특집 03

자동화된 머신러닝 기술 동향: AutoGluon 사례 분석

Isack Thomas Nicholaus · Peter Beatus · 신지용 · 강대기 (동서대학교)

목 차

1. 서 론

자동화된 머신러닝(Automated Machine Learning; AutoML)은 자동 데이터 정리(Automated Data Clean), 자동 피처 엔지니어링(Automated Feature Engineering), 하이퍼파라미터 최적화(Hyperparameter Optimization; HPO), 메타러닝 (Meta-Learning), 그리고 신경 아키텍처 검색(Neural Architecture Search; NAS) 등의 기술로 나누어진다[1]. 본고에서는 이 중 주된 기술들인 HPO와 NAS에 대해 알아보고, 아마존에서 제안한 AutoGluon의 사례를 보이고자 한다.

HPO는 머신 러닝 모델, 하이퍼파라미터(hyperparameters) 및 기타 데이터 처리 단계들을 총체적으로 고려하여, 현재 데이터 세트에 대해 주어진 머신 러닝 모델의 최적의 하이퍼파라미터 구성을 찾는 것을 목표로 한다. 따라서 HPO는 사람을 지루하고 오류가 발생하기 쉬운 하이퍼파라미터 조정 프로세스에서 해방시키고자 하는 것이다. NAS은 머신 러닝 분야에서 널리 사용되는 모델인 인공 신경망(ANN)의 설계를 자동화하는 기술이다. NAS는 손으로 설계한 아키텍처와 동등하거나 성능이 더욱 우수한 네트워크를 설계하고자 하는 목표를 가지고 있다. NAS에 대한 방법은 사용된 검색 공간, 검색 전략 및 성능 추정 전략에 따라 분류할 수 있다.

2. AutoML 프레임워크들 소개

AutoML은 데이터 전처리, 피처 엔지니어링, 모델 선택, HPO, 예측 결과 분석과 같은 반복적인 작업에서 시간과 노력을 아낄 수 있는 좋은 방법이다. 최근에는 오픈소스 AutoML 솔루션이 다양한 분야에 적용되고 있다. 이에 따라 9가지 AutoML 솔루션을 검토하고자 한다.

2.1 Auto-WEKA

AutoML의 초기 노력은 학계와 머신러닝 실무

자들로부터 시작되었다. 첫 번째 시도 중에 하나는 브리티시 컬럼비아 대학교(UBC)에서 개발한 Auto-WEKA이다[2]. 브리티시 컬럼비아 대학교(University of British Columbia)와 프라이부르크 대학교(Albert-Ludwigs-Universität Freiburg)가 WEKA에서 제공하는 알고리즘을 사용하여 개발하였다 [3]. Auto-WEKA는 2개의 앙상블(ensemble) 기법, 10개의 메타 방법, 27개의 기본 분류기 및 각 분류기에 대한 하이퍼파라미터 설정과 특징 선택 기법(3개의 검색 및 8개의 평가 방법 결합)을 제공한다.

2.2 Auto-Sklearn

Auto-Sklearn은 프라이부르크 대학교에서 개발됐으며 scikit-learn을 기반으로 한다[4]. 15개의 분류기, 14개의 특징 추출 기법, 4개의 데이터 전처리 방법을 사용하여 110개의 하이퍼파라미터로 만들어진 가설공간을 생성한다. 이 시스템은 과거의 비슷한 데이터 셋에서의 성능과 최적화 중에 평가된 모델에서 앙상블을 구성하여 자동으로 기존의 AutoML 방법을 개선한다.

2.3 TPOT

TPOT은 펜실베니아 대학교(University of Pennsylvania)에서 개발되었다[5]. TPOT은 트리 기반 파이프라인 최적화 도구(Tree-based Pipeline Optimization Tool; TPOT)를 의미한다. 파이썬으로 작성된 오픈 소스이며 일련의 시뮬레이션 및 실제 벤치마크 데이터 세트에 대한 효율성을 보여준다. 특히 사용자의 입력이나 사전 지식이 거의 또는 전혀 필요하지 않으면서 기본 기계 학습 분석에 비해 상당한 개선을 제공하는 기계학습 파이프라인을 사용한다. 또한, 분류 정확도를 희생하지 않으면서 컴팩트한 파이프라인을 생성하는 파레토(Pareto) 최적화를 통합하여 지나치게 복잡한 파이프라인이 되는 현상을 방지한다.

2.4 Auto-ml

오픈소스 파이썬 패키지인 Auto-ml은 2016년에 출시되었다[6]. 이 AutoML 시스템은 사용 가능한 시스템의 벤치마킹을 포함하여 프로덕션에서 AutoML을 적용할 수 있는 가능성과 한계를 보여준다. Auto-Sklearn, Auto-ml 및 TPOT은 모두 잘 알려진 scikit-learn 머신러닝 패키지를 기반으로 제작되었다.

2.5 AutoKeras

Keras 위에서 실행되는 AutoKeras는 텍사스 A&M 대학교(Texas A&M University)에서 개발되었다[7]. 효율적인 NAS 방법을 위해 네트워크 모피즘(network morphism)에 기반하여 베이지안 최적화를 수행하는 프레임워크이다. 이 프레임워크는 검색 공간을 효율적으로 탐색하기 위해 신경망 커널과 트리 구조의 획득 함수 최적화 알고리즘(tree-structured acquisition function optimization algorithm)을 사용한다.

2.6 H2O AutoML

H2O AutoML(H2O.ai)[8]은 H2O 플랫폼의 머신러닝 모델을 사용하며 H2O.ai[9]에 의해 도입되었다. H2O AutoML의 알고리즘은 H2O 머신러닝 알고리즘의 효율적인 훈련에 의존하여 짧은 시간에 많은 수의 모델을 생성한다. H2O AutoML은 빠른 랜덤 검색과 스택 앙상블의 조합을 사용하여 베이지안 최적화 또는 유전 알고리즘과 같은 더 복잡한 모델 조정 기술에 의존하는 다른 프레임워

크와 비교하였을 때 종종 더 나은 결과를 얻기도 한다. H2O AutoML은 다양한 알고리즘(Gradient Boosting Machine, Random Forests, Deep Neural Networks, Generalized Linear Model)을 훈련하여 후보 모델 전반에 걸쳐 상당한 양의 다양성을 생성하며, 스택 앙상블에서 이를 활용하여 강력한 최종 모델을 생성할 수 있다.

2.7 AutoGluon

아마존(Amazon)은 기계 학습에 대한 진입장벽을 낮추기 위해 AutoGluon을 출시했다[10]. 즉, AutoGluon을 통해서 복잡한 딥 러닝 모델을 설계하는 작업을 몇 줄의 코드로 구현하겠다는 의미이다. 여기서의 작업이란 데이터 형식의 벡터 분류, 모델 아키텍처 설계 및 하이퍼파라미터 최적화를 포함한다. 패키지는 이미지 분류, 객체 탐지, 텍스트 예측에 대한 예제를 제공한다. 시각화나 실험 통계는 포함되어 있지 않지만 AutoGluon API는 scikit-learn과 유사하게 설계되어 이해하기가 수월하다.

2.8 Neural Network Intelligence (NNI)

NNI는 마이크로소프트(Microsoft)에서 머신러닝을 모든 사람들이 동등하게 접근할 수 있도록 하기 위해 도입했다. 해당 프레임워크는 피처 엔지니어링 자동화(경사 기반 검색 알고리즘 사용), 모델 아키텍처 및 압축, 하이퍼파라미터 튜닝 실험 및 디스패치를 할 수 있는 기능들을 포함하는 툴을 제공한다. 또한, 이 프레임워크는 로컬 머신, 원격 머신, Kubeflow, Azure Machine Learning을 포함한 기타 하이브리드 클라우드와 같은 다양한 환경에서도 구동이 가능하다.

NNI는 command line interface(CLI), 파이썬 API 및 웹 GUI를 포함한 여러 머신러닝 프레임워크 및 라이브러리(scikit-learn, TensorFlow, PyTorch, Apache MXNet, XGBoost 등)에서도 동작한다. 또한, 실행하기 쉽고 간단한 명으로 시도할 수 있는 몇 가지 예제 시나리오를 제공한다.

2.9 AutoGL

AutoGL(Auto Graph Learning)은 칭화대학교에서 그래프 기반 문제를 해결하기 위한 도구로 개발되었다[11]. AutoGL을 사용하면 피처 엔지니어링, 모델 교육 및 앙상블, HPO를 자동화할 수 있다.

AutoGL은 PyTorch 위에서 작동하며 전체 프로세스를 완료하는 제한 시간을 매개변수로 가지고 있는 해결사(Solver)를 생성한다. Graph Boosting 및 배깅(Bagging)과 같은 기능과 링크 예측으로 인해 이 도구는 현재 사용하기에 흥미롭고 미래에도 유망하다.

3. AutoGluon 구조 분석

이러한 다양한 AutoML 라이브러리들을 일일이 다루는 것은 지면상 어려우므로, 본고에서는 본 연구팀이 최근에 분석하고 있는 AutoGluon에 대해 조금 더 심층적으로 다뤄보고자 한다.

AutoGluon은 사용이 용이하며 확장 가능하게 만들어진 AutoML 오픈소스 라이브러리이다. 자동 스택 앙상블, 딥러닝을 사용하여 현실의 텍스트, 이미지, 테이블 형식의 데이터를 사용한 문제를 해결하는 것에 중점을 두고 있다. AutoGluon은 자동 하이퍼 파라미터 튜닝과 모델 선택 및 앙상블, 아키텍처 검색, 그리고 데이터 프로세싱에서 이점을 가지고 있다. 이것을 이용하면 맞춤형 모델과 데이터 파이프라인을 쉽게 개선하거나 튜닝할 수 있다.

3.1 구현 로직

AutoGluon은 (그림 1)과 같이 검색자(Searcher), 스케줄러(Scheduler), 자원 관리자(Resource Manager)로 구성되어 있다.

3.1.1 검색 전략

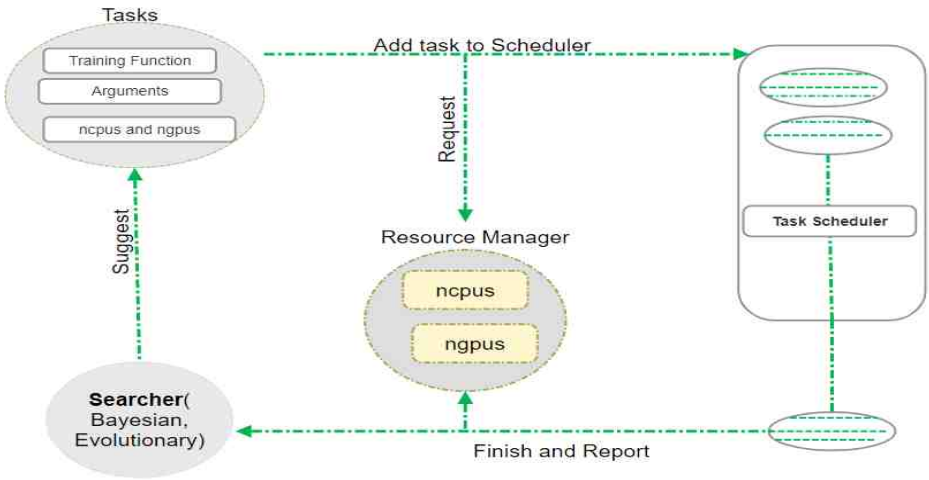
- 하이퍼 파라미터 최적화(Hyperparameter Optimization; HPO): 검색자는 다음 학습에 대한 하이퍼 파라미터값들을 제안해 준다. AutoGluon은 아키텍처를 검색할 때, 진화(강화학습) 와 베이시안 검색을 사용한다. 배포된 검색기로는 무작위 검색, SKopt 검색, RL 검색이 있다.
- 신경 아키텍처 검색(Neural Architecture Search; NAS): 신경망 아키텍처를 최적화하기 위해 AutoGluon은 프록시레스나스(ProxylessNAS) 방법을 사용한다.

3.1.2 스케줄 전략

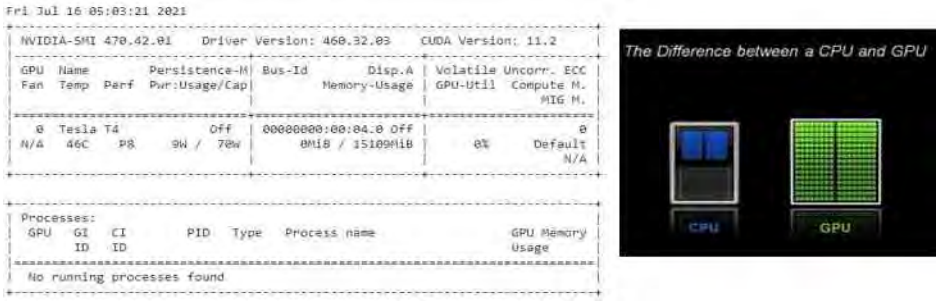
스케줄러는 모델 훈련에 필요한 계산할 자원이 충분할 때에만 훈련을 개시하는 역할을 한다. AutoGluon은 FIFO 스케줄러(FIFOScheduler), 하이퍼밴드 스케줄러 (HyperbandScheduler), RL 스케줄러(RLScheduler)를 가지고 있다. FIFO 스케줄러나 심플 스케줄러의 경우, 프로그램상의 순서대로 최적화를 시도하고, 하이퍼밴드 스케줄러(HyperbandScheduler)에는 비동기식 하이퍼밴드의 다양한 변형들이 구현되어 있다.

3.1.3 자원관리자

모든 구성 요소들은 모듈화되고 확장하기 쉽다. GPU나 CPU와 같은 자원관리자들은 학습 프로세스를 위한 구동 환경을 제공하고, (그림 2)와 같이 관리자가 볼 수 있도록 보고서를 제공한다. AutoGluon은 구조적으로 GPU 가속기가 필요하며, CPU는 한 번에 몇 개의 소프트웨어 스레드를 처리할 수 있는 많은 캐시 메모리가 있는 몇 개의 코어로 구성된다. 대조적으로 GPU는 동시에 수천 개의 스레드를 처리할 수 있는 수백 개의 코어로 구성되어 있다. GPU는 CPU보다 머신러닝 작업



(그림 1) AutoGluon 시스템의 구현 로직



(그림 2) AutoGluon 시스템의 자원관리자 화면

처리 과정에서 20배 빠른 처리 속도를 보였고, 딥러닝 분야에서 혁신을 일으켰다.

3.2 AutoGluon 모델 디자인

AutoGluon은 딥러닝 모델 디자인의 기술적인 부분들 즉, 앙상블 학습 협력, 다층 스태킹(Multi-layer Stacking), K-폴드 앙상블 배깅에서 다른 AutoML 패키지들보다 뛰어나다.

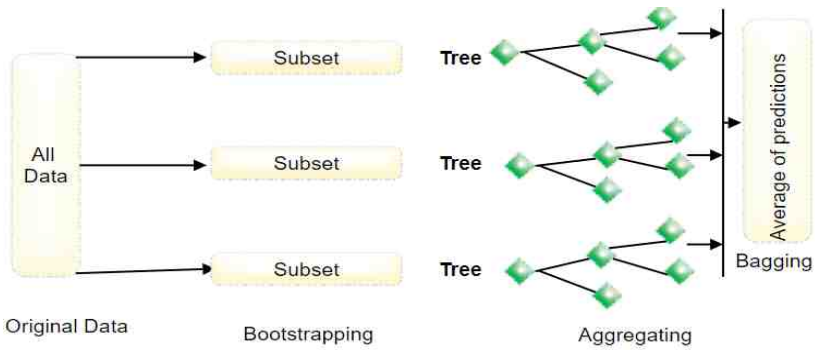
3.2.1 앙상블 학습

앙상블 학습법은 여러 개의 머신러닝 알고리즘을 결합해 더 나은 결과를 얻는 방법이다. 앙상블링은 모델들의 예측값들을 결합하고 일반화를 개선함으로써 모델들의 정확도를 개선하는 증명된 접근 방식이며 예시로는 결정트리의 앙상블인 랜덤 포레스트(Random Forest; (그림 3))가 있다.

3.2.2 다층 스태킹

숙련된 머신러닝 실무자들은 랜덤 포레스트, 캣부스트(CatBoost), K-최근접 이웃과 같은 여러 가지 방법들의 결과물을 결합하여 모델의 정확도를 높인다. 최근의 머신러닝 경진대회 커뮤니티에서는 모든 성공적인 방법들은 모델에 앙상블 기법을 사용했으며, 한 가지 모델로 1등을 한 사례를 찾아보기 힘들다.

스태킹은 (그림 4)와 같이 기본(base) 회귀 및



(그림 3) 랜덤 포레스트: 배깅 기술을 사용하여, 데이터 셋과 특징으로부터 랜덤 부트스트랩 샘플을 생성하여 결정트리를 만듦

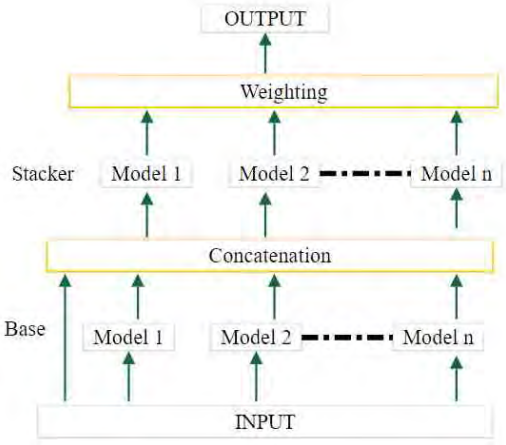
분류 모델에서 집계된 예측 값들을 토대로 스태커(stacker) 모델의 메타-분류기 혹은 stacker 회귀 모델의 학습에 특징으로써 사용하는 기법이다.

다층스태킹은 스태커 모델에서 출력된 예측이 추가적인 상위 계층의 스태커 모델의 입력으로 제공되는 것이다. 다층스태킹 앙상블은 사용하기 어렵지만 강력하고 견고하게(robust) 구현되며 현재 AutoGluon을 제외한 다른 AutoML 프레임워크에서는 사용되지 않고 있다.

AutoGluon은 어떤 전문 지식 없이 K-폴드 배깅을 포함한 다층 스택 앙상블링의 기본적인 형태를 자동으로 앙상블하고 훈련한다.

(그림 5)에서 보여주고 있는 AutoGluon의 다층 스택 앙상블링은 다음의 작업들을 수행한다.

- 베이스: 첫 번째 층은 k-폴드 앙상블을 사용하여 각자가 훈련하고 배깅되는 여러 개의 베이스 모델들을 가진다.
- 연결: 다음 층의 훈련 입력 값으로 사용되기 위해 베이스층 모델 예측과 입력 특징들이 연결된다.
- 스태킹: 다중 스태커 모델들은 연결층 출력에 대해 학습한다. 기존의 스태킹 전략과는 다르게, AutoGluon은 베이스층 모델과 같은 유형을 스태커로써 다시 사용한다. (하이퍼파라미터값도 재사용함)

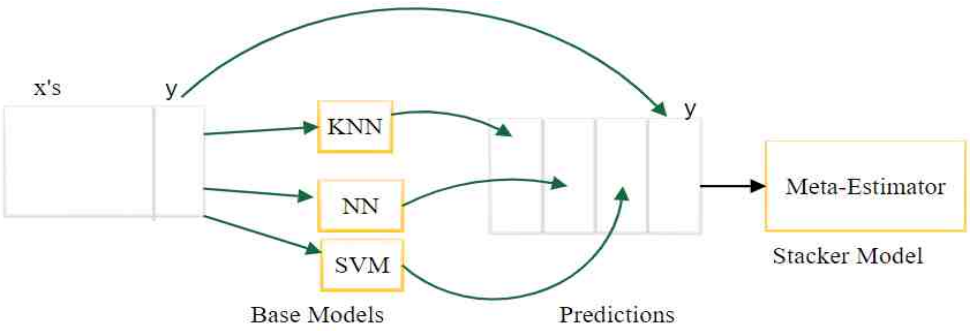


(그림 5) AutoGluon의 다층 스택 앙상블링

- 가중치 적용: 최종 스태킹층은 앙상블 선택을 통해 스태커 모델들의 예측들을 가중된 방식으로 집계한다. 대용량 모델 스택에서 예측을 집계하면 오버피팅(overfitting)에 대한 회복력이 향상된다.

3.2.3 K-폴드 앙상블 배깅

AutoGluon은 학습에 필요한 모든 활용 가능한 데이터들을 통해 스택 성능을 향상시키고, 모든 스택의 모든 모델의 모든 층에 대해 K-폴드 앙상블 배깅을 하여 검증한다. K-폴드 앙상블 배깅은



(그림 4) 스태킹 기법

K-폴드 교차검증과 유사하게, 학습 데이터 셋을 최대한 사용하면서 일반적으로 최적의 모델 파라미터를 결정하기 위한 하이퍼 파라미터 튜닝을 하기 위해 사용된다. 배깅(Bagging)은 부트스트랩 집계(Bootstrap Aggregation)는 같은 알고리즘을 사용하여 모델을 구축하면서도 각각의 학습자가 서로 다른 데이터 셋을 학습하도록 한다.

4. 결 론

본고에서는 이 중 주된 기술들인 HPO와 NAS에 대해 알아보고, 아마존에서 제안한 AutoGluon의 사례를 보였다. AutoGluon 아키텍처는 검색 전략 기법들, 스케줄 알고리즘, 모듈화 컴포넌트들로 구성되어 있고, 그것들이 앙상블 학습 다층 스태킹과 k-폴드 배깅과 같이 차별화된 모델 디자인을 통해 향상된다는 것을 소개하였다. AutoML은 머신러닝 연구자들과 관리자들을 시간이 오래 걸리고 반복적인 작업들로부터 해방시켜주며, 향후 GPU 및 CPU의 연산속도 향상과, 메타러닝과 같은 메타모델 연구의 발전 방향을 견주어보면 장래성과 파급효과가 높은 분야일 것으로 예상된다.

참 고 문 헌

[1] Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A survey of the state-of-the-art. Knowledge-Based Systems, Volume 212, 2021.

[2] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-weka: Automated selection and hyper-parameter optimization of classification algorithms. CoRR, abs/1208.3719, 2012. URL http://arxiv.org/abs/1208.3719.

[3] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian Witten. The weka data mining software: An update. SIGKDD Explor. Newsl., 11:10–18, 11 2008.

[4] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

[5] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. CoRR, abs/1603.06212, 2016. URL http://arxiv.org/abs/1603.06212.

[6] Jonathan Krauß, Bruno Machado Pacheco, Hanno Maximilian Zang, and Robert Heinrich Schmitt. Automated machine learning for predictive quality in production. Procedia CIRP, 93:443–448, 2020. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2020.04.039. URL https://www.sciencedirect.com/science/article/pii/S2212827120306016. 53rd CIRP Conference on Manufacturing Systems 2020.

[7] Haifeng Jin, Qingquan Song, and Xia Hu. Efficient neural architecture search with network morphism. CoRR, abs/1806.10282, 2018. URL http://arxiv.org/abs/1806.10282.

[8] Erin LeDell and Sebastien Poirier. H2O

AutoML: Scalable automatic machine learning. 7th ICML Workshop on Automated Machine Learning (AutoML), July 2020. URL https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf.

[9] H2O.ai. H2O AutoML, 2021. URL http://docs.h2o.ai/h2o/latest-stable/ h2o-docs/automl.html. H2O version 3.32. 1.2.

[10] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data, 2020.

[11] Chaoyu Guan, Ziwei Zhang, Haoyang Li, Heng Chang, Zeyang Zhang, Yijian Qin, Jiyan Jiang, Xin Wang, and Wenwu Zhu. Autogl: A library for automated graph learning. CoRR, abs/2104.04987, 2021. URL https://arxiv.org/abs/2104.04987.

저 자 약 력

니콜라우스 이삭 토마스

이메일 : d0195032@kowon.dongseo.ac.kr

- 2018년 United African University of Tanzania(UAUT) 컴퓨터공학 및 정보기술학과 (학사)
- 2021년 동서대학교 컴퓨터공학과 (석사)
- 2021년~현재 동서대학교 컴퓨터공학과 (박사 과정)
- 관심분야 : Hyperparameter Optimization and Network Architecture Search, Multi-Agent Reinforcement Learning

마타요, 피터 베아투스

이메일 : be120.pt@gmail.com

- 2019년United African University of Tanzania(UaUT) 컴퓨터공학 및 정보기술학과
- 2021년~현재 동서대학교 컴퓨터공학과(석사 과정)
- 관심분야 : Auto Reinforcement Learning(AutoRL), Neural Network Search.

신 지 용

이메일 : tkwjsup@kowon.dongseo.ac.kr

- 2018년 김천대학교 방사선학과 (학사)
- 2021년~현재 동서대학교 컴퓨터공학과 (석사 과정)
- 관심분야 : Hyperparameter Optimization and Network Architecture Search

강 대 기

이메일 : dkkang@dongseo.ac.kr

- 1992년 한양대학교 컴퓨터공학과 (학사)
- 1994년 서강대학교 컴퓨터공학과 (석사)
- 2006년 아이오와주립대학교 컴퓨터공학과(박사)
- 1994년~1999년 전자통신연구원 연구원
- 2006년~2006년 전자통신연구원 부설연구소 선임연구원
- 2006년-현재 동서대학교 컴퓨터공학과 교수
- 관심분야 : Hyperparameter Optimization and Network Architecture Search, Adversarial Machine Learning, Multi-Agent Reinforcement Learning