마스크 자기-어텐션에 제공하여 ARC 작업 예시를 학습한다. 하지만 이러한 그리드 변환에 대한 귀납 편향만으로는 ARC 데이터셋의 문제를 전부 해결하지 못한다. 그렇기에 이 모델이 해결할 수 있는 문제의 범주를 늘리기 위해서는 그리드 변환과 함께 추가적인 귀납 편향이 주어져야 한다. 그 중, 이 논문에서는 ARC 문제를 푸는 데에 필요한 색채 정보를 귀납 편향으로 모델에 제공하기 위해 색채 어텐션을 새로 도입했다. 이 색채 어텐션을 통해 기존의 모델이 풀 수 없었던 복잡한 색채 관련 문제를 풀 수 있으리라 기대한다.



[그림 3] 색채 어텐션이 추가된 트랜스포머 블록 모식도.

3.2 색채 어텐션의 작동 방식

색채 어텐션을 LatFormer모델에 도입하기 위해, [0, 1,⋯, 9] 행렬을 작업 예시의 입력과 함께 모델의 입력으로 넣어준다. 그 행렬의 이름을 색채 행렬이라고 하자. 두 입력은 같은 임베딩을 거치고, 마스크 자기-어텐션을 위해 트랜스포머 블록에 들어간다. 그 결과, 색채 행렬은 작업 예시의 입력에 따른 색채별 임베딩 정보를 포함하고 있다.

색채 어텐션은 작업 예시의 입력에 해당하는 V가 마스크 자기-어텐션과 곱해지기 전에 이루어진다. 색채 어텐션에 대한 식은 다음과 같으며, 여기서 C_k와 C_V는 각각 색채 행렬이 어텐션을 위해 변환된 값이다.

$$ColorAttention(V, C_k, C_V) = softmax(VC_k^{\top})C_V \quad (5)$$

이 식의 결과값은 V의 각 픽셀에서 0에서 9까지의 색채에 대한 어텐션 값이다. 자기-어텐션과 유사하게 위 과정이 학습되며 V의 각 픽셀에서 정답과 가까운 색채가 무엇인지를 인식하고 이에 가중치를 부여한다. 그 후, 작업 예시의 입력에 색채 변화를 유도하기 위해 V와 색채 어텐션의 결과에 각각 가중치 β와 $1-\beta$를 곱해 더해준 P를 계산한다. 색채 어텐션의 결과를 CA라고 할 때, P는 다음과 같다.

$$P = (\beta V + (1-\beta) CA) \quad (6)$$

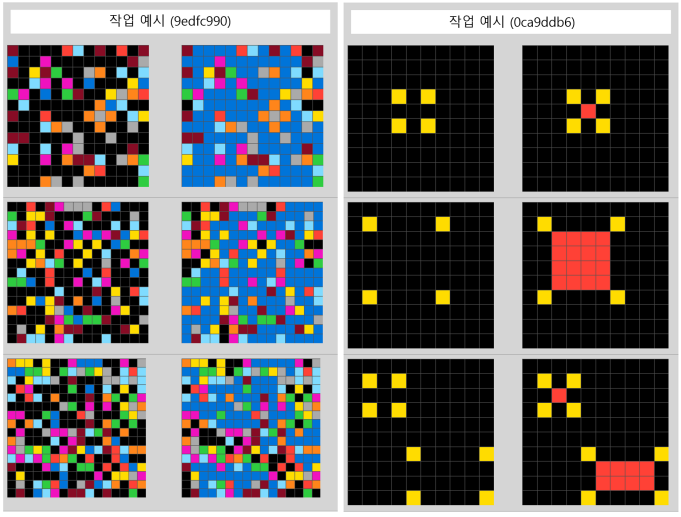
여기서 β는 하이퍼파라미터로, 이 실험에서는 0.9로 설정했다.

최종적으로 색채 어텐션을 추가한 멀티 헤드 어텐션의 출력은 다음과 같다.

$$Output(Q,K,V;M,P) = scale(softmax(\frac{QK^{\top}}{\sqrt{d}}) \odot M)P \quad (7)$$

4. 실험

본 연구에서는 LatFormer에 색채 어텐션을 추가한 모델이 기존 LatFormer 대비 복잡한 색채 문제에 대해서 성능이 얼마나 더 잘 나오는 지를 비교했다.



[그림 4] 실험에 사용된 ARC 데이터셋의 작업 예시이다. 왼쪽은 문제 번호 9edfc990.json이고, 오른쪽은 문제 번호 0ca9ddb6.json이다.

4.1 실험 설계

실험에서 사용된 LatFormer 아키텍처는 [2] 논문에서 설명된 구조와 동일하다. 논문에서 트랜스포머 구조에 대해서 추가적인 설명이 없는 부분은 ViT [5]에서 사용한 방법을 따랐다. 실험에 사용된 LatFormer 모델과 색채 어텐션이 사용된 모델 모두 작업 예시 입력의 픽셀들을 색상에 따라 0부터 9까지의 숫자로 변환한 후 입력으로 사용한다. 비교 실험에 사용된 ARC 문제는 색채에 대한 사전 지식을 요구하는 [그림 4]의 9edfc990와 0ca9ddb6이다. 두 모델의 학습 과정을 확인하기 위해, 문제별로 작업 예시 입출력 간의 규칙을 파악해 그와 같은 규칙을 공유하는 10X10 크기의 입출력 쌍 51,000개를 생성해, 50,000개는 학습 데이터, 1,000개는 검정 데이터로 분류했다. 이때 총 4개의 랜덤시드를 사용해 모델을 학습시켰다.

4.2 성능 지표

이 실험에서 손실 함수는 10X10의 픽셀 하나하나 당 Cross Entropy를 사용했다. 성능 측정 지표로는, 예측한 10X10의 픽셀이 정답과 완벽히 일치하면 정답으로 하는 정확도를 사용했다.

4.3 실험 결과

실험 결과는 [표 1]에서 확인할 수 있다. 각 문제의 데이터셋으로 실험을 한 결과를 분석했다.

0ca9ddb6의 데이터셋에 대해서는 색채 어텐션을 적용한 LatFormer와 원본 LatFormer 모두 학습을 마친 시점에서 랜덤 시드와 무관하게 100%의 정확도를 달성했다. 이는 간단한 색채 정보를 활용하는 문제에 대해, 트랜스포머 모델의 학습에 충분한 데이터양이 제공된다면 추가적인 편향 없이도 효과적으로 학습이 가능하다는 점을 시사한다.

9edfc990의 데이터셋에서 색채 어텐션을 추가한 LatFormer의 평균 성능은 85.77%로, 표준편차는 9.04였으며, 기존 LatFormer의 경우 평균 성능이 88.56%이고, 표준편차는 5.75로 나타났다. 색채 어텐션을 추가한 모델이 기존 LatFormer보다 학습 성능이 더 우수하다는 귀무가설을 설정했을 때, p-value는 0.70로 계산되었다. 따라서 0.95의 신뢰수준에서 귀무가설을 기각함으로써, 색채 어텐션을 추가한 LatFormer의 학습 성능 향상에 대한 증거가 충분하지 않음을 확인할 수 있다.

[표 1] LatFormer와 색채 어텐션을 적용한 LatFormer의 학습 종료 시점 정확도를 95% 신뢰구간에 대해 나타낸 표이다. 0ca9ddb6.json 작업은 두 모델 모두 100%의 성능을 보였다. 9edfc990.json 작업에 대해서 LatFormer는 약 88.56±5.63%, 색채 어텐션을 추가한 LatFormer는 약 85.73±8.87%의 성능을 보였다.

	LatFormer	LatFormer + 색채 어텐션
0ca9ddb6.json	100%	100%
9edfc990.json	88.56±5.63%	85.73±8.87%

5. 결론

이 논문은 LatFormer가 풀 수 있는 ARC 데이터셋의 문제 범주를 늘리기 위해 색채 어텐션을 제시했다. 이를 활용해 LatFormer가 색채 정보에 대한 사전 지식을 학습할 수 있게 했다. 그러나 LatFormer에 색채 어텐션을 추가한 모델이 기존의 LatFormer보다 유의미한 성능 향상을 가져왔는지 확인할 수 없었다. 하지만, 실험을 통해 색채 어텐션을 추가한 LatFormer가 기존 LatFormer보다 유의미한 성능 향상을 보였는지는 확인할 수 없었다. 추후에는 색채 어텐션의 효용을 보이기 위한 실험이 추가적으로 진행될 것이다. 또한, 같은 사전 지식을 공유하는 문제들을 색인으로 묶어서 모델이 학습하도록 하여 여러 사전 지식을 한 번에 학습하는 모델을 개발할 것으로 예상한다.

참고문헌

[1] Francois Chollet, On the measure of intelligence. arXiv preprint arXiv:1911.01547, 2022
[2] Mattia Atzeni et al., Infusing Lattice Symmetry Priors in Attention Mechanisms for Sample-Efficient Abstract Geometric Reasoning, International Conference on Machine Learning, 2023
[3] Vaswani, Ashish et al. Attention is all you need. Advances in Neural Information Processing Systems, 2017
[4] Zheng, Zangwei et al. Prompt vision transformer for domain generalization. arXiv preprint arXiv:2208.08914, 2022
[5]Alexey Dosovitskiy, undefined., et al, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in International Conference on Learning Representations, 2021