

# MAPPO 기반 CNN 하이퍼 파라미터 최적화

마지훈, 조인휘\*

한양대학교 컴퓨터 소프트웨어학과

한양대학교 컴퓨터 소프트웨어학과

mazhixin@hanyang.ac.kr, iwjoe@hanyang.ac.kr

## MAPPO based Hyperparameter Optimization for CNN

Zhixin Ma, Inwhoo Joe\*

Dept. of Computer Science, Hanyang University

Dept. of Computer Science, Hanyang University

### 요약

대부분의 머신러닝 및 딥러닝 모델의 경우 하이퍼 파라미터 선택은 모델의 성능에 큰 영향을 미친다. 따라서 전문가들은 작업을 수행하기 위해 모델을 구축할 때 하이퍼 파라미터 튜닝을 수행하는 데 상당한 시간을 소비해야 한다. Hyperparameter Optimization(HPO)을 해결하기 위한 알고리즘은 많지만 대부분의 방법은 검색을 수행하기 위해 각 epoch에서 실제 실험 결과를 필요로 한다. 따라서 HPO 검색을 위한 시간과 계산 지원을 줄이기 위해 본 논문에서는 Multi-agent Proximal Policy Optimization(MAPPO) 강화 학습 알고리즘을 제안한다. 2개의 이미지 분류 데이터 세트에 대한 실험 결과는 우리의 모델이 속도와 정확성에서 다른 기존 방법보다 우수하다는 것을 보여준다.

### 1. 서론

AI의 핵심 기술로는 머신러닝(ML)과 딥러닝(DL)이 갈수록 주목받고 있다. 최근의 CNN은 1000개의 레이어를 초과할 수 있으며 1억 개 이상의 파라미터를 포함할 수 있다. 매개 변수와 계층 수가 계속 증가함에 따라 교육 시간, 추론 시간 및 모델 크기가 급격히 증가하여 리소스가 제한될 경우 설계 시간이 길어지고 비용이 더 많이 들며 메모리 문제가 발생할 수 있다[1].

효율적인 모델 구축과 하이퍼 파라미터 튜닝을 지원하기 위해 automated ML(AutoML)이 등장했다[2]. HPO 방법에는 Grid Search, Random Search(RS), Bayesian Optimization 등이 포함된다. 이러한 방법은 neural architecture search(NAS)에도 사용되며, Reinforcement learning(RL)도 NAS의 방법의 하나다[3]. HPO는 하이퍼 파라미터의 최적 조합을 선택하는 것으로, 이는 과거 구성과 해당 결과를 바탕으로 최적의 솔루션에 대한 검색 과제로 볼 수 있으며, 이는 RL 접근방식을 사용하는 데 적절하다.

대부분의 교육기관에는 충분한 계산자원이 없으므로 제한된 계산자원의 조건으로 on-policy 알고리즘인 MAPPO는 off-policy 알고리즘에 비해 상당히 높은 알고리즘 run time 효율과 그에 필적하는(또는 더 높은) 데이터 샘플 효율을 가진다. 이를 바탕으로 본 논문에서 우리는 MAPPO 기반 HPO 방법을 제안한다.

### 2. 제안 방법

MAPPO는 PPO[4] 알고리즘의 변형이다. PPO는 고전적인 actor-critic 아키텍처를 사용한다. Actor 네트워크는 로컬 관찰을 수신하여 action을 출력한다. Critic 네트워크는 상태를 수신하여 actor 네트워크의 출력 action을 평가

하는 데 사용된다. MAPPO는 또한 actor-critic 아키텍처를 사용하지만, 다른 점은 critic이 centralized 가치 함수를 배운다는 것이다.

본 논문에서는 centralized training and decentralized execution(CTDE) 방법을 채택한다. 이 프레임워크에서는 각 에이전트가 별도의 actor-critic 네트워크를 가지고 있으며, 에이전트 자신의 actor 네트워크는 훈련 중에 각 critic에 의해 채점될 수 있다는 점에 유의할 필요가 있다. 그림 1은 이 방법의 centralized training, decentralized execution 프레임워크를 보여준다.

CNN의 구조를 지정한 후 CNN의 계층 수는 에이전트 수이다. 각 에이전트의 작업 공간은 해당 계층의 하이퍼 파라미터 수와 해당 값에 의해 정의된다. 액션 공간을 여러 개의 독립된 액션 공간으로 나눔으로써 각 에이전트에서 관찰되는 상태는 학습 과정 전체에서 다른 에이전트와 다르다. 각 에이전트 입력의 상태는 이전 에이전트 출력의 액션으로 간주할 수 있다. 모든 에이전트는 보상을 공유한다. 즉, 모든 에이전트에 대한 보상은 validation set의 정확성이다.

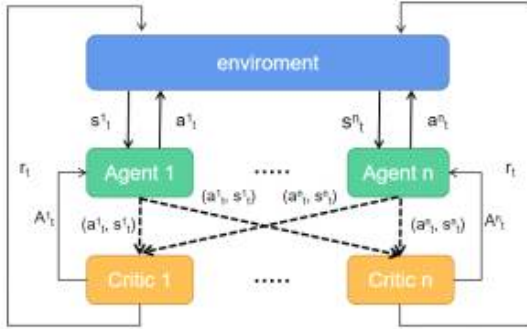
### 3. 실험 결과

공정성을 위해 Intel Core i9-11900K CPU에서 모든 검색 알고리즘을 실행했다. 시간을 절약하기 위해 CNN 모델을 training할 때는 NVIDIA GeForce RTX 3090 GPU를 사용했다.

#### A. 데이터 세트

2개의 이미지 데이터 세트에 대해 방법을 검증했으며 표 1은 이러한 데이터 세트의 세부사항을 보여준다. MNIST 데이터 세트와 Fashion MNIST 데이터 세트는 ML 분야의 고전적인 이미지 분류 데이터 세트이며, 두 데이터 세트

모두 60,000개의 training 샘플과 10,000개의 test 샘플로 구성되며, 각각 28×28픽셀 그레이스케일 이미지이다.



(그림 1) the centralized training and decentralized execution framework

B. 비교 방법

본 논문에서는 RS, Anneal, Tree-structured Parzen Estimators(TPE), Hyperband, BOHB 등의 최적화 알고리즘과 접근 방식을 비교한다.

C. 평가 지표

HPO 방법의 성능은 다음 지표를 사용하여 검증된다.

- time: 방법당 200개의 에피소드를 검색하는 데 필요한 평균 검색 시간.
- accuracy: 테스트 세트에 포함된 각 방법의 평균 정확도.
- kappa:분류의 효과를 측정할 때 사용한다.

D. CNN의 HPO

이 실험에서는 LeNet의 네트워크 구조를 자동으로 설계

표 1 Details of Datasets

Default task	Datasets	#Instances	#Attributes
Image Classification	MNIST	70000	28×28×1
	Fashion	70000	28×28×1
	MNIST	70000	28×28×1

하려고 한다. 손으로 쓴 글꼴을 인식하도록 설계된 LeNet-5는 CNN에서 가장 인기 있는 모델 중 하나이다. LeNet-5는 네트워크를 교묘하게 설계하여 convolution operation, parameter sharing, pooling 등의 연산을 사용하여 기능을 추출함으로써 컴퓨팅 비용을 대폭 절감한다. 그런 다음 분류 인식에 FC 계층을 사용한다. 게다가, 최근의 많은 새로운 CNN 모델은 이러한 네트워크 구조에서 영감을 얻었다.

표 2와 같이 LeNet의 각 레이어의 5가지 하이퍼 파라미터에 대해 설명한다. 실험은 ML 및 DL 모델의 성능을 테스트하는 데 일반적으로 사용되는 MNIST 및 Fashion MNIST 데이터 세트에 대한 LeNet으로 수행되었다.

E. 결과

우리의 방법 및 기타 기존 방법의 평균 시간, 정확도 및 kappa 결과는 표 3과 표 4에 나와 있다. 이 표에서 볼 수 있듯이 우리의 방법은 시간, 정확도 및 카파 측면에서 다른 방법보다 뛰어나다. 시간적인 측면에서는 다른 방법에 비해 우리의 방법이 사용하는 검색 시간이 크게 단축된다. 그림 2는 MNIST 및 Fashion MNIST 데이터셋에 대한 우리 방법의 하이퍼 파라미터 검색 시간이 다른 방법에 비해 절반으로 단축되었음을 명확히 보여준다.

표 2 Hyperparameter of LeNet

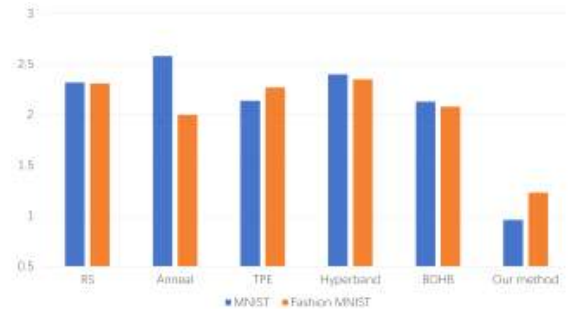
Name	Type	Ranges
conv_size	int	[2, 7]
hidden_size	int	[124, 1024]
batch_size	int	[16,32]
dropout_rate	float	[0.5, 0.9]
learning_rate	float	[0.0001, 0.1]

표 3 Performance on MNIST

Model	acc	time(h)	kappa
RS	0.9923	2.32	0.9914
Anneal	0.9931	2.58	0.9923
TPE	0.9925	2.14	0.9917
Hyperband	<b>0.9942</b>	2.40	0.9936
BOHB	0.9928	2.13	0.9920
Our method	<b>0.9942</b>	<b>0.96</b>	<b>0.9941</b>

표 4 Performance on Fashion MNIST

Model	acc	time(h)	kappa
RS	0.9099	2.31	0.8999
Anneal	0.9203	2.00	0.9114
TPE	0.9170	2.27	0.9078
Hyperband	0.9191	2.35	0.9101
BOHB	0.9178	2.08	0.9087
Our method	<b>0.9222</b>	<b>1.23</b>	<b>0.9137</b>



(그림 2) Search time(h) on MNIST and Fashion MNIST datasets of each method

4. 결론

이 논문에서는 MNIST 및 Fashion MNIST에서 CNN의 HPO를 수행하기 위해 multi-agent PPO를 사용한다. 실험 결과 제안된 방법은 시간과 정확도 면에서 RS, Anneal, TPE, Hyperband 및 BOHB를 능가하는 것으로 나타났다.

참고문헌

[1] Iranfar, Arman, Marina Zapater, and David Atienza. "Multi-agent reinforcement learning for hyperparameter optimization of convolutional neural networks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2021).

[2] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, p. 106622, 2021.[Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950705120307516>

[3] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," arXiv preprint arXiv:1611.01578, 2016.

[4] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).