<Hello Pokemon!>

CNN 을 활용한 포켓몬 이미지 식별 및 분류 모형 제작

ResNet50V2 Model 과 Nadam Optimizer 를 활용한 Transfer Learning

목적 및 목표

- 목적
  - SWFC 교육 과정 중 Machine Learning/Deep Learning 과정 내용 복습
  - 현업에서 수행할 기회가 없는 비업무 개인 창의 프로젝트 경험 습득
- 목표
  - Test Accuracy >= 0.9 : 목표 분류 정확도
  - Validation Loss <= 0.5 : Overfitting 방지

프로젝트 내용

- 모델 구성 방법
  - Pre-Trained Model 을 활용한 Transfer Learning 으로 모델 구성
  - 10 개의 Class, 각 Class 별 100 개의 이미지 데이터 직접 수집 및 전처리
  - Train Set : Validation Set : Test Set = 8 : 1 : 1
- Hyperparameter 목록
  - Activation Function : Softmax
  - Batch Size : 16
  - Learning Rate : 0.002
  - Loss : Sparse Categorical Crossentropy
  - Epochs : 50
  - Patience : 5

핵심 기술

- Train, Test, Validation 이미지 데이터 수집 및 전처리
  - 각 Class 별 Train 80 개, Test 10 개, Validation 10 개의 이미지 데이터 수집
  - 각 Class 별 포켓몬 이외에 다른 포켓몬이 나타나지 않은 이미지 데이터만 수집
  - Python Code 내에서 이미지 크기 재조정 및 정규화 작업 진행
- Pre-Trained Model 및 Optimizer 선택
  - 5 개의 Pre-Trained Model 별 Validation Accuracy Test (Optimizer : Nadam)
  - 5 개의 Optimizer 별 Validation Accuracy Test (Pre-Trained Model : ResNet50V2)
  - 각각 5 회 반복 Test 결과 ResNet50V2 와 Nadam 최종 선택

- Early Stopping 및 Data Augmentation 적용
  - 최대 epoch 횟수 = 50, 성능이 증가하지 않는 epoch 를 최대 5 회 까지만 허용
  - 최대 허용 횟수를 넘어가게 되면 더 이상 훈련을 수행하지 않고 종료
  - 이미지 좌우 반전, 회전, Zoom 등 다양한 Augmentation 기법 적용

개발 결과

- Data Augmentation 을 적용하지 않은 모델
  - Average Test Accuracy = 0.9150
  - Average Validation Loss = 0.7376
- Data Augmentation 을 적용한 모델
  - Average Test Accuracy = 0.8530
  - Average Validation Loss = 0.7930
- 결과 비교
  - Data Augmentation 을 적용하지 않은 모델이 오히려 더 좋은 성능을 보임
  - Hyperparameter 값을 다양하게 변경했을 때에도 동일한 결과 도출

결론

- 결과 분석
  - 최종 Average Test Accuracy = 0.9150 : 목표(0.9 이상) 달성
  - 최종 Average Validation Loss = 0.7376 : 목표(0.5 이하) 미달성
- 아쉬운 점
  - 더 많은 이미지 데이터를 확보했다면 인식률을 더 높일 수 있었을 것이다.
  - 각 Class 별로 Accuracy 를 분석하고 싶었으나 코드 적용에 실패하였다.

개발 후기

- 느낀 점
  - 직접 이미지 데이터를 수집하는 작업에만 이틀이 넘게 걸렸는데, 머리가 나쁘면 손발이 고생한다는 것을 다시 한번 느꼈습니다.
  - 주제를 직접 구상하고, 현업과 전혀 관계없는 데이터를 사용해서인지 오히려 프로젝트에 흥미와 애착을 더 많이 가질 수 있었습니다.
  - Test 를 수행할 때, 무작정 시작하는 것이 아니라 충분한 계획을 세우고 체계적으로 Test 해야 시간을 아낄 수 있다는 것을 깨달았습니다.