

인공지능 학습을 위한 웹 컴파일러 설계 및 구현

Design and Implementation of Web Compiler for Learning of Artificial Intelligence

박진태 · 김현국* · 문일영
한국기술교육대학교 컴퓨터공학부

Jin-tae Park · Hyun-gook Kim* · Il-young Moon

Department of Computer Engineering, KOREATECH, Chungcheongnam-do, 31253, Korea

[요 약]

4차 산업혁명과 ICT 기술의 중요성이 증가함에 따라 소프트웨어 중심 사회가 초래되었다. 기존 소프트웨어 교육은 학습 환경 구성에 제한적이었으며, 초기에 많은 비용이 발생하였다. 이를 해결하기 위하여 웹 컴파일러를 활용하는 형태의 학습 방법이 개발되었다. 웹 컴파일러는 다양한 소프트웨어 언어를 지원하며, 컴파일 결과를 사용자에게 웹을 통해 보여준다. 하지만 4차 산업혁명의 핵심기술인 인공지능에 대한 웹 컴파일러는 아직 미비한 상황이다. 본 논문에서는 구글 인공지능 라이브러리인 텐서플로우 기반의 웹 컴파일러를 설계, 구현하였다. nodeJS 기반의 서버에 텐서플로우와 텐서플로우 서빙, 파이썬 주피터를 구현하고, meteorJS 기반의 웹 서버를 구축하여 인공지능 학습을 위한 시스템을 구현하였다. 소프트웨어 중심 사회에서 인공지능 학습을 위한 도구로서의 활용이 가능할 것으로 기대된다.

[Abstract]

As the importance of the 4th industrial revolution and ICT technology increased, it became a software centered society. Existing software training was limited to the composition of the learning environment, and a lot of costs were incurred early. In order to solve these problems, a learning method using a web compiler was developed. The web compiler supports various software languages and shows compilation results to the user via the web. However, Web compilers that support artificial intelligence technology are missing. In this paper, we designed and implemented a tensor flow based web compiler, Google's artificial intelligence library. We implemented a system for learning artificial intelligence by building a meteorJS based web server, implementing tensor flow and tensor flow serving, Python Jupyter on a nodeJS based server. It is expected that it can be utilized as a tool for learning artificial intelligence in software centered society.

Key word : Artificial Intelligence, Web-compiler, Tensor flow, Tensor flow serving, E-learning.

<https://doi.org/10.12673/jant.2017.21.6.674>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 30 November 2017; Revised 7 December 2017

Accepted (Publication) 27 December 2017 (30 December 2017)

*Corresponding Author; Hyun-gook Kim

Tel: +82-10-5025-0874

E-mail: hy1392@koreatech.ac.kr

1. 서론

4차 산업혁명과 ICT 기술의 중요성이 날로 증가함에 따라 소프트웨어 중심 사회가 조래되었다. 2018년부터는 공교육에서의 소프트웨어 교육이 의무화 되고, 초등학교는 실과 교과에서 연간 17시간 이상의 소프트웨어 교육을 실시하게 되었다. 또한, 중학교는 ‘정보’교과가 필수 교과로 지정되면서 연 34시간 이상의 교육을 실시하게 되었다. 이러한 교과 과정의 변화를 대비하여 국내에서는 현재 전국에 약 1,200개교의 소프트웨어 선도학교가 운영 중에 있는 상황이다. 소프트웨어 선도학교는 소프트웨어 교육을 미리 실시하며 소프트웨어 교육의 방향성에 대해 연구하고, 학생들이 다양한 소프트웨어를 경험할 수 있는 경험의 장을 제공하고자 노력하고 있다. 여기서 중요한 것은 학생들(학습자)이 다양한 소프트웨어를 경험할 수 있는 환경을 만들어야 한다는 것이다.

기존 소프트웨어 교육을 실시 할 때, 가장 제한적인 것은 소프트웨어를 구동 시켜 학습 환경을 구축하는 비용에 대한 것이었다. 이러한 제한 사항을 해결할 수 있는 것으로 대표적인 기술이 웹 컴파일러다. 웹 컴파일러란 웹 환경에서 다양한 언어에 대한 컴파일 결과를 제공해 주는 것을 의미한다. 즉, 인터넷에 연결이 가능한 PC만 제공해 준다면, 학습자는 언제 어디서나 다양한 소프트웨어에 대한 학습을 진행해 볼 수 있음을 의미한다. 이러한 웹 컴파일러에 대한 서비스는 기존부터 많이 상용화 혹은 OpenAPI 형태로 제공되고 있었다. 국내외 대표적인 웹 컴파일러로는 python tutor가 있다. python tutor는 python coding을 보다 쉽고 빠르게 할 수 있는 환경을 제공해주며, visualize한 콘텐츠를 제공함으로써 그 학습의 효과를 증대 시키고 있다. coding의 경험이 적은 학습자를 위해 다양한 예제를 제공하며, step by step으로 실습할 수 있는 방법을 제시하고 있다. 또한, 부가적인 설치 과정 없이 웹을 통해 접속하여 자신의 학습을 진행할 수 있다는 가장 큰 장점을 갖고 있다. 다음의 그림 1은 python tutor의 실행 모습이다.

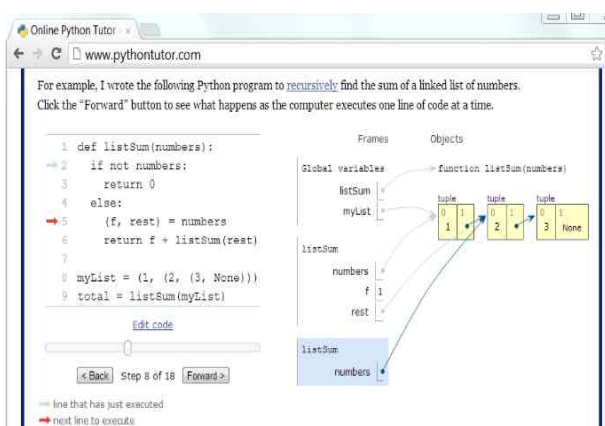


그림 1. Python tutor 웹 컴파일러
Fig. 1. Web compiler of python tutor.

위에서 언급한 것과 같이 다양한 S/W 언어를 학습할 수 있는 환경을 지원하는 웹 컴파일러에 대한 많은 연구와 개발이 진행되어 있다. 하지만, 최근 대두되고 있는 인공지능을 위한 웹 컴파일러 환경에 대한 연구는 미비한 상황이다.

인공지능은 4차 산업혁명이 주목을 받으면서 ICT의 핵심기술 중 하나로 자리매김하고 있다. 4차 산업혁명에는 ‘초지능화’라는 특성이 존재하고, 이를 가능하게 하는 것이 인공지능 기술이다. 산업 시장에서도 딥 러닝 등 기계학습과 빅데이터에 기반을 둔 인공지능이 급성장할 것으로 예상하고 있으며, 국내외 적으로 인공지능을 탑재한 머신의 규모가 날로 급성장하고 있다. 정부에서는 기반 기술 확보를 위한 교육 차원에서의 조치로써, 4차 산업 혁명 기술 변화에 맞춘 교육의 고도화와 기술연수를 확대 개편하였으며, ‘4차 산업분야’ 과정을 포함한 첨단 신산업 관련 기술연수를 신설하기도 하였다.

따라서 본 논문에서는 인공지능 교육에 필요한 웹 컴파일러의 효율적인 설계 방안 및 구현에 관한 연구를 진행하였다. 국내 실정에 맞는 초, 중, 고급 엔지니어 교육을 위한 인공지능 교육 시스템에 사용할 목적으로 진행하였으며, 인공지능 교육에서 가장 많이 사용되는 Tensorflow 기반의 웹 컴파일러를 설계하였다 [1-2].

II. 웹 기반 인공지능 컴파일러의 이해

인공지능 웹 컴파일러를 설계하기 위하여 인공지능 오픈 라이브러리인 Tensorflow와 Tensorflow-Serving, 백 엔드 컨테이너를 위한 도커를 사용하고자 한다.

2-1 관련 기술 동향

1) Tensorflow

텐서플로우(tensorflow)는 구글에서 오픈소스로 공개한 기계 학습 라이브러리이다. 텐서플로우는 리눅스나 맥의 운영체제인 OS X 뿐 아니라 Bash가 설치된 윈도우 10에서도 구동되는 2세대 기계학습 플랫폼이다[3-4]. 이 라이브러리는 구글이 내부적으로 사용해온 머신러닝 시스템 ‘디스트빌리프’의 후속 버전이다. 텐서플로우의 대표적인 특징은 (1) 데이터 플로우 그래프를 통한 풍부한 표현력 (2) 코드 수정 없이 CPU(central processing unit)/GPU(Graphics Processing Unit) 모드로 동작 (3) 아이디어 테스트에서 서비스 단계까지 이용 가능 (4) 계산 구조와 목표 함수만 정의하면 자동으로 미분계산을 처리 (5) Python/C++를 지원하며, SWIG(Simplified Wrapper and Interface Generator)를 통해 다양한 언어 지원 가능 과 같다.

텐서플로우는 이미지, 음성, 비디오 등의 대용량 데이터를 처리하며, 기계학습을 수행할 때 고속 병렬처리가 가능한 GPU도 활용한다. 텐서플로우의 핵심 알고리즘은 C++로 절차를 진행하기 위한 프론트 엔드는 파이썬으로 작성되었다[5].

지도학습 과정은 그림 2와 같이 먼저 학습하고자 하는 데이

터를 준비하고 이 후 데이터를 가지고 학습을 수행한다.

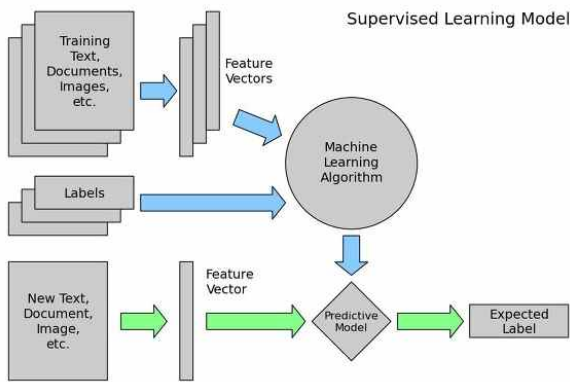


그림 2. 지도학습 절차
Fig. 2. Supervised learning model procedure.

학습중인 모델이 과적합이나 과소적합 인지를 검증한 후, 최적의 적합 구간을 찾아 최종 학습 모델을 정한다. 일련의 학습 과정을 모두 완료한 후 시험 데이터를 통해 최종 학습 모델의 성능을 평가한다[6].

구글에서는 텐서플로우를 활용하여 구글 포토(강아지, 고양이, 자동차, 사람 얼굴, 장소 등 구분할 수 있는 지능 습득), 랭크브레인(문장이 애매한 키워드를 입력해도 검색결과 출력), 스마트 답장(Gmail 인박스가 email의 내용을 이해하고 적절하게 짧은 답장을 자동생성) 서비스를 시행 중에 있다.

2) Tensorflow-Serving

텐서플로우의 활용을 도와주는 도구로서의 역할을 갖고 있으며, 텐서플로우를 활용한 머신러닝 모델을 구축하는 데 도움을 준다. 깃허브를 통해 오픈소스로 공개하였으며, 이를 이용하면 새로운 알고리즘이나 실험을 배치하는 작업이 쉬워진다. 텐서플로우 서빙은 퍼포먼스에 최적화되어 있다. 초당 10만 쿼리를 처리할 수 있으며, 처리속도를 올리기 위해서 그래픽 처리장치를 사용할 수 있다[7].

3) 도커

도커는 컨테이너 기반의 오픈소스 가상화 플랫폼이다. 2017년 설문조사에 따르면 90%가 개발에 도커를 사용 중이고, 80%가 DevOps에 사용할 예정이며, 58%가 운영환경에 사용 중이라고 답변했다.

도커는 서버에서 다양한 프로그램, 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 단순하게 해준다. 백엔드 프로그램, 데이터베이스 서버, 메시지 큐 등 어떤 프로그램도 컨테이너로 추상화 할 수 있고, 어디에서든 실행할 수 있다[8].

기존의 가상화 방식은 주로 OS를 가상화 하였다. VmWare나 VirtualBox 같은 가상머신은 전가상화 방식이라고 하는데, 호스트 OS위에 게스트 OS 전체를 가상화 하여 사용하는 방식이

다. 이 방식은 여러 가지 OS를 가상화 할 수 있고, 비교적 사용

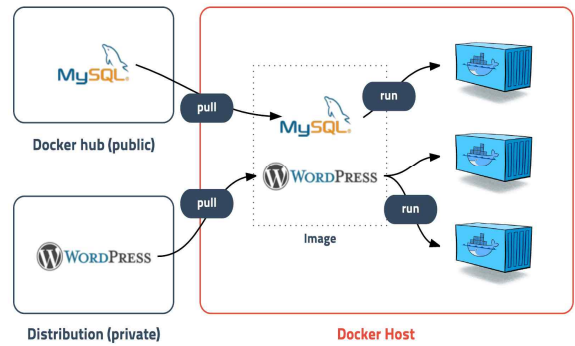


그림 3. 도커 이미지
Fig. 3. Docker image.

법이 간단하지만 무겁고 느려서 운영환경에선 사용할 수 없다. 이러한 상황을 개선하기 위해 CPU 가상화 기술(HVM, Hardware Virtual Machine)을 이용한 KVM(kernel-based virtual machine)과 반가상화 방식의 XEN이 등장하였다. 반가상화 방식은 게스트 OS가 필요하긴 하지만 전체 OS를 가상화 하는 방식이 아니었기 때문에 전가상화 방식에 비해 성능이 향상되었다. 이러한 기술들은 OpenStack 이나 AWS(Amazon Web Services), Rackspace 같은 클라우드 서비스에서 가상 컴퓨팅 기술의 기반이 되었다. 전가상화든 반가상화든 추가적인 OS를 설치하여 가상화하는 방법은 성능문제가 발생했고, 이를 개선하기 위한 방법으로 프로세스를 격리하는 방식이 연구되었다. 이러한 방식으로 리눅스 컨테이너(LXC, LinuX Containers)가 있고, 도커는 이를 기반으로 시작하였다. 도커에서 가장 중요한 개념은 컨테이너와 이미지라는 것이다. 위의 그림 3은 도커 이미지에 대한 설명이다. 이미지는 컨테이너 실행에 필요한 파일과 설정 값들을 포함하고 있는 것으로 상태 값을 가지지 않고 변하지 않는다. 컨테이너는 이미지를 실행한 상태라고 볼 수 있고 추가되거나 변하는 값은 컨테이너에 저장된다. 이미지는 컨테이너를 실행하기 위한 모든 정보를 가지고 있기 때문에 더 이상 의존성 파일을 컴파일하고 이것저것 설치할 필요가 없다. 그림 4와 같이 레이어 저장방식을 사용하여 새로운 서버가 추가되면 미리 만들어 놓은 이미지를 다운받고 컨테이너를 생성만 하면 된다.

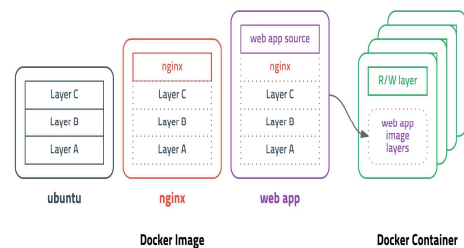


그림 4. 도커 레이어
Fig. 4. Docker layer.

III. 웹 기반 인공지능 컴파일러 구현

3-1 전체 시스템 구성

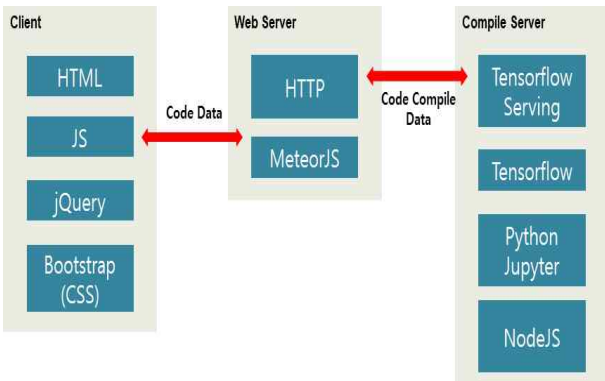


그림 5. 웹 컴파일러 시스템 설계
Fig. 5. Design of web compiler system.

전반적인 전체 시스템 구성은 위의 그림 5와 같다. 학습자가 접속하여 학습을 진행하는 클라이언트(프론트 엔드), 웹 상에서의 사용자 요청을 처리하는 웹서버 및 인공지능 알고리즘을 처리해 주는 컴파일서버(백 엔드)로 구성된다. 클라이언트는 웹 페이지 및 기능을 위해 HTML(HyperText Markup Language), CSS(Cascading Style Sheets), jQuery, Bootstrap을 사용하며, 웹 서버와 컴파일 서버는 각각 MeteorJS, 텐서플로우, 텐서플로우 서빙, 파이썬 주피터, 도커, NodeJS를 활용한다.

3-2 프론트 엔드 설계 및 구현

시스템의 프론트 엔드는 학습자에게 정보를 제공하고, 학습자의 특정 행동(이벤트)에 대한 전달 및 처리결과 화면 출력의 역할을 담당한다. 프론트 엔드의 요구사항은 다음과 같다. 요구사항1. 제공되는 웹페이지는 반응형에 기반을 두어 사용자의 장치 및 기기의 해상도에 따라 유동적인 페이지 구성을 제공한다. 요구사항2. 사용자는 로그인, 마이페이지, 학습페이지의 기능을 갖춘 웹 페이지를 통해 서비스를 제공한다. 요구사항3. 사용자에게 로그인 페이지를 통해 로그인, 회원가입, 로그아웃 등 계정 정보에 관련된 서비스 제공한다. 요구사항4. 사용자에게 마이페이지를 통해 개인정보 수정, 내 학습 현황, 내 학습 관리 등 개인 정보 및 수강 정보에 관련된 서비스 제공한다. 요구사항4. 사용자에게 학습페이지를 통해 과목 리스트 조회, 과목 상세보기, 학습 진행, 코드 등록하기, 코드 테스트하기 등 학습 진행 및 코드 실습에 관련된 서비스 제공한다.

이와 같이 웹 컴파일러 시스템의 목적은 학습자가 작성한 코드를 패키징하여 웹서버에 전달하고, 컴파일 결과를 화면에 보여주는 것 외에 다양한 이벤트 요청에 대한 즉각적인 반응이다. 이를 위해 레이아웃은 HTML과 Bootstrap, CSS 를 통해 구현하고, 패키징 및 동작은 jQuery를 통해 구현하였다. 또한, 프론트

엔드는 MeteorJS 서버 기반으로 동작한다. 프론트 엔드 측면의 시스템 세부 요소는 다음의 표 1과 같다.

표 1. 프론트 엔드 시스템 상세 구성요소

Table 1. Elements of frontend system.

function	details
Login	Log in using the information of the user registered in the web server
Logout	Log out the information currently logged in on the web page
Register	Register an user using information of users not stored in web server
Modify user information	Change the ID, PW, etc. of the user registered in the web server to a different value
My learning status	View the percentage of learning currently in progress for the user who is currently logged in. Display the learning period of the current learning
Manage my learning	Pause or cancel current learning Cancel current learning
Manage my courses (Teachers only)	Manage learning information of users who are currently taking courses
Course List	List of courses that is offered and being offered Provided learning Course search function Provided Learning Course Filtering Function Detailed inquiry function provided
Course Details	Course Detailed information inquiry page Provide users with an overview of the courses offered, learning objectives, learning details, etc. Enrich learning on My Page by taking classes
learning	Actual learning progress page Actual learning progresses according to the subject curriculum
Code registration	Tensor flow code registration page Registration and learning function of user's tensor flow code to server
Code test	Registered tensor flow code test page Execute the tensor flow code registered by the user in advance and output the result value

아래의 그림 6은 프론트 엔드 세부요소와 설계를 바탕으로 구현한 화면이다. 학습자를 위한 기능과 정보제공 페이지로 구성되며, 다양한 언어 컴파일 모듈을 추가할 수 있도록 구성하였다.

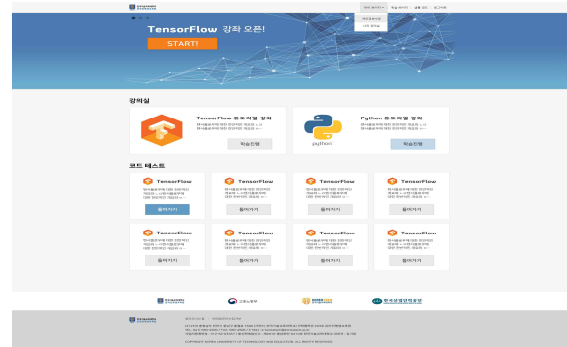


그림 6. 프론트 엔드 구현
Fig. 6. Implement of frontend system.

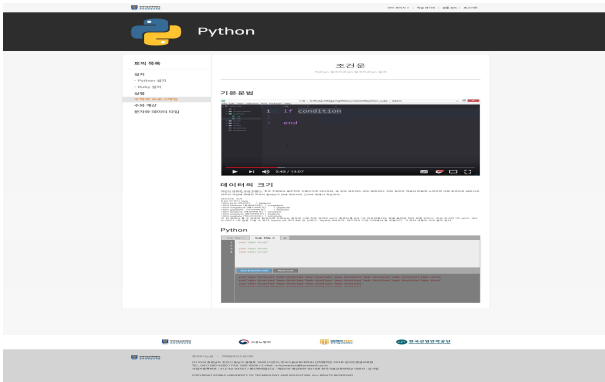


그림 7. 웹 컴파일러 시스템 구현
Fig. 7. Implement of coding system.

위의 그림 7은 프론트 엔드에서 구현한 코딩 시스템의 화면이다. 다양한 멀티미디어 자료를 학습자에게 제공할 수도 있으며, 직접 소스를 작성하고, 그 결과를 확인해 볼 수 있다.

3-3 백 엔드 설계 및 구현

백 엔드는 웹 서버와 컴파일러 서버로 나뉜다. 백 엔드 시스템의 구조는 아래의 그림 8과 같다. 웹 서버는 프론트 엔드와 연결되어 HTTP(HyperText Transfer Protocol) 통신을 통해 학습자와 직접적인 연결 및 반응을 제공하며, 컴파일러 서버는 웹 서버로부터 전달 받은 컴파일 소스를 컴파일 후 그 결과를 웹 서버에게 전달해 준다. 웹 서버는 다시 프론트 엔드에 결과를 반환해 준다. 또한, Sandboxing을 이용하여 코드의 적합성을 검사하고, 이를 위해 도커를 활용한다. 또한, 코드의 가독성을 위해 Syntax highlighting을 제공해 준다. 텐서플로우에는 미리 모델을 학습시켜 그 결과를 텐서플로우 서빙을 통해 웹서버로 전달해 줄 수 있다. 이러한 백 엔드 구현에 필요한 기술은 MeteorJS, 텐서플로우, 텐서플로우 서빙, 파이썬 주피터, 도커, NodeJS 이다. 백 엔드 측면의 시스템 상세 세부요소는 다음의 표 2와 같다.

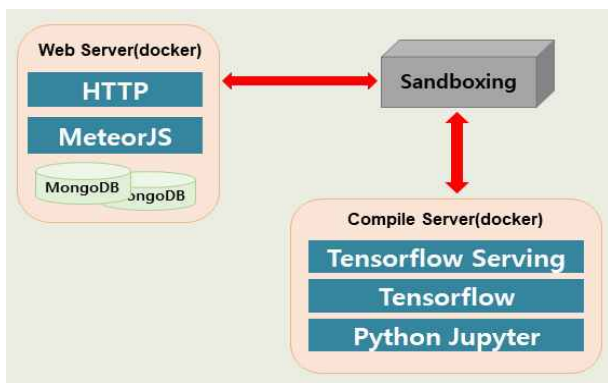


그림 8. 백 엔드 시스템 설계
Fig. 8. Design of backend system.

표 2. 백 엔드 시스템 상세 구성요소

Table 2. Elements of backend system.

function	details
Save user account information	Store user ID information Store user password information Sign up and sign in
Save courses information	Save detailed information on subjects such as subject name, subject goal, type of learning activity, content type Saving information for each user
Save source code information	Save detailed information such as file name, path, etc.
User input handling	Page navigation, log-in, inquiry of course information, inquiry of source code, database inquiry based on user input, and service of tensor flow service
Database lookup	Checking account information validity and redundancy through login and sign-up database inquiry List of courses per user, list of saved source codes
Returning the code result	Return the execution result of the code registered by the user
Tensor flow model registration	Register source code registered by user in tensor flow service Generate a tensor flow model after learning based on the source code registered by the user
Tensor flow model version management	Version control by registering, updating, and deleting the registered tensor flow model
Driving the tensor flow model	Run the registered tensor flow model on the server Output computation result using registered model and user input through gRPC communication

아래의 그림 9는 컴파일 서버와 웹 서버 환경 구축 후 테스트한 결과이다. 텐서플로우에 모델을 학습하고, 이를 HTTP 통신을 통해 학습자의 요청을 받아 서버에서 처리 후 그 결과를 전송하는 형태로 이루어진다. 그림 9는 학습한 모델에 학습자의 요청을 처리 후 전송한 결과이다.

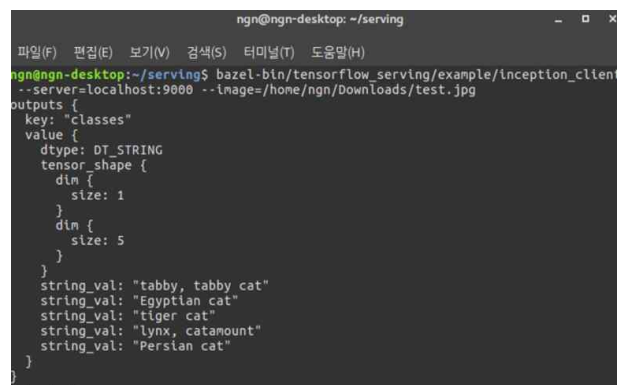


그림 9. 텐서플로우 학습 결과 및 구동 테스트
Fig. 9. Learning result-set and operating test of tensorflow.

IV. 결 론

본 논문에서는 4차 산업혁명으로 인해 주목받고 있는 인공지능 기술의 습득, 학습을 돕기 위한 웹 컴파일러의 설계 방안 및 구현에 대한 연구를 진행하였다. 웹 컴파일러는 다양한 언어를 습득할 수 있는 환경을 제공해 주며, 학습자로 하여금 언제 어디서나 활용 가능하게 할 수 있다는 장점이 있다. 기존 다양한 언어를 지원하는 웹 컴파일러가 존재하지만 인공지능을 위한 웹 컴파일러는 미비한 상황이다. 국내에서도 인공지능 기술의 중요성이 날로 증가하고 있고, 정부에서는 인공지능 교육을 위한 새로운 방안을 세우고 있는 상황에서, 인공지능 웹 컴파일러를 효율적으로 활용 가능할 것이다.

본 연구에서는 구글에서 제공하는 인공지능 라이브러리인 텐서플로 기반의 웹 컴파일러를 설계 구현하였다. 텐서플로우는 이미지, 음성, 비디오 등의 대용량 데이터를 처리하며, 기계학습을 수행할 때 고속 병렬처리가 가능한 GPU도 활용한다. 이를 활용하기 위해 텐서플로우 서버를 연동하여 텐서플로우 학습 모델 및 컴파일 결과를 웹 서버로 전달하였고, MeteorJS 기반의 웹 서버에서는 이를 바탕으로 학습자에게 결과를 피드백해주는 구성이다. MeteorJS 기반의 웹 서버에서는 프론트 엔드 언어로 구성된 학습 사이트와의 이벤트 연결 및 처리를 통해 학습자의 다양한 요구에 대한 처리를 담당한다. 웹 컴파일러는 SW 언어를 학습하는 새로운 형태로 자리매김하고 있다. 인공지능뿐만 아니라 사물인터넷 HW 설계, 마이크로 코드 등 다양한 최신 ICT 기술에 접목가능하며, 이를 위한 컴파일러 설계 방안에 대한 발전적 연구가 진행되어야 할 것이다.



박진태 (Jin-Tae Park)

2015년 8월 : 컴퓨터공학과 공학석사

2015년 9월 ~ 현재 : 한국기술교육대학교 컴퓨터공학과 박사과정

*관심분야 : Web Assembly, Standardization of Web Technologies, Web Application Engineering



김현국 (Hyun-Gook Kim)

2017년 8월 ~ 현재 : 한국기술교육대학교 컴퓨터공학과 석사과정

*관심분야 : 사물인터넷, 웹 어셈블리, 웹 표준



문일영 (Il-Young Moon)

2000년 2월 : 한국항공대학교 항공통신정보공학과 졸업 (공학사)

2002년 2월 : 한국항공대학교 대학원 항공통신정보공학부 졸업 (공학석사)

2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업 (공학박사)

2004년 ~ 2005년 : 한국정보문화진흥원 선임연구원

2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수

*관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP

Acknowledgments

본 연구는 2017년도 한국기술교육대학교 능력개발교육원의 지원에 의하여 이루어진 연구로서, 관계 기관의 지원에 감사드립니다.

References

- [1] Y. J. Song, and J. W. Lee. "A design and implementation of deep learning model for stock prediction using tensorflow," KIISE, pp.799-801, Jun.2017.
- [2] J. K. Kim, Y. H. Kim, M. S. Lee, and J. M. Ann, "Performance evaluation methodology on BPSK with tensorflow," KICS, pp.339-340, 2017.
- [3] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning," OSDI. Vol. 16. 2016.
- [4] Google - MNIST For ML Beginners : <https://www.tensorflow.org/>
- [5] J. K. Kim, "Performance evaluation methodology on BPSK with tensorflow," KICS, South Korea, pp.339-340, 2017.
- [6] Y. Lecun, Y Bengio, and G Hinton, "Deep learning," *Nature*, no. 521, pp 436- 444, May 2015.
- [7] tensorflow - deploy : <https://www.tensorflow.org/serving/>
- [8] docker - what is docker : <https://www.docker.com/>