가중치 공유를 이용한 소프트웨어 카테고리 자동 분류

김민하, 심규진, 이민수,
왕승재, 권준혁, 이찬근
*중앙대학교 소프트웨어학부
minas.rtse@gmail.com, kyshim28@gmail.com, als950901@naver.com,
sengjeawang4065@gmail.com, cccang8@naver.com, cglee@cau.ac.kr

Automated Classification of Software Category using Weight Sharing

Min-Ha Kim, Kyoo-Jin Shim, Min-Soo Lee,
Sheng-Tsai Wang, Jun-Hyeok Kwon, Chan-Gun Lee
Dept. of Computer Science and Engineering, Chung-Ang University

요 약

현재까지 심층 학습을 이용하여 텍스트를 자동으로 분류해주는 연구가 활발히 진행되었으며, 특히 소프트웨어 카테고리를 자동으로 분류해주는 연구가 이루어지고 있다. 최근 심층 신경망의 적절한 구조를 효율적으로 탐색할 수 있는 가중치 공유 기법이 연구되었다. 우리는 이를 응용하여 본 논문에서 가중치 공유를 이용한 소프트웨어 카테고리 분류 방법을 제안하며, 여러 실험을 통해 해당 기법의 성능을 측정하고 논의한다.

1. 서론

오픈 소스 소프트웨어가 세계를 혁명적으로 변화시키면서, 전 세계의 개발자들은 오픈 소스 공개 저장소를 탐색하고 모범 사례를 학습하며, 수 백만명의 사용자가 사용하는 소프트웨어를 개선하기 위해 협력할 수 있다[1]. 이러한 소프트웨어 공개저장소에는 엄청난 양의 소프트웨어 개발 데이터가 포함되어 있으며[2], 이에 따라 공개저장소에는 소프트웨어 시스템의 전반적인 개발에 대한 역사적, 가치 있는 정보를 포함한다[3].

시간이 지남에 따라 다양한 소프트웨어 공개저장소가 새롭게 등장하며 그들의 규모도 지속적으로 증대되고 있으나[4], 대규모의 저장소 탐색을 위한 제반 기능이 만족스러운 수준으로 제공되지 못하여 코드 탐색과 같은 작업에서 사용자들의 노력에 대한 비용이 매우 크다[5].

이러한 이유로 사용자들의 코드 탐색 작업을 도울 수 있는 소프트웨어 카테고리 분류 자동화 기법이 제안되었고 활발하게 연구되고 있다. 아직까지 소프트웨어 카테고리 분류 자동화를 위한 기존 연구에서 심층학습 기법을 이용하여 카테고리를 자동으로 분류하는 방법을 적용한 사례가 많지 않았다. 최근 심층 신경망 학습 기법을 이용하여 분류 실험을 진행한 연구로, Kim 등은 계층 분류 기법을 이용하여 소프트웨어 카테고리 자동 분류를 수행하였으며, 이를 통해 모델의 타당성을 입증한 바 있다[6].

본 논문에서 우리는 소프트웨어 카테고리 자동 분류기의 성능 향상을 위한 새로운 기법을 제안한다. 해당 기법의 주된 아이디어는 상위 카테고리 학습의 최종 가중치 뿐만 아니라 중간 계층에서의 가중치를 함께 이용해서 하위 카테고리의 분류에 적용하는 것이며, 심층 신경망의 구체적인 설계와 구현 방안으로 상위 카테고리에서의 가중치를 하위 카테고리의 가중치에 공유하는 방식으로 개발한다.

제안된 방법의 효과를 평가하기 위해 기존에 발표된 소프트웨어 카테고리 자동 분류 방법을 베이스라인으로 하였으며, 두 방법의 특징과 실험을 통해 성능을 비교한다.

2. 관련 연구 및 기법

본 절에서는 본 논문에서 제안하는 모델과 관련된 연구와 기법을 소개한다.

2-1. 가중치 공유(Weight Sharing)

신경망에서의 가중치 공유는 매개변수의 수를 줄여 모델을 압축시키는 하나의 형태이다. 주로 합성 곱 신경망 모델에서 쓰이는 기법으로, 매개변수 공유라고도 불리며, 그림 1과 같은 방식이다.

Hieu Pham 등은 매개변수 공유 기법을 처음으로 소개하였으며[7], 상위 계산 그래프와 하위 그래프를 구축한 후, 모든 아키텍처에 해당 매개변수를 공유하는 방식이며. 이를 통해 더 나은 소프트웨어 아키텍처를 탐색한다.

이를 바탕으로 본 논문에서는 신경망 모델에서 매개변수에 해당하는 가중치를 합성 곱 신경망끼리 공유시킴으로써 분류 실험을 진행한다.

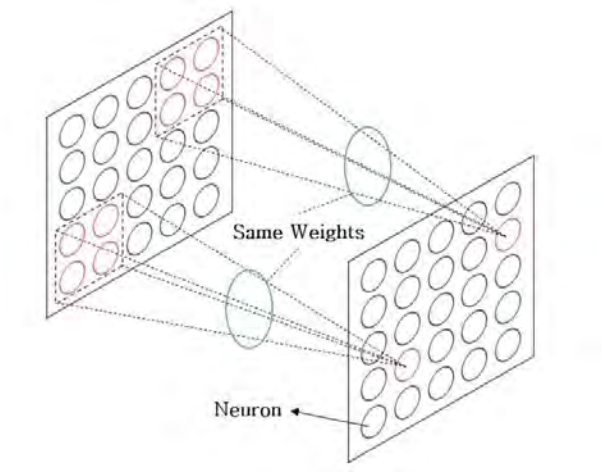

그림 1 뉴런에서의 가중치 공유

2-2 계층 분류 기법

기존 연구에서 진행했던 계층 분류 기법은 API와 AST를 입력 데이터로 사용하여, 상위 모델에서 상위 카테고리에 대한 학습이 이루어진 후 출력 데이터를 바탕으로 하위 모델에서 하위 카테고리에 대한 학습이 추가적으로 진행되어진다. 기존 연구에서 사용한 모델의 구성은 그림 2와 같다.

3. 실험

본 절에서는 본 논문에서 진행하는 실험에 사용되어지는 가중치 공유 모델을 소개한다.

3.1 실험 방법

본 논문에서 구성한 모델에서는 기존 연구에서 사용하였던 계층 분류 모델에 가중치를 공유하는 기법을 응용한 모델로, 사용한 모델의 구성도는 그림 3과 같다.

기존에 사용 했던 모델에서는 계층 분류 방식으로 연속적으로 학습을 진행하였으나, 하위 카테고리로 학습이 넘어갈 때에, 전체적으로 상위 카테고리에서의 영향력을 받아야 한다는 판단 하에 상위 모델과 하위 모델들 각각 첫 번째 합성 곱 신경망 계층들 끼리 가중치를 공유하도록 진행하였다.

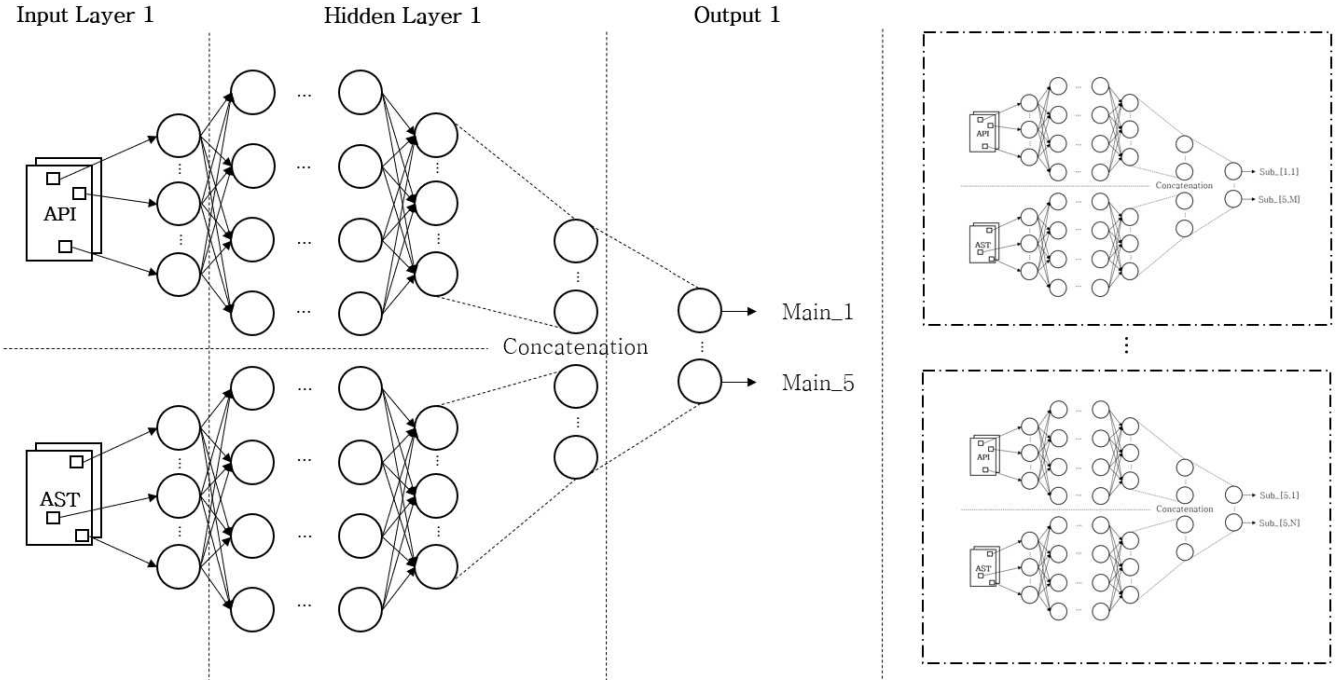

그림 2 기존 연구[6]에서 사용한 계층 분류 모델 구성도

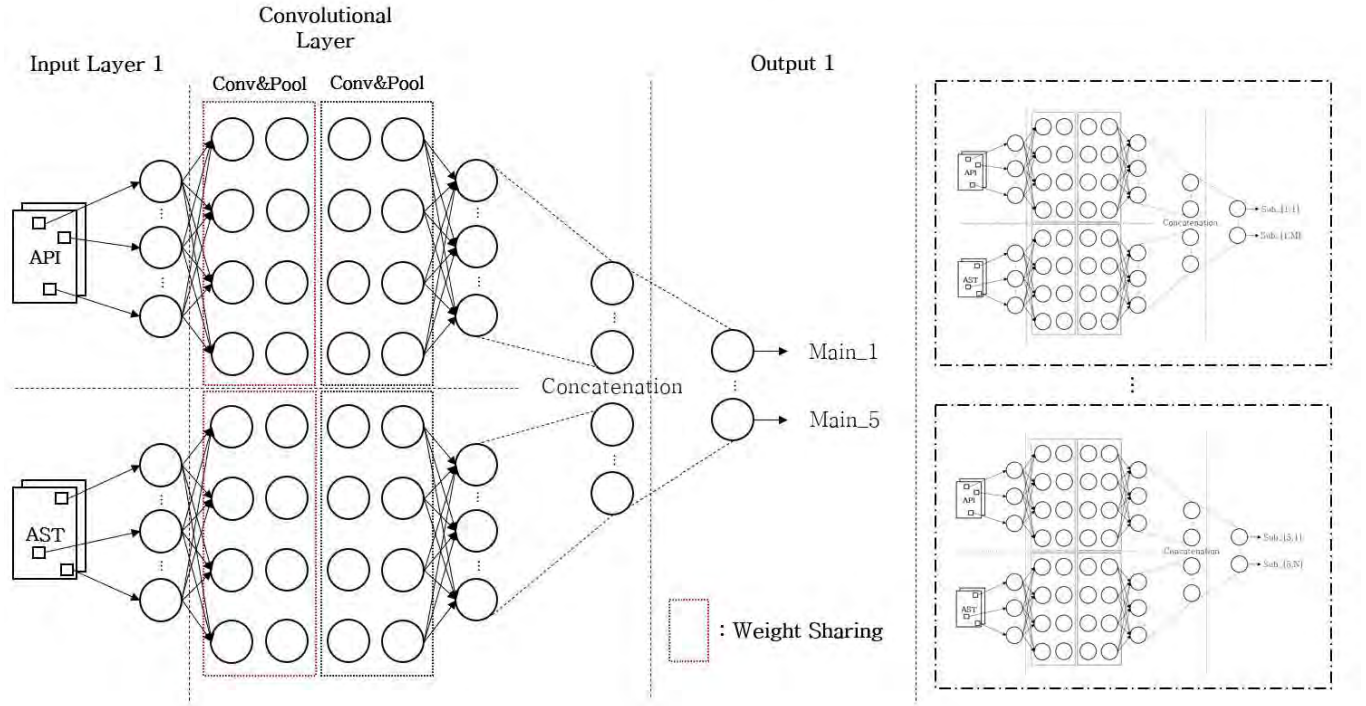

그림 3 본 논문에서 사용한 가중치 공유 모델 구성도

사용한 데이터는 기존 연구에서 사용하였던 SourceForge.net에서 수집한 데이터를 기반으로 임베딩(Embedding)을 진행하였고, 성능 평가를 위한 기준은 비교를 위해 기존 연구에서의 평가 기준인 정확도(Accuracy), Soft F1 Score, 자카드 유사도(Jaccard Similarity), 정밀도(Precision) 그리고 재현율(Recall)을 사용하여 실험을 진행하였다.

4. 실험 결과

본 절에서는 본 논문에서 진행하는 모델에 대한 실험 결과와 계층 분류 모델의 성능을 비교 분석한다.

기존 연구와의 성능 비교를 위해 기존 연구에서 사용한 모델은 "Hierarchical CNN(H-CNN)", 본 논문에서 사용한 모델은 "Weight Sharing CNN(WS-CNN)"이라고 칭하며, 실험 결과는 표 1과 같다.

표 1의 결과를 보았을 때, 정확도는 이전 연구에 비해 성능이 3.6%p 감소하였으나, 자카드 유사도는 13.1%, Soft-F1 Score는 14.9%p, 정밀도는 29.8%p 그리고 재현율은 20.7%p 증가한 것을 알 수 있다.

다른 평가지표에 비해 정확도만 성능이 소폭 감소된 것을 확인할 수 있는데, 이 전 연구에 비해 모델의 복잡도가 상대적으로 단순해지면서 약간의 과소적합(underfitting)이 일어난 것으로 판단되어진다.

그러나 정확도보다는 다른 평가지표들이 다중 레이블 모델을 평가하기에 더 적합한 평가지표들이며, 이러한 결과를 통해 단순 계층 분류 모델보다는 가중치 공유 기법을 더한 계층 분류 모델이 카테고리 분류에 대해 더 적합한 모델임을 확인하였다.

표 1 이전 연구의 모델과의 성능 비교 결과

평가기준	H-CNN	WS-CNN
Accuracy	97.7	94.1
Jaccard Similarity	48.6	61.7
Soft-F1 Score	51.1	66.0
Precision	49.5	79.3
Recall	48.6	69.3

단위 : %p

5. 결론 및 향후 연구

본 논문에서는 신경망을 이용한 카테고리 자동 분류 연구와 성능 비교의 맥락에서 가중치 공유를 이용하여 분류하는 방법을 제안하였으며, 이전 연구와 비교하여 비약적인 성능 향상을 이루었다. 하지만, 본 논문에서 사용한 데이터의 수는 실제 세계에서 적용하기에는 부족한 수량이다. 그리고 부족한 데이터임에도 불구하고 과다한 메모리의 사용을 발견하였다.

향후 우리는 다른 소프트웨어 공개저장소에서 데이터를 수집하고, 카테고리를 추가함으로써 타당성을 입증하고자 하며, 임베딩 기법을 새로 정립한 후 메모리 과다 사용을 방지하고자 한다. 이를 통해 사용자들이 카테고리로 소프트웨어를 탐색함에 있어 쉽게 진행할 수 있도록 고도화시키고자 한다.

감사의말

본 연구는 한국연구재단 기초연구사업(과제 번호 NRF-2017R1E1A1A01075803)과 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업 (2017000100041001)의 연구결과로 수행되었음

참고문헌

[1] Yury Zhauniarovich, Yazan Bosh,maf, Husam Al Jawaheri, Mashael Al Sabah, "Characterizing Bitcoin donations to open source software on GitHub," arXiv:1907.04002, 2019

[2] Ahmad Abdellatif, Khaled Badran, Emad Shihab, "MSRBot: Using Bots to Answer Questions from Software Repositories," An International Journal of Empirical Software Engineering 25: 1834-1863, 2020

[3] David Spadini, Mauricio Aniche, Alberto Bacchelli, "PyDriller: Python framework for mining software repositories," Proceeding of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering November 4-9: 908-911, 2018

[4] Gabriele Bavota, "Mining Unstructured Data in Software Repositories: Current and Future Trends," 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER) 14-8: 1-12, 2016

[5] Gang Hu, Min Peng, Yihan Zhang, Qianqian Xie, Wang Gao, Mengting Yuan 2020, "Unsupervised software repositories mining and its application to code search," Software: Practice and Experience, vol 50, no. 3, pp. 299-322

[6] Hun Sung, Min-Ha Kim, Hyun Sung, Seung-Jae Wang, Chan-Gun Lee, "Software Category Automated Classification Using Hierarchical Classification Technique," Proceeding of Korea Multimedia Society 23-1: 45-48, 2020

[7] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, Jeff Dean, "Efficient Neural Architecture Search via Parameter Sharing," Proceedings of the 35th International Conference on Machine Learning, PMLR 80: 4095-4104, 2018