

인공지능 교육을 위한 멀티 플랫폼 오목 프로그램 설계

차주형 · 우영운*

동의대학교

Design of a Multi-Platform Omok Program for Artificial Intelligence Education

Joo Hyoung Cha · Young Woon Woo*

Dong-eui University

E-mail : aoikazto@naver.com / ywwoo@deu.ac.kr

요 약

본 논문은 프로그래밍의 기초 교육을 이수한 개발자가 빅데이터와 인공지능을 학습하기 위해, C/C++ 언어로 프로그래밍을 할 수 있는 인공지능 교육서비스에 대해 다룬다. 또한 개발 환경에 따른 맞춤형 개발 환경 구성 시스템과 사용자가 인공지능 구현하여 테스트하는 방법에 대해 설명한다. 이 외에도 다양한 내부 파라미터 조작을 통해 인공지능에 미치는 영향을 확인할 수 있는 기능을 갖추고 있다. 향후 네트워크 통하여 언어의 제약이 없는 인공지능 교육 서비스 개발이 가능할 것으로 예상된다.

ABSTRACT

This paper deals with AI education service that enables developers who have completed basic programming education to program in C/C++ language in order to learn big data and artificial intelligence. In addition, a customized development environment configuration system according to the development environment and how the user implements and tests artificial intelligence are explained. And also it has a function to check the effect on artificial intelligence through manipulation of various internal parameters. It is expected that it will be possible to develop artificial intelligence education services without language restrictions through networks in the future.

키워드

Artificial intelligence, Omok, AI education, Game tree

1. 서 론

최근 인공지능 산업의 발전과 함께 데이터 사이언스 교육 프로그램이 다수 개발되고 있다. 따라서 빅데이터를 다루는 방법과 프로그래밍 교육에 대한 필요성이 대두되고 있다. 하지만 대부분의 인공지능이나 데이터 사이언스 교육은 저학력이거나 해당 분야에 문외한 사람을 위한 교육이 많다.[1] 또한 프로그래밍 교육기관에서 인공지능 교육을 수강할 때 딥 러닝, 유전 알고리즘, 퍼지 이론과 같은 인공지능 기법을 먼저 접하여 어려움을 겪고 포기하게 되는 경우가 많다.

인공지능 교육은 대부분 웹, 블록 코딩, Python

을 활용하여 수치적인 값과 그래프 형식으로 값의 변화를 나타내 보여준다. 하지만 대부분의 고등, 대학 과정에서는 블록 코딩, 웹이 아닌 C/C++로 학습한다. 따라서 고등, 대학생들 대상으로 인공지능 학습에 있어 C/C++과 같은 친숙한 언어로 교육하는 것이 효율적이다.

C/C++ 언어는 다른 언어에 비해서 운영체제나 컴파일러, 통합 개발 환경에 영향을 많이 받는다. 그래서 사용자의 컴퓨터 별로 개발 환경을 구현해야 하는 문제가 있다. 해당 문제를 해결하기 위하여 빌드 시스템과 패키지 매니저를 활용 및 분석하여 환경과 상관없이 C/C++[2] 언어로 인공지능 교육 서비스를 제공하는 방법을 제안한다.

* corresponding author

II. 환경 구현

C++언어는 Python, Java, .Net과 같은 인터프리터 언어가 아닌 컴파일 언어이기 때문에 여러 운영체제에 따라 빌드와 실행이 가능한 환경[3]을 만들어 주어야 한다. 그러므로 다양한 환경과 상황에 따라 올바른 개발 환경을 제공할 수 있어야 한다. 멀티 플랫폼을 지원하기 위해 빌드 시스템인 CMake[4]와 패키지 매니저인 Vcpkg[5]를 활용하여 외부 종속성 문제를 해결하였다.

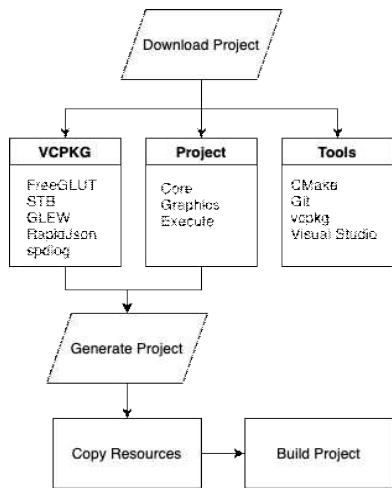


그림 1. 사용자 정의 인공지능 모듈 개발 환경

사용자에게 시각적으로 화면을 표현하기 위해 OpenGL와 FreeGLUT(OpenGL Utility ToolKit)를 활용하여 GUI(Graphic User Interface)를 제공하였다. 사용자 친숙함을 위해 이미지 렌더링을 위해 이미지 프로세싱 라이브러리 stb와 glew를 사용하였다.

위 그림1은 프로젝트가 생성기를 통하여 어떻게 개발 환경이 구축이 자동화가 되는지, 외부 라이브러리 종속성을 해결하는지 나타낸다. 프로젝트 생성기를 개발할 때 사용자의 커널의 종류가 중요하다. 커널은 대표적으로 NT 와 Unix가 있다. 운영체제의 점유율[6]이 NT 계열 75%, Unix 계열이 25%이다. 많은 사용자가 NT 계열을 사용하고 있으며 Visual Studio를 사용한다. NT계열은 배치 파일로 프로젝트 다운로드를 통해 환경 설정을 자동화하였으며, Unix계열은 쉘 명령어로 자동화하였다.

프로젝트 생성기를 통해서 CMake, Git, vcpkg를 다운로드를 받아 프로젝트를 빌드하기 위한 리소스를 다운을 받고 오목 인공지능 코드와 코드 종속성을 해결하기 위해 외부 라이브러리를 설치한다. CMake를 통하여 솔루션 및 프로젝트를 생성한다. NT 계열이라면 Visual Studio 솔루션이 결과로 나와 기존과 동일한 개발이 가능하며, Unix 계열은 CMake와 GCC를 활용하여 빌드가 가능하다.



그림 2. macOS에서 실행한 결과물

프로젝트를 제대로 실행하면 그림2 로 정상적으로 동작이 가능하다.

III. 인공지능 모듈 개발 및 환경 설정

인공지능을 구현하기 위해 모델을 저장하고 불러오기 위해서 JSON[7][8] 형태로 저장한다. 데이터 내부 구조는 현재 수의 점수와 다음 수에 대한 점수가 트리 형식으로 정의되어 있다. 점수를 계산하는 방식은 Discount Factor 방식으로, 현재의 가치는 미래의 승리, 패배 한 수의 보상치의 합으로 정의했다. 즉 승리한 수에 가까울수록 미래의 가치가 높을 것이며, 패배한 수가 많을수록 미래의 가치는 낮추는 방식으로 점수를 계산하였다.

인공지능은 총 3개가 있으며 각 인공지능 별로 추론한 수가 없으면 다른 인공지능을 택하는 방식으로 되어 있다. 첫 번째 인공지능은 기보 기반으로 이전에 둔 수와 이긴 게임의 수를 계산하여 Discount Factor로 기보 점수를 계산해 가장 높은 수를 두는 방식이다.

```

vector<scorePoint> userAi(color term, vector<vector<color>> board)
{
    return vector<scorePoint>();
}

color winCheck(vector<vector<color>> board)
{
    return color::none;
}
  
```

그림 3. 사용자 정의 인공지능 모듈 개발 환경

두 번째 인공지능은 사용자의 인공지능이며, 교육 및 학습 목적으로 개발되었다. 또한 사용자가 인공지능을 작성할 수 있는 공간이며 가장 많은 변화가 일어날 수 있다. 현재 보드 상태와 다음 턴이 누군지에 대한 정보를 입력받아 승리 판정과, 사용자 AI를 만들어서 적용할 수 있다. 세 번째 인공지능은 평가 함수와 통해 알파-베타 가지치기 기법[9]을 이용해 가장 높은 점수를 가지는 수를 두도록 한다.

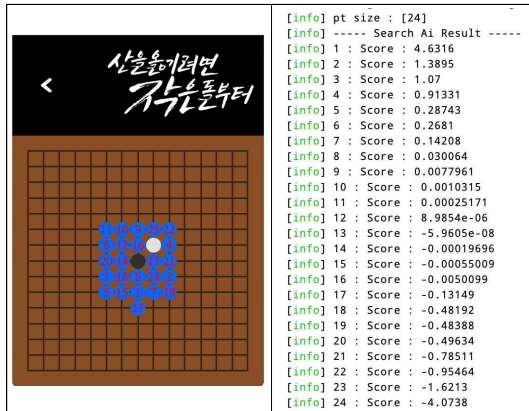


그림 4. 인공지능 학습 및 추론 결과

터미널 창에서 인공지능이 동작하는 순서와 점수를 출력하여 학습된 결과를 나타낸다. 이를 통해 사용자 인공지능에서 인공지능 검색 결과와 실행 시간을 나타내면서 인공지능 실행시간 제한, AI의 오류 검증, 테스트를 할 수 있는 모듈을 구현 하였다.

사용자 정의가 가능한 파라미터는 첫 번째 인공지능의 Discount Factor 수치, 세 번째 인공지능의 제한시간, 사용자 인공지능, 사용자 정의 승리판정 알고리즘이 있다.

IV. 활용 가능성

인공지능 교육에서 사용자 정의 인공지능과 승리판정 알고리즘을 활용하여 오목게임이 아닌, 대각선만 승리 판정만 하게 하여 대각선 오목 게임 인공지능을 만들어 보는 교육을 할 수 있다. 현재 보드 상태를 받고, 다음 턴의 데이터를 활용하여 강화 학습을 통한 교육이 가능할 것이다. 함수의 반환 값인 다음 수의 위치만 반환이 가능하다면 위의 플랫폼을 활용하여 게임 보드 인공지능 교육이 가능하다. 그리고 인공지능 수업을 할 때 운영체제의 영향을 받지 않는 OpenGL을 활용해 개발하여 코드를 직접 작성하고 인공지능을 모바일 기기에서 테스트가 가능하다. 사용자가 C/C++ 언어로 자신만의 인공지능을 만들어 모바일 기기에서 직접 테스트할 수 있을 것으로 예상된다.

V. 결론

본 논문에서 다양한 운영체제와 사용자 정의가 가능한 파라미터를 활용한 인공지능 알고리즘 작성 방안을 제안하였다. 사용자가 프로그래밍을 많이 접해보지 않았거나, 개발하는 운영체제가 달라도 서비스를 활용할 수 있는 구조로 나타냈다.

향후 시스템 추가 구현을 통해 오목 게임을 넘

어 체스, 오셀로와 같은 또 다른 보드 게임에도 적용할 것이다. 또한 코드로 작성해서 개발하는 것이 아닌 네트워크 통신을 통하여 인공지능 구현 하는 방식을 제공할 예정이다.

References

- [1] 이정미, 강의선. “비전공자를 위한 인공지능 기초 교양 교육 프로그램 개발,” 한국디지털콘텐츠학회 논문지, vol. 22, no. 9, pp. 1431-1440, 2021.
- [2] ISO/IEC 14882:2017 [Internet]. Available : <https://www.iso.org/standard/68564.html>.
- [3] M. Wojtczyk and A. Knoll, “A Cross Platform Development Workflow for C/C++ Applications,” 2008 The Third International Conference on Software Engineering Advances, 2008, pp. 224-229, doi: 10.1109/ICSEA.2008.41.
- [4] Build with CMake. Build with Confidence. [Internet]. Available : <https://cmake.org>.
- [5] Get started with vcpkg. [Internet]. Available : <https://vcpkg.io/en/getting-started.html>.
- [6] Desktop Operating System Market Share Republic Of Korea [Internet]. Available : <https://gs.statcounter.com/os-market-share/desktop/south-korea>.
- [7] ECMA-404 [Internet]. Available : <https://www.ecma-international.org/publications-and-standards/standard/s/ecma-404/>.
- [8] The JavaScript Object Notation (JSON) Data Interchange Format [Internet]. Available : <https://tools.ietf.org/html/rfc7159>.
- [9] Zhang-Congpin and Cui-Jinling, “Improved Alpha-Beta Pruning of Heuristic Search in Game-Playing Tree,” 2009 WRI World Congress on Computer Science and Information Engineering, 2009, pp. 672-674, doi: 10.1109/CSIE.2009.527.