Jess | Portfolio

Hello,
I am a junior engineer who believes that accumulated data can change the world.

📧 E-Mail: dev.bearabbit@gmail.com

📂 Blog: https://hyeonji-ryu.github.io

⚒️ GitHub: https://github.com/Hyeonji-Ryu

📄 Resume: https://hyeonji-ryu.github.io/Resume/

😊 About Me

- My motto is ‘Dev Anything’, and I try to learn without the boundaries of field.
- I’m interested in quality of data and distributed system.
- I live with 6 cats.
- I love coffee, but I keep my rule of one cup a day.

🏢 Work Projects

Project Name	Period	Skills ans Tools	Company
Bulid bigdata system of MG Bank	2021.08 - 2022.03	Cloudera Hadoop, Sqoop, Hive, Impala, Kudu	MG Bank
Develop cluster auto-building module	2021.06 - 2021.08	Shell script, Ansible, Centos	GIT

🧑🏻‍💻 Side Projects

Project Name	Period	Skills ans Tools	Product
KBO Scraping Package	2022.01 - 2022.02	Python, Poetry	Package
KBO Analytical Dashboard	2020.10 - 2020.11	Python, Flask, Bootstrap, Dash, Plotly, MariaDB	Dashboard
Deep learning from scratch in Julia	2020.05 - 2020.08	Julia, Tensorflow	Lecture

🌈 Activities

Community

- Pycon Korea Organizer
  2021.02 - Present

ETC.

- AI study - EfficientNet
  2020.10.18
- Translate Julia Docs
  2020.02 - 2020.03

Build bigdata system of MG Bank

Work Project

Summary

The goal of this project is to build a big data system for data analysis and to integrate separated data into a single data warehouse. To this end, we build a cluster with multiple servers and build a big data platform using Cloudera Enterprise Hadoop. After that, we collect and load data from several DB using Sqoop, NiFi, etc

Period

2021.08 - 2022.03

Skills & Tools

Cloudera Hadoop, Sqoop, Hive, Impala, Kudu

Position

- Build a big data platform with Cloudera Enterprise Hadoop and set up security (Kerberos, TLS)
- Support making of workflows about data warehouse and tuning query using Sqoop, Hive, Impala
- Make a workflow to load externally collected data to HDFS using NiFi
- Create and build custom Docker images to provide packages requested by the analytics team
- Improve query performance more than 20% by generating Impala statistical data

What I Learned

- I learned creating batch program using Shell. The batch process that used in this project is below.
    - Sqoop import → put HDFS → make Hive schema
- In Kudu, schema change and CURD are possible unlike HDFS. So it is a storage suitable for tables that are being updated. However, Kudu has its own limitations.
    - Recommend number of tablets per tablet server is 1000 (up to 2000)
    - Recommend loading less than 8TB of data per tablet server
    - Recommend number of Tablets per table under 60 and columns per table under 300
    - Primary key is required, ERD is not used
    - Recommended storing less than 50 GB per tablet. → performance issues
- NiFi is not suitable for large batch jobs. use NiFi to store data files requested by our analysts in real time in Hadoop.
- Creating statistical data for the Impala table improves query performance because the planner optimizes the query plan.

Develop cluster auto-building module

Work Project

Summary

This project develops modules that make it easier to build and operate Hadoop clusters. Modules are largely divided into Ansible installation, hosts creation, cluster environment distribution, kerberized, etc. This no need to set up the environment by connecting to each cluster server, and it is possible to manage cluster server information at once in hosts file.

Period

2021.06 - 2021.08

Skills & Tools

Shell script, Ansible, Centos

Position

- Automate cluster building tasks that were previously performed manually by using Ansible
- Write a playbook for automating the environment setting of each server in a cluster
- Write a playbook that links the cluster with Active Directory, an account management tool

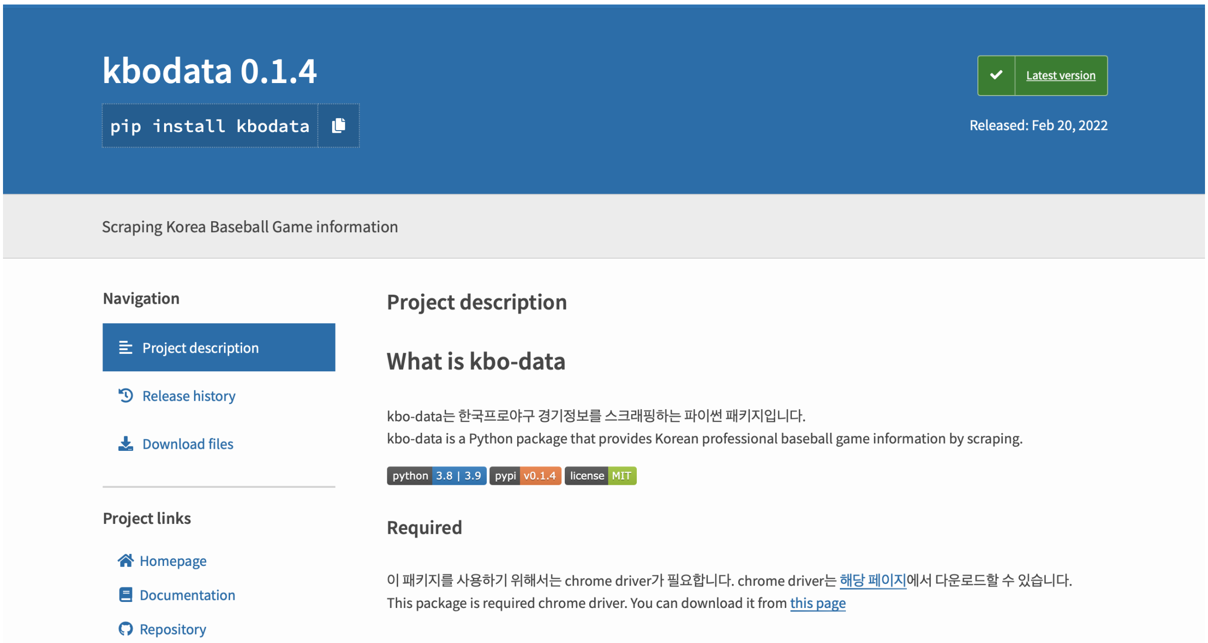
What I Learned

- Playbook is written in YAML file. It was definitely easier to write than XML and JSON, and the schema structure was clearly visible. But, sometimes when the number of space is wrong, it was difficult to reallize it.
- Variables in the host file can be used in the playbook. Through this, all the items that may be different for each customer are set as variables and can be changed in the host file.
- I learned about Kerberos protocol and Active Directory.
  - AD supports Kerberos as one of the protocols for user authentication.
  - It is linked with AD using the realm join command in a Linux.
  - It manages the OU by dividing it into Computer(Server) / SPN / Group / User in AD.
  - In system account, Jobs can be executed without password leakage using keytab.

KBO Scraping Package

Side Project

PyPi: https://pypi.org/project/kbodata/
Github: https://github.com/Hyeonji-Ryu/kbo-data/blob/main/README.md



Scraping Korea Baseball Game information

Project description

What is kbo-data

kbo-data는 한국프로야구 경기정보를 스크래핑하는 파이썬 패키지입니다.
kbo-data is a Python package that provides Korean professional baseball game information by scraping.

Required

이 패키지를 사용하기 위해서는 chrome driver가 필요합니다. chrome driver는 해당 페이지에서 다운로드할 수 있습니다.
This package is required chrome driver. You can download it from this page

Summary

This project was started to provide easily KBO data for sabermetric analysis. This package gets the game schedule for a specific date and then gets the game data for the schedule date. The first data imported is a JSON structure, which is divided into team, batter, and pitcher information and provided in the form of Dataframe and Dict.

Period

2022.02. - 2022.03

Skills & Tools

Python, Poetry

Position

- Develop of module to scrape data from KBO homepage
- Create ERD and data modeling through data analysis
- Create and upload package source code with Poetry

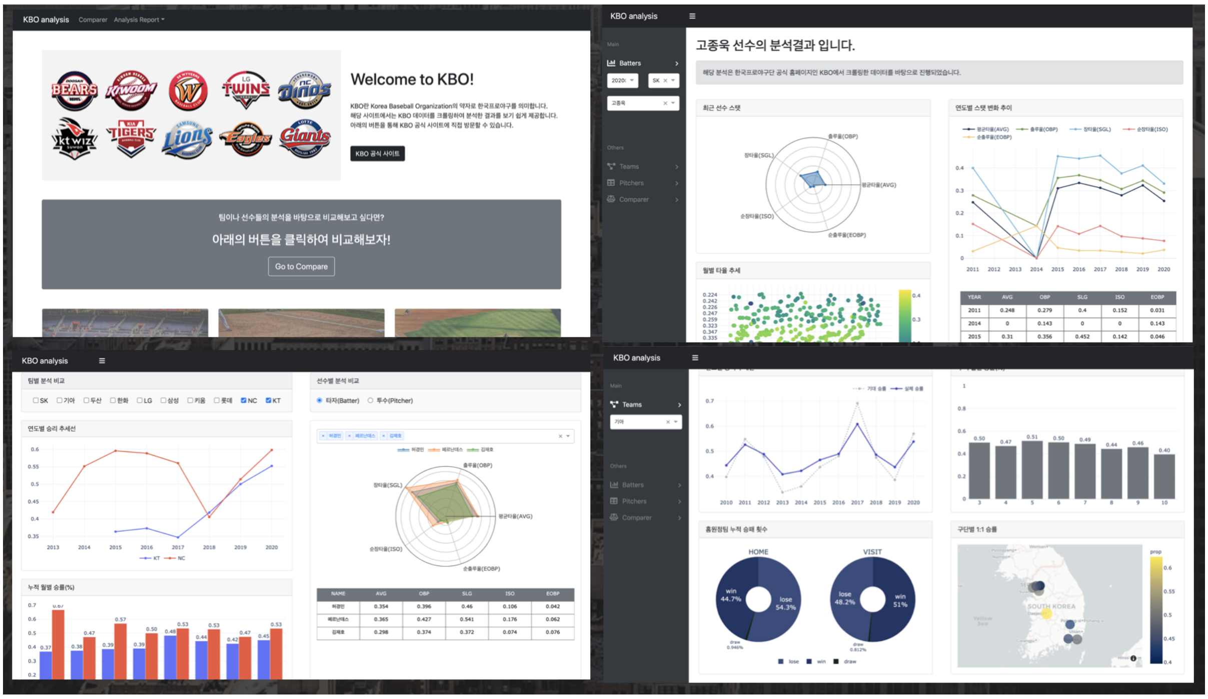
What I Learned

- I learned Poetry, a Python package management library, and used it.
  - Create easily a package directory structure with the Poetry command.
  - Manage easilly package info as pyproject.toml instead of setup.py, requirement.txt.
  - Deploy easily packages to PyPI with the Poetry publish command.
- Before web scraping, It has to be checked robot.txt to see which URLs can be collected.
- Use configparser library, It easily changed and managed environment setting.
- When using selenium, add option "headless" to speed up.

KBO Analytical Dashboard

Side Project

Dashboard: http://kbo.lopes.co.kr
Github: https://github.com/Hyeonji-Ryu/KBO-dashboard/blob/main/README.md



Welcome to KBO!

KBO란 Korea Baseball Organization의 약자로 한국프로야구를 의미합니다.
해당 사이트에서는 KBO 데이터를 크롤링하여 분석한 결과를 보기 쉽게 제공합니다.
아래의 버튼을 통해 KBO 공식 사이트에 직접 방문할 수 있습니다.

팀이나 선수들의 분석을 바탕으로 비교해보고 싶다면?

아래의 버튼을 클릭하여 비교해보자!

고종욱 선수의 분석결과 입니다.

해당 분석은 한국프로야구단 공식 홈페이지인 KBO에서 크롤링한 데이터를 바탕으로 진행되었습니다.

YEAR	AVG	OBP	SLG	ISO	EOBP
2011	0.248	0.279	0.4	0.152	0.031
2014	0	0.143	0	0	0.143
2015	0.31	0.356	0.452	0.142	0.046

☐SK ☐기아 ☐두산 ☐한화 ☐LG ☐삼성 ☐키움 ☐롯데 ☑NC ☑KT

● 타자(Batter) ○ 투수(Pitcher)

NAME	AVG	OBP	SLG	ISO	EOBP
허경민	0.354	0.396	0.46	0.106	0.042
페르난데스	0.365	0.427	0.541	0.176	0.062
김재호	0.298	0.374	0.372	0.074	0.076

Summary

I started this project to learn Flask. I use Flask as a simple web server, and the data collected from KBO was stored in the DB and then linked using SQLAlchemy ORM. The data on the dashboard are the results of using simple baseball data analysis statistical techniques, and when additional analysis models are completed, we plan to upload them to the relevant web page.

Period

- 1st updated: 2020.10 - 2020.11
- 2nd updated: 2022.02 - 2022.03

Skills & Tools

Python, Flask, Bootstrap, Dash, Plotly, MariaDB

Position

- Develop of analytic functions based on KBO data
- Build a basic architecture using Flask and Dash libraries
- Create visualizations graph using the Plotly library
- Use MariaDB and SQLAlchamy ORM
- Create dashboard design using Bootstrap5
- Deploy server with AWS lightsail
- Update to data DB through Github Action

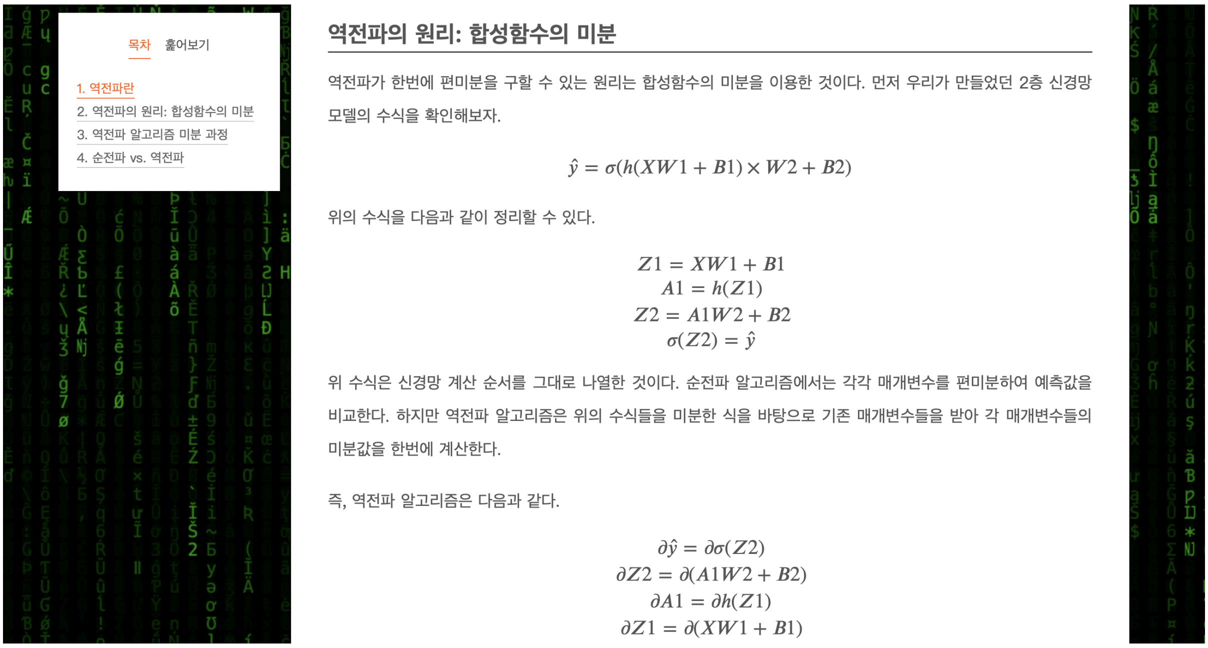
What I Learned

- Normalize database to reduce data duplication between tables.
    - delete date data of player table.
- For toy projects, AWS lightsail is much cheaper than AWS EC2.
- I learned basic HTML and CSS while using Bootstrap5.
- Create workflow through Github Action.
    - Workflow of scraping match data and adding it to DB after preprocessing.

Deep learning from scratch in Julia

Side Project

Blog: https://hyeonji-ryu.github.io/categories/PROJECT/Deep-Learning-in-Julia/
Github: https://github.com/Hyeonji-Ryu/Deep-Learning-in-Julia/blob/master/README.md



1. 역전파란
2. 역전파의 원리: 합성함수의 미분
3. 역전파 알고리즘 미분 과정
4. 순전파 vs. 역전파

역전파의 원리: 합성함수의 미분

역전파가 한번에 편미분을 구할 수 있는 원리는 합성함수의 미분을 이용한 것이다. 먼저 우리가 만들었던 2층 신경망 모델의 수식을 확인해보자.

$$\hat{y} = \sigma(h(XW1 + B1) \times W2 + B2)$$

위의 수식을 다음과 같이 정리할 수 있다.

$$\begin{gathered} Z1 = XW1 + B1 \\ A1 = h(Z1) \\ Z2 = A1W2 + B2 \\ \sigma(Z2) = \hat{y} \end{gathered}$$

위 수식은 신경망 계산 순서를 그대로 나열한 것이다. 순전파 알고리즘에서는 각각 매개변수를 편미분하여 예측값을 비교한다. 하지만 역전파 알고리즘은 위의 수식들을 미분한 식을 바탕으로 기존 매개변수들을 받아 각 매개변수들의 미분값을 한번에 계산한다.

즉, 역전파 알고리즘은 다음과 같다.

$$\begin{gathered} \partial\hat{y} = \partial\sigma(Z2) \\ \partial Z2 = \partial(A1W2 + B2) \\ \partial A1 = \partial h(Z1) \\ \partial Z1 = \partial(XW1 + B1) \end{gathered}$$

Summary

This project uses only Julia language to implement deep learning module. I made a module based on what I understood while studying deep learning. The functions used in this project are designed for educational purposes only. I uploaded some posts about the project's process on my blog, but they are written only Korean.

Period

2020.05 - 2020.08

Skills & Tools

Julia, Tensorflow

Position

- Convert formula of deep learning to code
- Write and test deep learning model scripts
- Write explanations about formulas and code

What I Learned

- I studied the basics of deep learning while reading the Deep Learning from Scratch.
- By using the Julia language array and struct, the parameter can be updated.
- I understood the formulas of deep learning layer functions and implemented them as codes.
    - Check how the training data goes through each layer
    - Write on the blog I understand while implementing the layer