기능성에 따른 프로그래밍 소스코드 분류를 위한 Deep Learning Model 연구

윤주성*, 이은헌*, 안진현*, 김현철*
*고려대학교 컴퓨터학과
e-mail:eagle705@korea.ac.kr

A Study on Deep Learning model for classifying programs by functionalities

Joo-Sung Yoon*, Eun-Hun Lee*, Jin-Hyeon An*, Hyun-Cheol Kim*
*Dept of Computer Science and Engineering, Korea University

요 약

최근 4차 산업으로 패러다임이 변화함에 따라 SW산업이 더욱 중요하게 되었다. 이에 따라 전 세계적으로 코딩 교육에 대한 수요도 증가하게 되었고 기업에서도 SW를 잘 만들기 위한 코드 관리 중요성도 증가하게 되었다. 많은 양의 프로그래밍 소스코드를 사람이 일일이 채점하고 관리하는 것은 사실상 불가능하기 때문에 이러한 문제를 해결할 수 있는 코드 평가 시스템이 요구되고 있다. 하지만 어떤 코드가 좋은 코드인지 코드를 어떻게 평가해야하는지에 대한 명확한 기준은 없으며 이에 대한 연구도 부족한 상황이다. 최근에 주목 받고 있는 Deep Learning 기술은 이미지 처리, 자연어 처리등 기존의 Machine Learning 알고리즘이 냈던 성과보다 훨씬 뛰어난 성과를 내고 있다. 하지만 Programming language 영역에서는 아직 깊이 연구된 바가 없다. 따라서 본 연구에서는 Deep Learning 기술로 알려진 Convolutional Neural Network의 변형된 형태엔 Tree-based Convolutional Neural Network를 사용하여 프로그래밍 소스코드를 분석, 분류하는 알고리즘 및 코드의 Representation Learning에 대한 연구를 진행함으로써 이러한 문제를 해결하고자 한다.

1. 서론

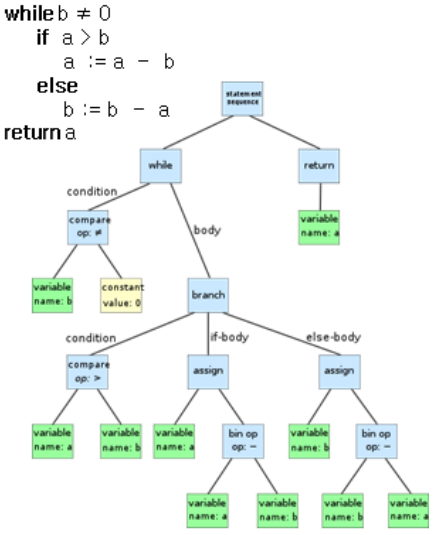
최근 4차 산업으로 패러다임이 변화함에 따라 SW가 더욱 중요하게 되었다. 이에 따라 SW교육에 대한 관심도 증가하게 되었고 프로그래밍 소스코드를 어떻게 평가할 것인가 에 대한 문제가 제기되고 있다. 많은 양의 프로그래밍 소스코드를 사람이 일일이 채점하는 것은 사실상 불가능하기 때문에 이러한 문제를 해결할 수 있는 코드 평가 시스템이 요구되고 있다. 최근에 주목 받고 있는 Deep Learning 기술은 이미지 처리, 자연어 처리 등의 분야에서 뛰어난 성과를 내고 있지만 아직 Programming language 영역에서는 깊이 연구된바가 없다. 따라서 본 연구에서는 Deep Learning 기술 중 하나인 Convolutional Neural Network의 변형된 형태인 Tree-based Convolutional Neural Network를 사용하여 프로그래밍 소스코드를 분석, 분류하고 이러한 프로그래밍 소스코드를 Embedding하는 Representation Learning에 대해 연구하려 한다.

2. 관련 연구

프로그래밍 소스코드를 Neural Network의 Input data로 사용하기 위한 방법 및 Tree-based Convolutional Neural Network(TBCNN)에 대해서 설명하고 관련연구에 대해 소개하고자 한다.

2-1. Abstract Syntax Tree(AST)

프로그래밍 언어는 자연어와 달리 구조적이고 문법적인 정보가 매우 중요하다. 프로그래밍 소스코드에 이러한 구조적인 정보를 더 잘 반영 할 수 있는 Representation을 사용한다면 코드의 의미를 더 잘 알아낼 수 있을 것이다. Abstract Syntax Tree란 프로그래밍 언어를 문법적 구조를 따라 Tree 형태로 변환한 형태다.

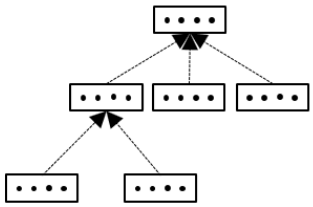


(그림 1) Abstract Syntax Tree

2-2. Representation Learning for AST Nodes

AST node에 대해서 Representation Learning을 통해 discrete symbol을 real-value vector로 변환할 수 있다. 기존적인 가정은 Parent node와 Child node간의 구조적인 정보를 고려하는 것이다. Parent node의 vector와 Child node로 표현된 vector의 Euclidean distance가 최소가 되도록 목적함수를 정하고 Negative sampling을 통해 학습시킨다.

$$vec(parent) \approx \tanh(\sum_i W_{child,i} \cdot vec(child_i) + b_{child})$$



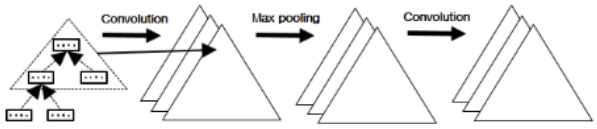
(그림 3) Representation Learning for AST Nodes

3. Multilayer TBCNN 기반의 분류기법

본 연구에서는 기존에 연구되었던 TBCNN의 단점을 개선하기 위한 방법을 제안한다.

3-1. Multilayer TBCNN

기존의 연구에서는 Convolutional layer 한 개의 Depth는 깊었지만 Layer자체가 많지는 않았다. CNN은 Layer가 많아질수록 상위 Layer에서 High feature를 얻어낼 수 있으므로 Convolutioanl layer를 추가하고 중간에 Max pooling을 통해 중요한 feature들을 더 잘 얻어내도록 Model의 Architecture를 바꾼다.

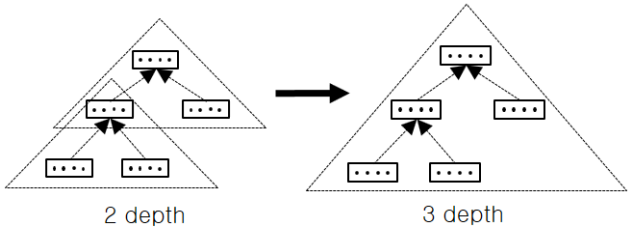


(그림 4) Multilayer TBCNN

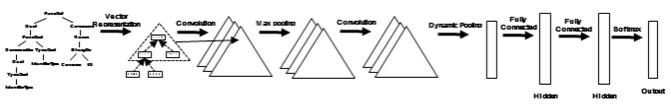
Layer의 개수를 늘리는 대신에 발생할 수 있는 Vanishing gradient 문제를 해결하기 위해 Rectified Linear Unit(ReLU)를 Activation function으로 사용한다.

3-2. Improved Tree-based Receptive Field

기존의 연구에서 TBCNN의 Convolutional Layer를 만들 때 Receptive field는 Depth를 2로 정해서 사용했다. 이러한 경우 임의의 노드에서 바로 인접한 노드와의 구조적정보만을 반영할 수 밖에 없게 된다. Receptive field의 Depth에 관한 hyper parameter 조건에 대해 연구하려 한다.



(그림 5) Receptive Field of TBCNN



(그림 6) Advanced Deep Learning Model Architecture

4. 결론

본 연구는 최근 Machine Learning 분야에서 활발하게 연구가 진행되고 있는 Deep Learning 기술을 이용하여 프로그래밍 소스코드를 자동으로 분류하는 알고리즘에 대해서 연구한다. 이를 위한 주된 아이디어는 두 가지로 프로그래밍 소스코드를 Neural Network의 Input data로 사용할 수 있게 하는 Representation을 AST 구조를 이용해서 학습하는 것과 CNN의 변형된 아키텍처인 TBCNN을 사용하는 것이다. 기존의 연구에서 이러한 방법은 이미 효과가 있다고 보고된바 있다. 본 연구에서는 기존연구의 단점을 분석하고 극복하기 위한 방법을 제안하였다. 프로그래밍 소스코드를 더욱 잘 분석하기 위해서는 Representation Learning을 통해 프로그래밍 소스코드를 분류하기에 적합한 형태로 변형하는 것과 Tree형태의 데이터를 분류하는 아키텍처를 연구하려는 노력이 앞으로도 계속 필요할 것으로 보인다.

참고문헌

[1] Mou, L; Peng, H.; Li, G.; Xu, Y.; Zhang, L.; and Jin, Z. “Discriminating neural sentence modeling by tree-based convolution,” In EMNLP, 2315-2325, 2015.

[2] Peng, H.; Mou, L; Li, G.; Liu, Y.; Zhang, L.; and Jin, Z. “Building program vector representations for deep learning,” In Proc. 8th Int. Conf. Knowledge Science, Engineering and Management, 547-553, 2014.

[3] Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin. “Convolutional Neural Networks over Tree Structures for Programming Language Processing.”The 30th AAAI Conference on Artificial Intelligence (AAAI), 2016

감사의 글

“본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음”

(R71151610080001002)