**Weekly report**

날짜 : **2022-12-29**

연구원 : **최윤석**

* **이전 수행계획**
* **차량 이미지 분류 :**
  + 실험결과 정리, 각 실험마다 mAP / 수행시간(모델 학습 , 테스트)을 테이블 형태로 정리
* **관계형 테이블 임베딩** :
  + 수치형 데이터를 이산화한 것을 범주형 데이터로 바꾸는 임베딩 방식을 보완하는 방법을 찾고 실험하여, 기존의 윤종찬 연구원의 연구와 **융합 가능도 척도**를 사용하여 성능 비교
  + 각 컬럼마다의 중요도를 고려해서 임베딩하는 방식과 수치형 컬럼에서 숫자의 크기가 크고 작음에 부여된 의미를 고려하는 임베딩 방식 연구
  + Network On Network for Tabular Data Classification in Real – world Applications 논문 읽기
  + 융합 가능도 평가 결과에 대한 다른 사람들과의 공유 및 절충
  + 나만의 딥러닝 아키텍처 그려보기, 기존의 것과 나만의 것을 분리하여 정리
* **수행결과**
* **차량 이미지 분류 :** 
  + SSD 모델 성능 표

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Epoch 10만** | **Epoch 20만** | **Epoch 20만 learning rate 점차 감소** | **Epoch 30만 learning rate 점차 감소** | **Epoch 50만 learning rate 점차 감소** |
| **mAP** | 0.920 | 0.953 | 0.958 | 0.971 | 0.972 |
| **Train Time** | 28시간 | 56시간 | 56시간 | 84시간 | 140시간 |
| **Detection Time** | 2초 / 1장 | 2초 / 1장 | 2초 / 1장 | 2초 / 1장 | 2초 / 1장 |

* learning rate 초기값은 기본 0.004 에서 0.008로 늘리고 1만 Epoch마다 learning rate에 0.9를 곱해 작아지게 설정하여 Epoch 50만으로 학습한 결과

텍스트이(가) 표시된 사진

자동 생성된 설명



* 30만 epoch와 마찬가지로 class 12를 제외하고는 전부 0.9 이상의 average\_precision을 가지는 것을 확인하였고, 미세하지만 mAP가 상승한 것을 확인.
* 차량 이미지 분류 1차 보고서 작성은 김민중 연구원과 상의한 결과, 이번주 내로 김민중 연구원이 YOLO와 데이터 부분을 작성 후 제가 SSD 파트를 추가해 완성할 예정임.
* **관계형 테이블 임베딩** :
  + 융합 가능도 평가 결과에 대한 다른 사람들과의 공유 및 절충 작업완료
  + 테이블 임베딩 벡터 2개를 묶은 한 쌍에, 융합 가능 / 불가능 2개의 라벨을 붙여 모델을 학습을 진행

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 위의 코드는 기존의 1 - 10으로 라벨링한 것에서 1의 값을 가지는 라벨은 융합 불가능(0)으로 1이외의 값을 가지는 라벨은 융합 가능(1)으로 변경하는 것

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 모델의 출력부분을 기존 10개에서 2개로 변경

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 학습한 모델의 성능은 위와 같음
  + 테이블 임베딩 벡터 2개를 묶은 한 쌍에, 융합 가능성 상 / 중 / 하 3개의 라벨을 붙여 모델을 학습을 진행

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 위의 코드는 기존의 1 - 10으로 라벨링한 것에서 5보다 크면 상(2), 1보다 크고 5보다 작거나 같으면 중(1), 1이면 하(0)로 변경하는 것

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 마찬가지로 모델의 출력 부분을 3개로 변경

텍스트이(가) 표시된 사진

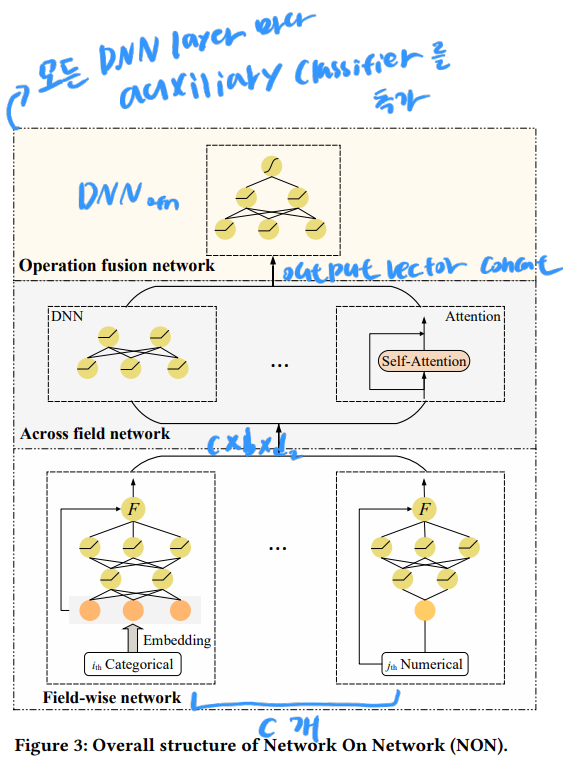
자동 생성된 설명

* + 학습된 모델의 성능은 위와 같음.
  + 기존의 1 – 10으로 나눈 라벨로도 모델 학습을 진행

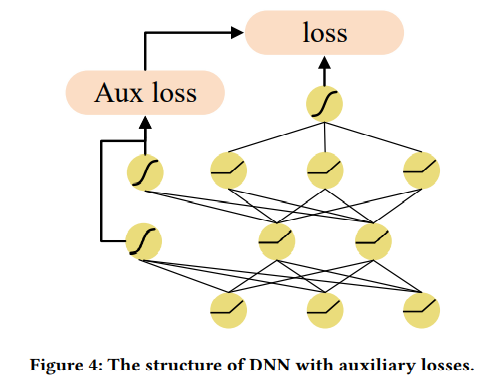
텍스트이(가) 표시된 사진

자동 생성된 설명

* + 2개의 라벨, 3개의 라벨, 10개의 라벨 전부 비슷한 성능의 모델이 나오는 것을 확인. 이는 라벨 대부분이 융합 불가능(융합 하)으로 되어있기에, 불가능으로만 예측해도 좋은 성능이 나오기에 이런 결과가 나온 것으로 생각됨. 따라서 이후 AUC를 적용해 모델의 성능을 검사할 예정임.
  + Network On Network for Tabular Data Classification in Real – world Applications 논문 읽고 정리



* + 위의 사진은 논문에서 제시한 Network On Network 구조
  + Network On Network는 3개의 파트로 이루어져 있음
    - Field-wise network : 각 필드 내부에서 나올 수 있는 정보를 포착하기 위해 존재하는 네트워크. C개의 필드(컬럼)가 있으면, 각 필드별로 DNN을 만들어 총 C개의 DNN 모델이 생김. / b : mini batch size , d2 : output size
    - Across field network : 필드간의 관계를 알기 위해 다양한 기법(DNN, self-attention, Linear Regression, FM)을 사용하는 네트워크
    - Operation fusion network : Across field netwokr의 결과를 DNN으로 합해 최종 예측을 하는 네트워크
  + 네트워크 여러 개를 쌓아서 학습이 잘 안 될 수 있기에, DNN의 레이어마다 auxiliary classifier를 추가함.



* + 기존 윤종찬 연구원의 연구는 필드별 임베딩없이 바로 레코드를 임베딩하여, 각 필드의 내부 정보를 세밀하게 반영하지 못함.
  + Filed-wise network를 사용하면 수치형 데이터의 높고 낮음의 의미를 포함하는 임베딩을 할 수 있음.
* **수행계획**
* **차량 이미지 분류** :
  + 차량 이미지 분류 1차 보고서 작성
* **관계형 테이블 임베딩** :
  + 수치형 데이터를 이산화한 것을 범주형 데이터로 바꾸는 임베딩 방식을 보완하는 방법을 찾고 실험하여, 기존의 윤종찬 연구원의 연구와 **융합 가능도 척도**를 사용하여 성능 비교
  + 각 컬럼마다의 중요도를 고려해서 임베딩하는 방식과 수치형 컬럼에서 숫자의 크기가 크고 작음에 부여된 의미를 고려하는 임베딩 방식 연구
  + 나만의 딥러닝 아키텍처 그려보기, 기존의 것과 나만의 것을 분리하여 정리
  + AUC를 적용해 모델의 성능을 제대로 검증해보고, 만약 성능이 좋지 않고, 융합 불가능으로만 예측하는 경향이 보이면 학습 데이터와 라벨을 융합 가능한 것만 존재하게 교체해서 학습
* **기타사항**
  + **데이터 융합 프로세스**



* **융합 가능도 척도**
  + 임베딩 거리 기준 값을 정하고 기준값보다 임베딩 거리가 가까우면 융합 가능한 데이터셋이라 판단함
  + 융합 가능한 데이터셋을 사람이 하나씩 확인하여 유의미한 융합이 되는 데이터셋을 찾음
  + 을 융합 가능도 척도라 정의함



혼합형 데이터셋 예시

* **1월 초 연구 내용 정리 후 발표**