자동 기계학습(AutoML) 기술 동향

Recent Research & Development Trends in Automated Machine Learning

문용혁 (Y.-H. Moon, yhmoon@etri.re.kr) 스마트데이터연구실 선임연구원
신익희 (I.H. Shin, ihshin@etri.re.kr) 스마트데이터연구실 UST학생연구원
이용주 (Y.J. Lee, yongju@etri.re.kr) 스마트데이터연구실 책임연구원
민옥기 (O.G. Min, ogmin@etri.re.kr) 지능정보연구본부 책임연구원/본부장

ABSTRACT

The performance of machine learning algorithms significantly depends on how a configuration of hyperparameters is identified and how a neural network architecture is designed. However, this requires expert knowledge of relevant task domains and a prohibitive computation time. To optimize these two processes using minimal effort, many studies have investigated automated machine learning in recent years. This paper reviews the conventional random, grid, and Bayesian methods for hyperparameter optimization (HPO) and addresses its recent approaches, which speeds up the identification of the best set of hyperparameters. We further investigate existing neural architecture search (NAS) techniques based on evolutionary algorithms, reinforcement learning, and gradient derivatives and analyze their theoretical characteristics and performance results. Moreover, future research directions and challenges in HPO and NAS are described.

KEYWORDS 자동 기계학습(Automated Machine Learning), 하이퍼 파라미터 최적화(Hyper Parameter Optimization), 신경망 아키텍처 탐색(Neural Architecture Search)

Ⅰ. 자동 기계학습(AutoML)

최근 인공 신경망을 활용한 연구는 이미지, 비디오, 자연어 기반 태스크의 추론 정확도를 높이는 것에서 더 나아가 신경망 최적화 및 자동 구조화 분야로 그 폭을 넓혀가고 있다. 인공 신경망 기반 데이터 분석 태스크는 데이터 탐색, 데이터 전처리/정제, 특징 추출, 모델 선택, 모델 훈련 및 최적화의 단계로 진행된다. 특히 목표 데이터별로 모델 선정, 훈련 및 최적화 단계를 수행하기 위해서는 도메인 전문 지식은 물론 많은 시간과 컴퓨팅 자원이 요구된다. 따라서 목표 데이터 또는 태스크가 다를 경우 이와 같은 작업이 반복 수행되어야 하는 문제점을 극복하고, 더욱 빠른 훈련 및 분

* DOI: https://doi.org/10.22648/ETRI.2019.J.340404
* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행한 연구임(No. 2018-0-00278, 부하분산과 능동적 적시 대응을 위한 빅데이터 엣지 분석 기술 개발).

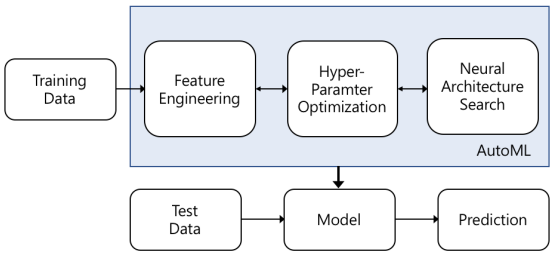


그림 1 자동 기계학습 시스템

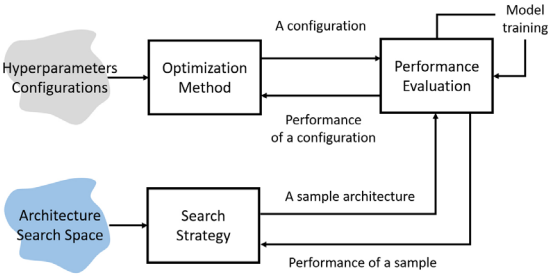


그림 2 하이퍼 파라미터 최적화 및 신경망 아키텍처 탐색

석을 보장하기 위해 많은 AI 관련 연구자들이 자동 기계학습(AutoML: Automated Machine Learning) 기술에 주목하고 있다. 본 기술은 도메인 데이터 특징 추출, 훈련 하이퍼 파라미터 최적화(HPO: Hyper Parameter Optimization) 및 신경망 아키텍처 탐색(NAS: Neural Architecture Search)과 같은 주요 프로세스를 자동화하는 데 이용된다(그림 1).

본 고에서는 자동화 프로세스 중 훈련 고도화를 위해 핵심적으로 요구되는 하이퍼 파라미터 최적화 및 신경망 아키텍처 탐색 연구 분야를 중심으로 주요 기술의 특징과 동향에 대해 살펴보고자 한다.

먼저 하이퍼 파라미터는 머신러닝 및 딥러닝 모델의 입력값으로 해당 모델이 목표 데이터 특성으로부터 일반화된 추론 성능을 훈련할 수 있도록 제어하는 기능을 수행한다. 이러한 하이퍼 파라미터는 학습률, 학습률 스케줄링 방법, 손실함수, 훈련 반복횟수, 가중치 초기화 방법, 정규화 방법, 적층할 계층의 수 등과 같이 모델 훈련 성능에 직접적인 영향을 미치는 다양한 변수들로 구성되어 있다. 즉 개별 변수 조율 방식에 따라 다양한 하이퍼 파라미터 설정이 도출될 수 있어, 이의 최적 조합을 탐색하는 기술이 요구된다(그림 2). 하이퍼 파라미터 최적화 기술로 베이지안 최적화(Bayesian Optimization)가 주류 이론으로 연구됐으나, 탐색의 복잡성을 줄이기 위해 대안 기술이 제안된 바 있다. 최근에는 베이지안 최적화의 속도 향상을 목적으로 하는 연구가 주로 진행되고 있다.

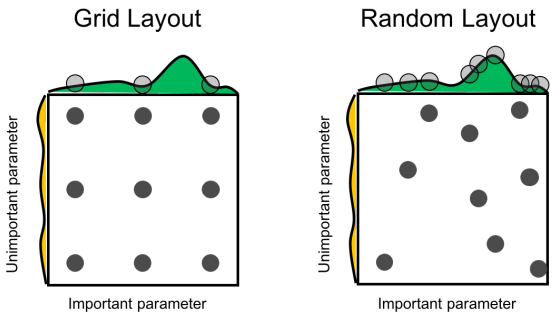
다음으로 신경망 아키텍처 탐색 기술은 목표 데이터와 태스크를 대상으로 가장 효과적으로 훈련될 수 있는 신경망을 자동 생성하기 위해 뉴런 연결구조와 가중치를 탐색 대상으로 삼는다. 신경망 탐색 과정은 탐색 영역 설계, 탐색 최적화 기법 고안, 성능 평가 전략 정의 및 기본 연산(Primitive Operations) 설정으로 구성된다(그림 2). 특히 신경망을 구성하는 단위 구조를 어떻게 설계하는가에 따라 생성 가능한 신경망의 조합수(탐색 영역)가 결정되는 특성이 있어, 탐색 영역 설계가 신경망 자동 탐색 절대적인 복잡도를 결정한다. 본 기술은 구글, 카네기 멜런 대학교에 의해 주도되고 있으며, 주로 합성곱 신경망(CNN: Convolutional Neural Network)을 탐색하는 데 초점을 맞추고 있다.

II. 하이퍼 파라미터 최적화(HPO)

본 장에서는 신경망 하이퍼 파라미터 최적화 기술을 제안된 기법의 관점에서 분류하여 설명한다.

1. 그리드 탐색과 랜덤 탐색

그리드 탐색은 특정 하이퍼 파라미터 구간에서



출처 Reprinted with permission from J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization,"*J. Mach. Learn. Res.*, vol. 13 Feb. 2012, pp. 281-305.

그림 3 그리드 탐색과 랜덤 탐색

일정 간격으로 하이퍼 파라미터값을 선택하여 성능을 측정하고, 가장 높은 성능을 보장하는 하이퍼 파라미터값을 최적해로 도출한다. 본 기법은 구간 전역을 탐색하기 때문에 하이퍼 파라미터의 종류가 많아질수록 탐색 시간이 기하급수적으로 증가한다는 단점이 있다. 또한, 균일한 간격으로 탐색하기 때문에 그림 3의 좌측 예처럼 최적 하이퍼 파라미터값을 찾지 못하는 경우가 발생할 수 있다. 이와 같은 문제를 차단하기 위해 랜덤 탐색[1]이 제안되었는데, 본 탐색 기법은 그리드 탐색과 달리 그림 3의 우측 예처럼 하이퍼 파라미터 구간 내에서 임의로 값을 선택한다. 따라서 불필요한 반복 탐색을 줄여 보다 빠르게 최적 하이퍼 파라미터를 발견할 수 있는 가능성을 높였다.

2. 베이지안 최적화

베이지안 최적화는 알려지지 않은 목적 함수를 최대로 하는 최적해를 찾는 기법이다. 본 알고리즘은 Surrogate 모델과 Acquisition 함수로 구성되는데, Surrogate 모델은 현재까지 조사된 입력값과 함숫값을 바탕으로 목적 함수의 형태에 대한 확률적 추정을 수행하고, Acquisition 함수는 Surrogate 모델의 결과를 이용해 최적해를 찾는 데 유용한 후보를 추천한다. 본 기법을 통해 최적해가 도출되는 과정이 예시[2]된 그림 4를 살펴보면, 먼저, t = 2에서 Surrogate 모델은 현재까지 조사된 (입력값 – 함숫값)을 사용해 입력 x에 따른 표준 편차(파란 음영)와 평균을 추정한다. Acquisition(초록 실선) 함수는 Surrogate 모델의 추정 결과에서 표준 편차가 큰 영역 또는 현재까지 조사된 값들 중 함숫값이 큰 입력 x의 근방을 탐색하여 목적 함숫값을 최대로 만드는 x_t를 예측한다. t = 3에서 x_t를 Surrogate 모델의 새로운 입력으로 반영하여 t = 2에서의 과정을 반복한다. 반복을 거듭할수록 Surrogate 모델에서 입력 x에 따른 표준 편차가 작아지고, 추정된 목적 함수(검정 실선)가 실제 목적 함수(검정 점선)에 가까워지는 것을 확인할 수 있다. 따라서 블랙박스인 목적 함수를 추정하는 최적 입력해의 탐색이 가능해진다.

위와 같은 방식으로 동작하는 베이지안 최적화를 2012년 Jasper Sneok[3]이 하이퍼 파라미터 탐색에 적용한 이후, 최근에는 베이지안 최적화와 Hyperband[4] 기법을 조합한 하이퍼 파라미터 최적화 기술이 등장하고 있다.

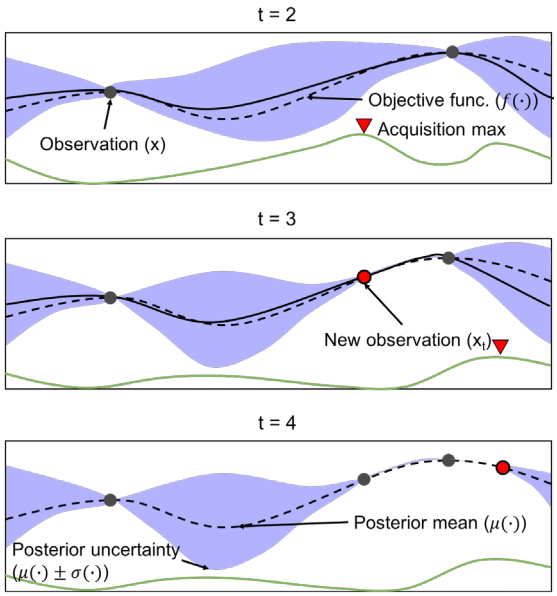
먼저, FABOLAS[5]는 모든 하이퍼 파라미터 집합을 개별적으로 베이지안 최적화할 경우 수일 혹은 몇 주가 걸리는 문제를 해결하기 위해 무작위로 선별된 하이퍼 파라미터 집합과 집합의 크기를 이용하여 최적해를 탐색한다. 본 연구는 멀티 태스크 베이지안 최적화[6]와 엔트로피 탐색[7]을 활용하여 하이퍼 파라미터를 부분 집합으로 나누어 베이지안 최적화를 적용하며, 하이퍼 파라미터 부분 집합과 전체 집합의 상관관계에 기반하여 최적해를 추정한다. 이를 통해 기존 멀티 태스크 베이지안 최적화보다 10배, 엔트로피 탐색보다 100배 빠른 탐색 성능을 보장한다.

Hadrien Bertrand[8]는 베이지안 최적화와 Hyperband를 조합한 탐색 기법을 제안했다. 본 기법은 하이퍼 파라미터 집합에서 하나의 부분 집합을 선택하여 Hyperband로 평가하고 Surrogate 모델로 학습한 후 탐색되지 않은 부분 집합들에 대해 성능 향상 기댓값을 계산하여 정규화를 통해 확률 분포를 구성한다. 이후 확률 분포를 이용하여 다음으로 탐색할 부분 집합을 정하며, 본 과정은 최적해를 탐색할 때까지 반복된다. 본 제안 방식은 베이지안 최적화보다 손실값이 적고, 약 2배 정도 빠르게 최적해를 탐색한다. 또한, Hyperband와 손실값은 유사하지만 약 10배 정도 신속한 탐색 성능을 보여준다.

BOHB[9] 역시 베이지안 최적화 기법와 Hyperband를 조합한 기법인데, 앞서 언급한 연구들과 달리 베이지안 최적화에 Tree Parzen Estimate[10]를 사용하여 간결성과 계산 효율을 증가시켰다. Hadrien Bertrand[8]가 Hyperband를 이용해 무작위로 하이퍼 파라미터 집합을 선택한 방식을 취한 것과 달리 BOHB는 베이지안 최적화의 하이퍼 파라미터 집합 선택 방식을 사용한다. 본 기법에서는 속도 향상을 위해 병렬 처리를 지원하며, 32개의 병렬 워커를 사용할 경우 약 15배의 속도 향상이 보장되었다. 또한, 고차원 Toy Function, SVM, Feed-forward 신경망, 베이지안 신경망, 심화 강화 학습 에이전트, 합성곱 신경망을 대상으로 수행한 실험에서 가장 우수한 성능을 보였다.

3. 그 외 최적화 기법

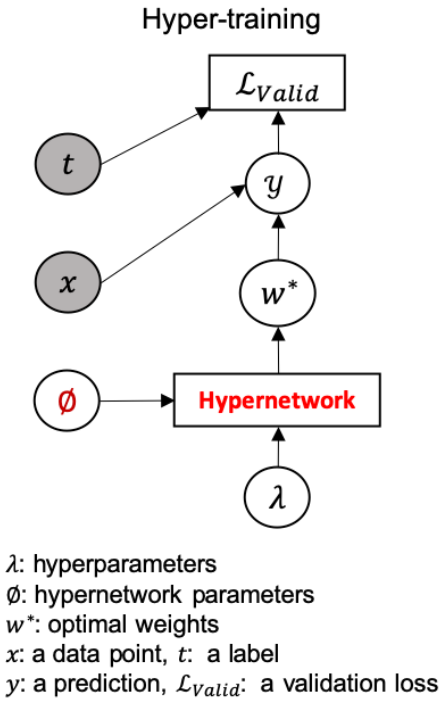
일반적으로 HPO는 하이퍼 파라미터가 주어졌을 때 훈련 손실함수를 최소화하는 가중치를 탐색하는 내부 최적화와 교차 검증 오류를 줄이는 하이퍼 파라미터를 선택하는 외부 최적화로



출처 Reprinted with permission from E. Brochu, V.M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," arXiv preprint arXiv:1012.2599, 2010.

그림 4 베이지안 최적화 기반 최적해 도출 과정

구성된 문제이다. 따라서 하이퍼 파라미터를 변경할 경우 모델 가중치를 재초기화한 후 처음부터 다시 훈련을 수행해야 한다. 이와 같은 낭비를 막기 위해 Hyperband[4]는 훈련이 초기화되지 않고 재개되는 방식을 취했다. 본 기술은 하이퍼 파라미터 선택 문제를 탐험과 착취의 관점에서 최적화가 가능한 Infinite-Armed Bandit 문제로 설정하고, 전체 하이퍼 파라미터 설정(Configuration) 중 더 우수한 조합에 상대적으로 많은 자원(예, 훈련 횟수, 데이터 표본 등)을 적응적으로 할당하여, 균일한 자원할당 전략으로 인하여 효율적 탐색이 어려웠던 SuccessiveHarving[11] 알고리즘의 문제점을 개선하였다. 랜덤 탐색을 이용하여 최적화 속도를 기존 베이지안 최적화 기법보다 5배에서 30배까지 향상시킨 반면, 하이퍼 파라미터의 차원이 증가할 경우 확장성에 문



출처 Reprinted with permission from J. Lorraine and D. Duvenaud, "Stochastic Hyperparameter Optimization through Hypernetworks," arXiv preprint arXiv:1802.09419, 2018.

그림 5 Hyper-training

제가 있는 것으로 평가되고 있다.

캐나다 토론토 대학교 J. Lorraine[12]는 고차원 하이퍼 파라미터로부터 최적 설정을 빠르게 탐색하기 위해 그림 5에 도시된 바와 같이 하이퍼 네트워크(Hypernetwork)와 하이퍼 파라미터를 결합 최적화하는 기법인 Hyper-training[12]을 제안하였다. 본 방식은 하이퍼 파라미터 탐색 영역상에서 현재 설정에 이웃한 설정에 적합하게 근사 최적화된 가중치를 훈련할 수 있을 정도의 하이퍼 네트워크만 사용해도 충분한 탐색을 보장한다. 따라서 기존 SMASH[13]가 상이한 설정에 대한 최적 가중치를 근사화하기 위해 충분히 규모가 큰 하이퍼 네트워크를 요구하는 문제점을 개선할 수 있다.

III. 신경망 아키텍처 탐색

본 장에서는 신경망 아키텍처 탐색 프로세스를 자동화하는 기술을 대표적인 세 가지 탐색 기법의 관점에서 분류하여 설명한다. 각 기법은 탐색 영역을 상이하게 설정하거나 별도의 성능 평가 전략을 사용하는 특징이 있어, 이에 대해서도 논의한다.

1. 진화 알고리즘 기반 탐색

진화 알고리즘(Evolutionary Algorithms)은 그림 6과 같이 임의 후보해 집합(Population)을 생성한 후, '선택-크로스오버-뮤테이션-평가'의 과정을 반복하여 해당 집합을 업데이트함으로써 문제에 따라 정의된 적합성 지표(Fitness)를 가장 잘 만족시키는 해를 탐색한다. 이와 같은 절차를 신경망 구조 탐색에 적용한 NEAT[14] 기법이 2002년 제안된 이후, 최근 진화 알고리즘 기반으로 합성곱 신경망을 자동 탐색하는 제안들이 본격적으로 등장하고 있다.

먼저 구글 브레인의 AmeobaNet[15]은 NASNet[16]이 제안한 탐색 영역을 활용하여, 후보해 집합을 구성한다. 본 기법은 탐색의 효율을 높이기 위해 특정 세대 이상 후보해 집합에 머물러 있는 해를 노후(Aging)의 관점에서 제거하여 후보해 간의 불필요한 경쟁 확률을 줄인다. 또한, 뮤테이션만을 이용하여 부모해를 자식해로 변환함으로써 크로스오버로 인한 최적해 수렴 저해 요소를 제거하였다. 특히 ImageNet을 대상으로 Top-1 및 Top-5 정확도를 각각 83.9%, 96.6% 획득하여 ResNet 계열의 이미지 분류 모델과 대등한 성능을 달성하였다. 그러나 CIFAR-10 및 ImageNet를 대상으로 대략 2만 개의 모델을 탐색하는 데 3,150 GPU Days가 소요되어 매우 큰 연산 자원이 요구되는 문제점

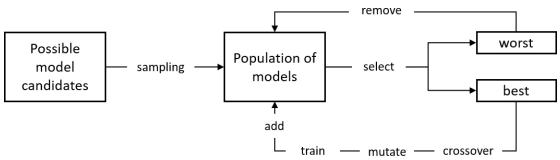


그림 6 진화 알고리즘 기반 탐색

을 안고 있다.

카네기 멜런 대학교와 딥마인드가 협력하여 제안한 Hierarchical NAS[17]는 노드와 엣지로 구성된 계산 그래프를 계층화하여 반복적인 Motifs로 구성된 합성곱 신경망의 탐색 영역을 생성한다. 따라서 임의 후보해는 계산 그래프 형태로 표현되며, 노드는 특징 맵을, 노드 간의 엣지는 연산으로 기능한다. 본 연구에서는 단순화된 뮤테이션을 통해 각 노드 간 엣지에 부여될 연산을 확률적으로 변경하여 더욱 우수한 구조와 가중치의 신경망을 획득할 수 있도록 만든다. 한 가지 주목할 점은 Hierarchical NAS가 AmeobaNet에 비해 짧은 탐색 시간을 보장한다는 것이다. 이는 탐색 복잡도를 결정하는 요소 기술이 바로 탐색 영역 설계라는 것을 의미한다.

2019년 공개된 JASQNet[18]은 신경망 아키텍처와 양자화 비트 수를 동시에 탐색하여 최적의 조합을 결정하는 방법을 제안한다. 즉, 추론 정확도와 모델 경량화의 두 가지 성능을 조율하여 파레토 최적화(Pareto Optimality) 달성이 가능하도록 아키텍처를 탐색한다. 특히, 양자화의 경우 신경망의 각 계층별로 상이한 비트 수를 설정하여 혼합 정밀도(Mixed-Precision) 연산을 지원하는 특징이 있다. AmeobatNet과 동일하게 NASNet의 셀(Cell) 구조를 이용하고 있으며, 양자화 비트 수 추가 탐색으로 인한 복잡도 증가를 고려해야 하는 문제점에도 불구하고, JASQNet 기법은 진화 알고리즘 기반 탐색의 문제점으로 지적됐던 학습 시간을 1~3 GPU Days 이내로 축소하는 실험 결과를 도출하였다.

2. 강화 학습 기반 탐색

강화 학습(Reinforcement Learning) 기법의 에이전트 '액션, 액션 공간, 보상'은 신경망 아키텍처 탐색 문제에서 '탐색, 탐색 영역, 신경망의 성능'으로 각각 연결될 수 있다(그림 7). 이러한 구조적 적합성에 근거하여, 2017년 구글 브레인이 제안한 NASNet[16]은 NAS[19] 기술의 탐색 복잡도를 축소하기 위해 신경망을 구성하는 단위 구조인 블록을 제시하고 제약조건을 두어 탐색 영역을 한정한다. 블록은 NAS 기술의 순환 셀 구조와 유사하게 두 개의 입력을 받아 연산 처리하는 각각의 노드와 이 두 노드의 처리 결과를 병합하여 결괏값을 산출하는 노드로 구성된다(그림 8). 다음으로 순환 신경망(RNN: Recurrent Neural Network) 제어기는 하나의 블록을 구성하는 두 입력값, 두 연산자, 병합 연산자라는 다섯 가지 파라미터를 결정하도록 PPO(Proximal Policy Optimization) 이용하여 훈련된다. 본 제안 기법은 훈련을 5회 반복하여 하나의 합성곱 셀을 탐색하는데, 각 셀은 합성곱 신경망의 한 계층으로 기능한다. 본 기술은 500대의 GPUs를 활용하여 CIFAR-10을 대상으로 신경망 아키텍처 탐색 시 4일의 시간이 소요된다. 본 결과는 기존 NAS기법과 비교했을 때 단위 구조에 제약을 두어 탐색 영역을 한정하는 것의 성능적 중요성을 입증한 것으로 이해될 수 있다.

이후 CPU 기반의 모바일 기기에서 추론의 신속성과 높은 정확도를 조율하여 보장할 수 있는 합성곱 신경망을 자동 탐색하기 위해 MnasNet[20]이 제안되었다. 본 기법은 RNN 제어기를 훈련시켜 NAS와 동일한 구성 요소들을 도출한다. 그러

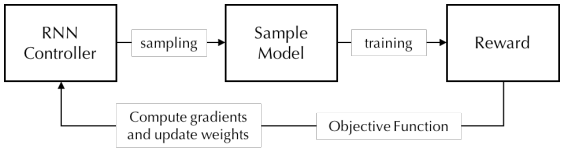
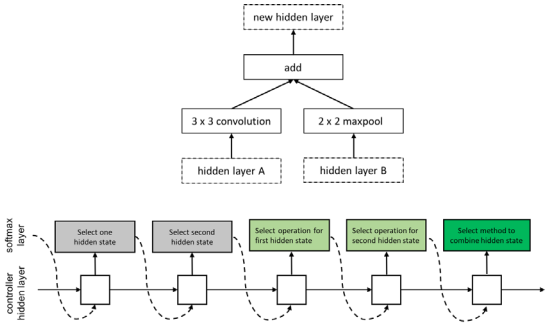


그림 7 강화 학습 기반 탐색

나 탐색 성능을 향상할 목적으로 합성곱 신경망을 여러 블록의 순차적 적층으로 구조화한 후, 개별 블록을 구성하는 복수 계층들의 연산자 종류, 커널 크기, 필터 개수, 계층 개수 등과 같은 세부적인 사항들을 RNN을 통해 산출하도록 탐색 영역을 계층적으로 설계하여, 탐색 영역의 크기를 축소하는 데 기여하였다는 차이점이 있다. 또한, 탐색복잡도를 종전 연구와 같이 FLOPS를 통해 간접적으로 산출한 것이 아니라, 실제 모바일 기기에서 추론 시간을 측정하여 에이전트의 보상 함수에 반영함으로써 탐색의 정확도를 개선하였다. 탐색된 MnasNet 기본 모델은 MobileNetV2[21] 대비 유사한 추론 지연 시간과 2% 우수한 Top-1 정확도를 보장하나, 파라미터 개수와 곱셈 누산의 횟수가 다소 증가한다는 단점을 보인다.

강화 학습 기반 신경망 아키텍처 탐색 기법의 탐색 비효율성을 극복하기 위한 대안으로 ENAS[22]가 구글 브레인, 카네기 멜런 대학교 및 스탠퍼드 대학교 소속 연구자들에 의해 제안되었다. 본 기법의 핵심 아이디어는 장단기 기억(LSTM: Long Short Term Memory) 신경망 제어기로부터 샘플링된 차일드 모델들(Child Models) 간의 가중치 공유를 통해 신규 탐색된 모델이 처음부터 재학습되지 않도록 만드는 것이다. 기존 NAS, NASNet, MnasNet의 경우 모델에서 학습된 가중치의 재활용이 없어 계산 복잡도 문제에 취약했다. 이를 극복하기 위해 ENAS에서는 서로 다른 아키텍처로 이루어진 차일드 모델 간의 가중치 공유가 가능하도록 탐색할



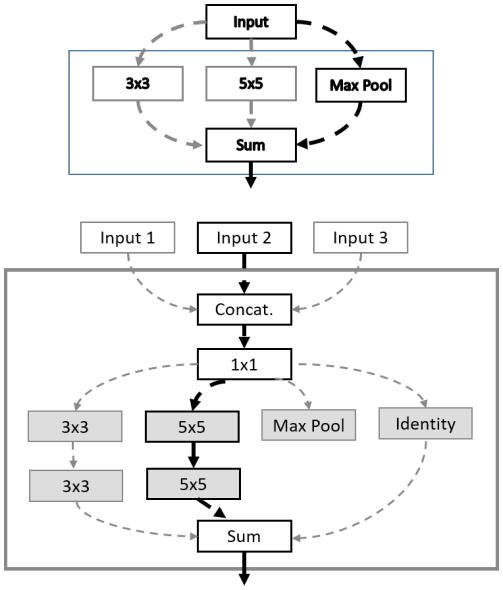
출처 Reprinted with permission from B. Zoph et al., "Learning transferable architectures for scalable image recognition,"in Proc. IEEE CVF Conf. Comput. Vision Pattern Recog.(CVPR), Salt Lake City, UT, USA, June 2018, pp. 8697-8710.

그림 8 NASNet 블록 구조 및 탐색

수 있는 모든 모델을 하나의 DAG (Directed Acyclic Graph)로 표현하고, DAG의 서브 그래프를 샘플링된 차일드 모델로 정의한다. 본 탐색 영역 설계에 따르면 DAG는 모든 모델의 중첩이므로, 이전 모델의 훈련된 가중치 중 일부는 새롭게 샘플링된 모델에 바로 재활용될 수 있다. Pentree Bank 데이터를 대상으로 진행한 실험에서 최신 LSTM 모델에 준하는 테스트 성능을 제공하는 순환 모델(Recurrent Model)이 단일 GPU상에서 10시간을 소요하여 탐색되었다. 특히 CIFAR-10의 경우 DenseNet의 오류율 2.56%에 근접한 2.89%를 달성할 수 있는 합성곱 신경망 구조를 0.45 GPU Days만에 탐색해내는 성과를 보여줬다. 이는 NAS, NasNet, Hierarchical NAS 대비 각각 50,000배, 4,000배, 666배의 탐색 속도 개선을 이룬 것이다.

3. 경사 하강법 기반 탐색

진화 알고리즘 및 강화 학습 기반의 탐색 기법의 대안 연구로서 신경망 연결구조에 대한 학습 및 변형을 위해 이산적이고 미분할 수 없는 탐색 영역을, 연속적이고 미분이 가능한 도메인으로 완화



출처 Reprinted with permission from G. Bender et al., "Understanding and Simplifying One-Shot Architecture Search,"in Proc. Int. Conf. Mach. Learning (ICML), Stockholm, Sweden, 2018, pp. 549-558.

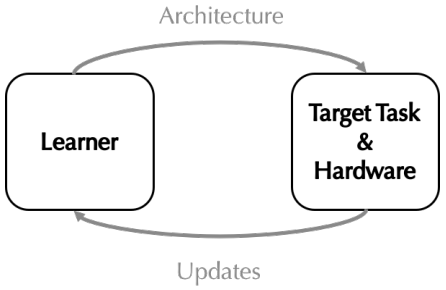
그림 9 One-Shot NAS

(Relaxation)한 기술들이 제안되고 있다[23-25].

구글 브레인에 의해 2017년 제안된 One-Shot NAS[26]은 ENAS처럼 가중치 공유 효과를 통해 탐색 복잡도를 낮추고자 단일한 모델에서 적용 가능한 연산들을 모두 고려하여 훈련하는 원-샷 모델을 제안한다. 이를 통해 여러 신경망 아키텍처들을 모두 학습시키는 효과를 낼 수 있고, 서로 다른 아키텍처들이 같은 가중치 값들을 재활용할 수 있게 된다. 그러나 모든 연산자를 고려할 경우 메모리 효용성이 매우 떨어지는 문제를 해결하기 위해 그림 9와 같이 블록에서 하나의 연산을 선택하고 나머지 가능한 연산들은 제거(Zeroing Out)하는 방식으로 탐색을 진행한다. 탐색 완료 후, 성능이 가장 우수한 아키텍처 모델을 선별하고 해당 모델을 처음부터 다시 훈련하여 목표 데이터에 대해 최적 성능을 제공할 수 있는 신경망으로 생성한다. 본 기법에서 신경망은 동일 구조의 셀로 적층되어 있고, 셀은 하나 이상의 블록으로 구성된다. 목표 데이터를 CIFAR-10으로 설정한 실험에서 본 제안 기법은 ENAS 및 SMASH v2[13]와 유사 정확도를 제공하면서 동시에 파라미터 축소 측면에서 우월한 결과를 도출하였다.

2019년 카네기 멜런 대학교와 구글 딥마인드에 의해 공개된 DARTS[27]은 ENAS의 순환 셀 설계를 차용해 셀을 정의하고, 셀 내부 노드 간에 적용될 수 있는 모든 가능한 연산자들을 혼합 연산자(Mixed Operations)의 형태로 고려한다. 이와 같은 설정은 신경망에서 사용될 연산을 확정짓도록 탐색하는 기존 기법과는 달리 탐색 영역을 연속적 영역으로 완화해 미분할 수 있도록 만든다. 따라서 셀을 구성하는 노드 수를 미리 결정할 경우 노드 간에 적용될 수 있는 여러 연산자 중 확률적으로 가장 우위의 성능을 보장하는 연산자를 경사 하강법을 통해 선정하는 방식으로 신경망 아키텍처 탐색을 수행할 수 있고, 아키텍처를 고정한 상태에서 가중치 역시 경사 하강법을 통해 갱신할 수 있다. 이와 같은 양단식 최적화(Bilevel Optimization)를 반복하여 신경망 구조 및 가중치를 모두 탐색한다. DARTS는 CIFAR-10 분류 문제를 대상으로 탐색 수행 시, 테스트 오류 측면에서 ENAS의 2.89%와 유사한 2.77~2.94%를 획득하였고, ENAS의 0.5 GPU Days에 비해 다소 긴 1.5~4 GPU Days를 요구했으나, 파라미터 개수가 4.6백만 개에서 2.9~3.4백만 개로 개선된 효과를 보였다. 또한, DARTS는 Pentree Bank를 이용한 순환 신경망 탐색 시 ENAS를 능가하는 테스트 Perplexity 성능을 달성하였다.

ProxylessNAS[28]는 One-Shot NAS 및 DARTS와 같은 그래디언트(Gradient) 기반의 신경망 아키텍처 탐색 기법들이 지나치게 일반화된 아키텍처 후보군을 대상으로 탐색을 수행한다는 점으로 인해 GPU 메모리 사용량이 급속도로 증가한다는 문제



출처 Reprinted with permission from H. Cai, L. Zhu, S. Han, "ProxylessNAS: direct neural architecture search on target task and hardware,"in *Proc. Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1-13.

그림 10 ProxylessNAS

에 주목하고 있다. 또한, 기존 연구들이 작은 데이터 셋에 적합하게 탐색된 신경망을 전이학습(Transfer Learning)을 통해 ImageNet과 같은 큰 데이터 셋에 맞게 재학습한 후 탐색 능력의 우수성을 입증하려는 문제점에 대해서도 지적하고 있다. 상기 두 가지 약점을 극복하기 위해 먼저 ProxylessNAS는 셀 내부 노드 간의 혼합 연산자를 설정하고, 복수의 중복된 연결이 노드 간에 형성될 수 있는 것으로 가정한다. 이때 가지치기(Pruning)와 유사한 개념[13,23,29]을 도입하여 개별 연결의 유지 또는 제거를 나타내는 이진 파라미터를 연결마다 할당된 연산자와 함께 훈련의 대상으로 포함한다. 따라서 노드 간 복수 연결 중 연결이 유지되는 연산자들에 대해서만 업데이트를 수행하므로 탐색 시 다뤄야 하는 아키텍처의 크기를 대폭 줄일 수 있어 메모리 효율성을 크게 개선할 수 있다. 이와 같은 장점은 전이학습 없이 큰 규모의 데이터 셋을 직접 훈련하여 이에 최적화된 신경망을 탐색하는 데 기여한다(그림 10). ImageNet을 대상으로 한 실험에서 추론 지연시간을 80ms로 제한한 경우 MobileNetV2, AmeobatNet, MnasNet보다 개선된 정확도를 보장함과 동시에 GPU Hours를 100배 절감하는 효과를 보였다. 더불어, DARTS 대비 10배 적은 GPU 메모리를 사용하여 탐색이 이뤄진 것으로 실험을 통해 검증되었다.

Ⅳ. 결론

본 고에서는 최적화와 탐색의 관점에서 자동 기계학습의 기술 동향에 대해 분석하고, 성능적 특징에 대해 논의하였다.

전통적으로 연구되어온 베이지안 최적화 기법이 최근 기계학습 및 딥러닝 하이퍼 파라미터 최적화에 적용된 이후 탐색 복잡도 문제를 개선한 후속 연구들이 제안되고 있다. 특히 탐색 속도 향상을 목적으로 하는 Hyperband, Hyperband와 베이지안 최적화 병합 기술, Hyper-training 등이 주요 대안 기술로 고려된다. 그 외 게임 이론, PSO(Particle Swarm Optimization), RBF(Radial Basis Function) Surrogate 모델에 기반한 연구들이 존재하나 관련 연구가 활발하지 못한 편이다. 최근에는 하이퍼 파라미터 최적화와 신경망 아키텍처 탐색을 하나의 문제로 풀고자 하는 연구들도 출현하고 있다.

또한, 신경망 아키텍처 탐색 기술은 진화 알고리즘, 강화 학습, 경사 하강법 기반 연구들이 경쟁하는 양상을 보이나, NASNet에서 제안한 셀 구조를 활용해 탐색 영역을 설정하고 검증 데이터(Validation Set)를 이용해 탐색된 신경망의 성능을 판단하는 공통점이 있다. 최근 탐색 영역을 재설계하거나 한정하여 GPU Days를 대폭 개선한 연구 사례가 도출되고 있어, 이를 기반으로 기본 연산자 축소, 양자화 도입, 큰 데이터를 대상으로 직접 탐색 등을 시도하는 후속 논문들이 나타날 것으로 예상된다. 특히 NetScore[30] 관점에서 정확도 대비 파라미터 개수 또는 연산 횟수를 줄이기 위한 기술적 진전이 요구된다.

용어해설

하이퍼 파라미터 최적화 인공 신경망 훈련 시 가장 우수한 성능을 도출할 수 있는 하이퍼 파라미터를 찾아내는 기술을 의미하며, 하이퍼 파라미터로는 학습률, 학습률 스케줄링 방법, 손실함수, 훈련 반복횟수, 가중치 초기화 방법, 정규화 방법, 적층할 계층의 수 등이 고려될 수 있음

신경망 아키텍처 탐색 목표 데이터 및 태스크 (분류, 회귀 등)에 가장 적합한 인공 신경망 구조 및 가중치를 자동으로 탐색하는 기술

약어 정리

AutoML	Automated Machine Learning
FLOPS	Floating Operations Per Second
GPU	Graphics Processing Unit
HPO	Hyper Parameter Optimization
NAS	Neural Architecture Search

참고문헌

[1] J. Bergstra, Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learning Research*, vol. 13, Feb. 2012, pp. 281-305.

[2] E. Brochu, V.M. Cora, N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," arXiv preprint arXiv:1012.2599, 2010.

[3] J. Sneok, H. Larochelle, R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Infor. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, Dec. 2012, pp. 2951-2959.

[4] L. Li et al., "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learning Research*, vol. 18, Apr. 2018, pp. 1-52.

[5] A. Klein et al., "Fast Bayesian optimization of machine learning hyperparameters on large datasets," in *Proc. Artif. Intell. Statistics (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 528-536.

[6] K. Swersky, J Snoek, R.P. Adams, "Multi-task Bayesian optimization," in *Proc. Adv. Neural Infor. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, Dec. 2013, pp. 2004-2012.

[7] P. Hennig, C.J. Schuler, "Entropy search for information-efficient global optimization," *J. Mach. Learning Research*, vol. 13, June 2012, pp. 1809-1837.

[8] H. Bertrand, R. Ardon, I. Bloch, "Hyperparameter optimization of deep neural networks: combining hyperband with Bayesian model selection," in Proc. Conf. sur l'Apprentissage Automatique, France, June 2017, pp. 1-5.

[9] S. Falkner, A Klein, F. Hutter, "BOHB: robust and efficient hyperparameter optimization at scale," in *Proc. Int. Conf. Mach. Learning (ICML)*, Stockholm, Sweden, 2018, pp. 1436-1445.

[10] J. Bergstra et al., "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Infor. Process. Syst. (NIPS)*, Granada, Spain, Dec. 2011, pp. 2546-2554.

[11] K. Jamieson, A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," in *Proc. Artif. Intell. Statistics (AISTATS)*, Cadiz, Spain, 2016, pp. 240-248.

[12] J. Lorraine, D. Duvenaud, "Stochastic Hyper- parameter Optimization through Hypernetworks," arXiv preprint arXiv:1802.09419, 2018.

[13] A. Brock et al., "SMASH: one-shot model architecture search through hypernetworks," in *Proc. Int. Conf. Learning Representations (ICLR)*, Vancouver, Canada, 2018, pp. 1-22.

[14] K.O. Stanley, R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computat.*, vol. 10, no. 2, 2002, pp. 99-127.

[15] E. Real et al., "Regularized evolution for image classifier architecture search," in *Proc. Association Adv. Artif. Intell. (AAAI)*, Honolulu, HI, USA, 2019, pp. 1-16.

[16] B. Zoph et al., "Learning transferable architectures for scalable image recognition," in *Proc. IEEE CVF Conf. Comput. Vision Pattern Recog.(CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 8697-8710.

[17] H. Liu et al., "Hierarchical representations for efficient architecture search," in *Proc. Int. Conf. Learning Representations (ICLR)*, Vancouver, Canada, 2018, pp. 1-13.

[18] Y. Chen et al., "Joint Neural Architecture Search and Quantization," arXiv preprint arXiv:1811.09426, 2018.

[19] B. Zoph, Q.V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, Toulon, France, Apr. 2017, pp. 1-16.

[20] M. Tan et al., "Mnasnet: Platform-aware neural architecture search for mobile," arXiv preprint arXiv:1807.11626, 2018.

[21] S. Mark et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE/CVF Conf. *Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 2018, pp. 4510-4520.

[22] H. Pham et al., "Efficient neural architecture search via parameter sharing," in *Proc. Int. Conf. Mach. Learning (ICML)*, Stockholm, Sweden, 2018, pp. 1-11.

[23] Saxena, Shreyas, Jakob Verbeek. "Convolutional neural fabrics," in *Proc. Adv. Neural Infor. Process. Syst.*, Barcelona, Spain, 2016, pp. 4053-4061.

[24] K. Ahmed, L. Torresani, "Connectivity Learning in Multi-Branch Networks," arXiv preprint arXiv:1709.09582, 2017.

[25] R. Shin, C. Packer, D. Song, "Differential Neural Network Archi-

tecture Search," in *Proc. Int. Conf. Learning Representations (ICLR)*, Vancouver, Canada, 2018, pp. 1-4.

[26] G. Bender et al., "Understanding and Simplifying One-Shot Architecture Search," in *Proc. Int. Conf. Mach. Learning (ICML)*, Stockholm, Sweden, 2018, pp. 549-558.

[27] H. Liu, K. Simonyan, Y. Yang, "Darts: Differentiable Architecture Search," in *Proc. Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1-13.

[28] H. Cai, L. Zhu, S. Han, "ProxylessNAS: direct neural architecture search on target task and hardware," in *Proc. Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1-13.

[29] A. Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 1586-1595.

[30] A. Wong, "NetScore: towards universal metrics for large-scale performance analysis of deep neural networks for practical on-device edge usage," arXiv preprint arXiv:1806.05512, 2018.