류현지 | 포트폴리오

안녕하세요,
축적된 데이터로 세상을 바꿀 수 있다고 믿는
주니어 엔지니어 류현지입니다.

📧 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

- 제 좌우명은 'Dev Anything'으로 배움에 경계를 두지 않기 위해 노력합니다.
- 좋은 품질의 데이터와 분산처리시스템에 관심이 많습니다.
- 6마리 고양이와 동거동락하고 있습니다.
- 커피를 좋아하지만 1일 1잔의 약속을 지키고 있습니다.

🏢 Work Projects

Project Name	Period	Skills ans Tools	Company
새마을금고 빅데이터 시스템 구축	2021.08 - 2022.03	Cloudera Hadoop, Sqoop, Hive, Impala, Kudu	새마을금고
클러스터 구축 자동화 모듈 개발	2021.06 - 2021.08	Shell script, Ansible, Centos	굿모닝아이텍

👩‍💻 Side Projects

Project Name	Period	Skills ans Tools	Product
KBO 스크래핑 패키지 작성	2022.01 - 2022.02	Python, Poetry	Package
KBO 분석 대시보드 구축 및 운영	2020.10 - 2020.11	Python, Flask, Bootstrap, Dash, Plotly, MariaDB	Dashboard
교육용 딥러닝 스크래치 코드	2020.05 - 2020.08	Julia, Tensorflow	Lecture

🌈 Activities

Community

- 파이콘 한국 준비위원회
  2021.02 - 현재

ETC.

- 인공지능 스터디 - EfficientNet 발표
  2020.10.18
- Think Julia 문서 번역
  2020.02 - 2020.03

새마을금고 빅데이터 시스템 구축

Work Project

Summary

이 프로젝트의 목표는 데이터 분석을 위한 빅데이터 시스템을 구축하고 기존에 분리되어 있던 데이터들을 하나의 데이터 웨어하우스(data warehouse)로 통합하는 것입니다. 이를 위해 여러 대의 서버들로 클러스터를 구축하고 클라우데라 엔터프라이즈 하둡을 사용하여 빅데이터 플랫폼을 구축합니다. 이후 여러 DB에서 Sqoop, NiFi 등을 이용하여 데이터를 수집 및 적재합니다.

Period

2021.08 - 2022.03

Skills & Tools

Cloudera Hadoop, Sqoop, Hive, Impala, Kudu

Position

- 클라우데라 엔터프라이즈 하둡으로 빅데이터 플랫폼 구축 및 보안(Kerberos, TLS) 셋팅
- Sqoop, Hive, Impala를 사용하여 Data Warehouse 워크플로우 생성 및 쿼리 튜닝 지원
- NiFi를 사용하여 외부수집데이터를 HDFS에 적재하는 워크플로우 생성
- 분석팀에서 요청하는 패키지를 제공하기 위해 커스텀 Docker 이미지 생성 및 빌드
- Impala 통계데이터 생성으로 쿼리 성능 20% 이상 향상

What I Learned

- Shell 프로그램을 작성하여 배치 구성 방법을 배웠다. 해당 프로젝트에서 사용한 대략적인 배치 구조는 다음과 같다. (수집: Sqoop import → HDFS 적재 → Hive 스키마 생성)
- Kudu의 경우, HDFS와 달리 스키마 변경, CURD가 가능하기에 파기적재가 요구되는 데이터에 적합한 스토리지이다. 다만, Kudu 자체의 Limitation이 있기에 관리가 요구된다.
    - 테블릿 서버 당 권장되는 테블릿 수는 1000 (최대 2000)
    - 테블릿 서버 당 8TB 이하의 데이터만 적재 권장
    - 테이블 테블릿은 60개 이하 컬럼은 300개 이하로 사용 권장
    - 기본키는 필수로 요구되며, ERD는 구성 불가
    - 테블릿 당 저장되는 데이터는 50GB 이하로 권장 → 성능 상 문제 확인
- NiFi의 경우, 대량의 배치 잡에는 적당하지 않다. 보통 분석가들이 필요한 데이터 파일을 특정 디렉토리에 넣어두면 이를 실시간으로 하둡에 올려주는 워크플로우로 작성하는 방식으로 사용한다.
- Impala는 테이블의 통계데이터를 생성해주면 플래너가 쿼리계획을 최적화하기에 쿼리 성능이 올라간다.

클러스터 구축 자동화 모듈 개발

Work Project

Summary

이 프로젝트는 하둡 클러스터를 좀 더 편하게 구축 및 운영할 수 있는 모듈을 개발합니다. 모듈은 크게 Ansible 설치, hosts 작성, 클러스터 환경 배포, kerberized 등으로 나눠 구성하였습니다. 이를 통해 클러스터 서버에 각각 접속하여 환경을 셋팅할 필요가 없어졌으며, 클러스터의 서버 정보도 hosts에서 한 번에 관리가 가능합니다.

Period

2021.06 - 2021.08

Skills & Tools

Shell script, Ansible, Centos

Position

- Ansible을 도입하여 기존에 수작업으로 진행하던 클러스터 구축 작업을 자동화
- 클러스터로 묶이는 각 서버들의 환경 셋팅 자동화를 위한 playbook 작성
- 계정 관리 툴인 Active Directory와 클러스터를 연동하는 playbook 작성

What I Learned

- Playbook을 YAML 구조로 작성하였다. 확실히 XML, JSON 보다는 작성하기 편했고 스키마 구조도 한눈에 들어온다. 다만 들여쓰기가 잘못 들어갈 때가 있는데 이 부분을 작성 당시에는 확인하기 어려웠다.
- host 파일에 있는 변수들을 playbook에서도 사용할 수 있다. 이를 통해 고객사마다 달라질 수 있는 사항들을 모두 변수로 설정하여 host 파일에서 변경할 수 있도록 셋팅했다.
- SSO 기능을 제공하는 Kerberos 프로토콜과 Active Directory에 대해 배웠다.
    - AD에서는 Kerberos를 사용자 인증을 위한 프로토콜 중 하나로 지원한다.
    - Linux서버에서는 realm join 명령어로 AD와 연동한다.
    - 보통 AD는 OU를 Computer(Server) / SPN / Group / User 로 나눠서 관리한다.
    - 시스템 계정의 경우 keytab을 발급받아 사용하면 비밀번호 유출 없이 Job을 실행할 수 있다.

KBO 스크래핑 패키지 작성

Side Project

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

Summary

이 프로젝트는 세이버메트릭스 분석을 위한 KBO 데이터를 비교적 쉽게 제공하고자 시작하였습니다. 해당 패키지는 특정 날짜의 게임 스케쥴을 가져온 후 스케쥴 날짜의 게임 데이터를 가져옵니다. 처음 가져오는 데이터는 JSON 구조이며, 이를 팀, 타자, 투수 정보로 나누어 Dataframe, Dict의 형태로 제공합니다.

Period

2022.02. - 2022.03

Skills & Tools

Python, Poetry

Position

- KBO 홈페이지에서 데이터를 스크래핑하는 모듈 개발
- 데이터 분석을 통한 ERD 작성과 데이터 모델링
- Poetry로 패키지 소스코드 작성 및 업로드

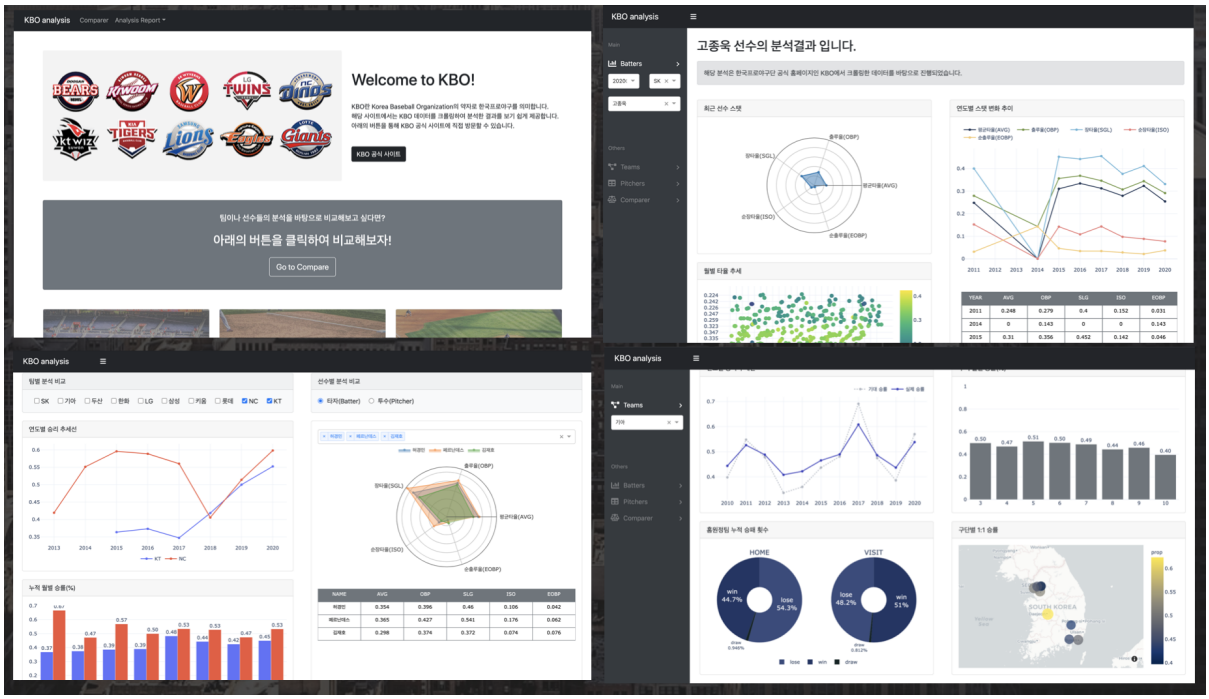
What I Learned

- Python 패키지 관리 라이브러리인 Poetry를 공부하고 이를 활용하였다.
  - Poetry 명령어로 쉽게 패키지 디렉토리 구조를 생성할 수 있음
  - setup.py, requirement.txt 의 역할을 pyproject.toml 파일 하나로 구성 가능하여 편리함
  - Poetry publish 명령어로 쉽게 PyPI에 패키지 배포 가능
- 웹 스크랩핑을 하기 전에는 robot.txt 를 확인하여 수집 가능한 URL을 확인해야 한다.
- configparser 라이브러리를 사용하면 환경 정보들을 쉽게 변경 및 관리할 수 있다.
- selenium을 사용할 때는 "headless" 모드로 사용해야 속도가 빨라진다.

KBO 분석 대시보드 구축 및 운영

Side Project

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



Summary

개인적으로 Flask를 공부하고자 시작하였습니다. Flask로 간단한 웹서버를 구성하였으며, KBO에서 수집한 데이터를 DB에 저장한 후 SQLAlchemy ORM을 사용하여 연동하였습니다. 대시보드에 나와있는 자료들은 간단한 야구데이터분석 통계기법들을 사용한 결과이며, 앞으로도 분석모델들이 추가로 완성되면 해당 웹페이지에 업로드 할 예정입니다.

Period

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

Skills & Tools

Python, Flask, Bootstrap, Dash, Plotly, MariaDB

Position

- KBO 데이터를 바탕으로 분석 함수 개발
- Flask와 Dash 라이브러리를 사용하여 기본 아키텍처 구성
- Plotly 라이브러리를 사용하여 시각화 구성
- MariaDB 구성 및 SQLAlchamy ORM 사용
- Bootstrap5를 사용하여 대시보드 디자인 구성
- AWS lightsail을 통한 서버 배포
- Github Action을 통해 데이터 DB에 업데이트

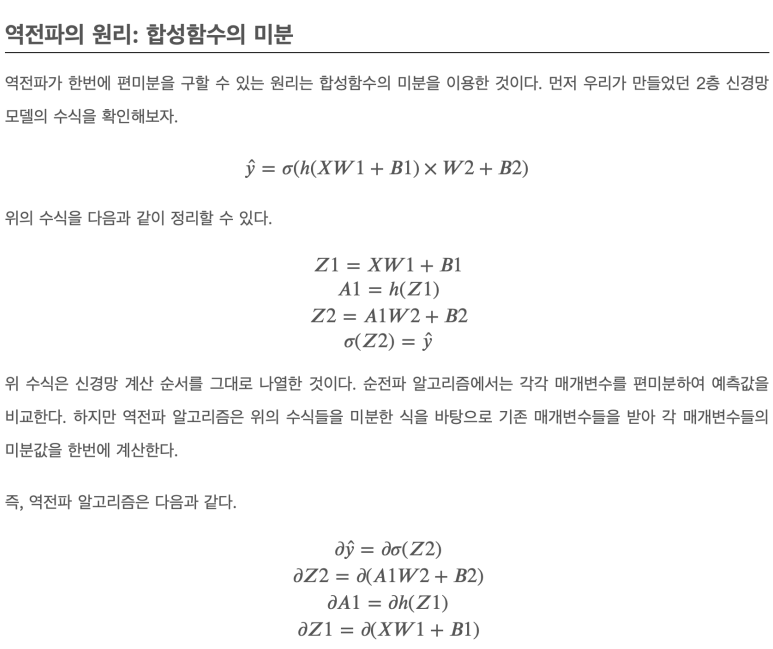
What I Learned

- 테이블 간 데이터 중복을 줄이기 위해 정규화 진행하였다.
  - 경기 데이터와 선수 데이터를 분리하여 날짜 데이터 단일화
- 토이 프로젝트에는 AWS EC2 보다는 AWS lightsail이 가격이 훨씬 저렴하다.
- Bootstrap5을 사용하면서 기본적인 HTML, CSS 지식을 습득하였다.
- Github Action을 통해 워크플로우를 구성하였다.
  - 경기데이터를 스크래핑하여 전처리 후 DB에 추가하는 워크플로우

교육용 딥러닝 스크래치 코드

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



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.

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

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.

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

Summary

이 프로젝트는 줄리아(Julia)만을 사용하여 딥러닝 모듈을 구현합니다. 개인적으로 딥러닝을 독학하면서 이해한 내용을 바탕으로 모듈을 만들고 모델을 구성했습니다. 이 프로젝트에서 사용된 함수들은 오로지 교육을 목적으로 설계되었습니다. 제 블로그의 Deep Learning in Julia 카테고리에 구현 과정이 업로드되어 있습니다.

Period

2020.05 - 2020.08

Skills & Tools

Julia, Tensorflow

Position

- 딥러닝 Layer 수식 코드화
- 딥러닝 모델 스크립트 작성 및 테스트 진행
- 수식 및 코드 설명 작성

What I Learned

- 밑바닥부터 시작하는 딥러닝 책을 완독하면서 딥러닝 기초를 공부하였다.
- Julia 언어의 배열과 구조체(struct)를 사용함으로써 Parameter가 업데이트 되도록 구성하였다.
- 딥러닝 Layer 함수들의 수식을 이해하고, 이를 코드로 구현하였다.
    - 학습 데이터가 Layer를 거쳐갈 때마다 어떤 형태로 넘어가는지 확인
    - Layer를 직접 구현하면서 이해한 내용들을 블로그에 글로 작성