18/1/4 세미나

뉴럴 넷 기반 메트릭 러닝을 위한 기본적인 알고리즘 설명 (from Siamese to triplet)

1. 메트릭  
   흔히 이야기하는 디스턴스 와 비슷한 개념으로, 두 엘리먼트 간의 유사도를 나타내는 거리개념이다. 양수여야 하며 시메트릭이고 삼각형 법칙이 정의되며 거리가 0이면 두 엘리먼트는 같다 라는 네가지 요건을 만족해야 한다. 만족하지 않는 것도 메트릭이라고 하긴 하는데 기본적으로는 만족했으면 좋겠다.
2. Metric learning   
   주어진 메트릭을 러닝을 하면 실제 공간이 변하는 것은 아니지만 메트릭이 각각의 클래스를 좀 더 잘 표현하도록 조정이 된다. 정해진 매트릭 함수가 있고 이를 잘 표현하도록 피쳐 도메인을 바꾸는 식으로 학습이 된다.   
   Q. 보면 의 A를 조절하는 것으로 표현되어 있는데, 이게 함수를 바꾸는건지 피처를 밖는건지 하는 이슈를 이해를 잘 못했음.   
   여튼 서로 같은 클래스의 엘리먼트들을 가까운 공간에 묶이도록 조정을 하는 것으로, 마진 밖의 다른 클래스 원소들이 서로 얼마나 가까이 있는지 등의 요소들은 고려 대상이 아니다. 밑에 붙어있는 요건 스텝 함수 라는건데 모르겠고 일단 패스하자.
3. 옛날옛날 것들은 패스하고 쓰게 될지도 모르는 것부터 시작을 하면 Siamese 와 Triplet 이렇게 두 가지가 있다. 용어는 다양하게 쓰이니 대충 알아서 보자.
4. Siamese를 보면 Y는 인덱스로 같을 때는 0이고 ㄹㄴㅇㅁ asd  
   이거 보면 두개의 인풋을 받아서 하나를 뱉어내는건데, Back propagation을 두번 하는 식으로 해결했다. 임지를 배치로 나눈 것 처럼.
5. 트리플렛은 인풋에 원래 인풋과 같은 클래스의 인풋 다른 클래스의 인풋 이렇게 세 개의 인풋을 넣어서 학습을 시킨다. Positive와 Negative의 ratio를 갖고 로스를 계산하도록 설계를 했고 이건 구글의 FaceNet에서 사용한 모델이라고 한다.
6. 여기서 샘플링 관련 이슈가 있고 semi hard examples sampling 줄 아래에 있는 조건을 만족하고 있는 것들을 사용했다. 즉 이미 문제를 잘 해결하고 있는 데이터들을 좀 더 잘 분류를 하겠다는 의미가 된다. 그림상에선 현재 바운더리 바깥~ 마진 안쪽의 데이터들을 참여시키겠다는 의미가 된다.   
   이걸 갖다 쓰면서 얼굴 인식의 정확도가 굉장히 잘(99%) 나오게 되어 이 필드가 끝나버렸다.
7. 적용1  
   데이터셋은 타이니 이미지넷이라고 작은게 있고 ㅇㅇ? 여기서부터 길잃음.
8. 적용2   
   트리플렛 네트워크에 분류를 위한 레이어를 하나 더 추가..   
   데이터 보면 스타일에 관한 건데 유사 기준은 손목 길이 넥라인 등등의 옷 형태에 관한 attribute 들이 있고 이 값들이 비슷할 때 유사 여부를 판단한다. 보면 분류기를 통해 얻은 로스에 웨이트 를 달아놨는데 이건 실험적으로 구해진 듯 하다. 다만 유사 하다는 사진들도 attribute가 겹치지 않는게 많으므로 분류기는 dissimilar에다 달아서 의사결정을 하도록 디자인을 했다. 라고 추정한다.   
   클래시파이어는 디싓밀러하다는 이미지의 어트리뷰트를 뽑아 내는 것이다.   
   디테일하게는 분류기가 negative 의 feature 위에 달려 있기 때문에 네거티브에 관한 피쳐는 R에 의해 바뀌고 나서도 다시한번 클레스에 따라서 다른 변화를 겪게 된다.
9. 어플3!   
   이걸 모든 데이터들의 페어를 사용하면 경우의 수가 너무 많이 나와서 결정한 것이 semi hard 이고 그걸 하고도 너무 많아서 여러가지 알고리즘들이 나온다. 그 중 하낙 ㅏ프록시 랭킹 로스가 된다. Positive를 나타내는 것들의 대표 값과 네거티브를 나타내는 값들의 대표 값을 두고 쓰는거같다. 이름은 프록시 벡터라고 부르고 거리는 그냥 유클리드 거리다. 프록시 개수를 클래스 개수보다 적게 유지하면서 학습을 하면 빠르다. 빠르다.   
   앵커는 데이터가 들어가는거고 Pos와 Neg를 실제 데이터가 아니라 가상의 점을 사용하는 것이다. 요걸 하면서 프록시도 점점 변해가는 것이다. K-mins 라는거랑 비슷한가보다. 여기 들어있는 p는 네트웍이 아니고 임베딩이다.   
   뭐 들어가는 인풋에 대해 상한 하한 norm 등등 제한 조건을 이것 저것 많이 써놓았는데 정작 실험 뒤에선 그냥 했다 라는 식으로 서술되어있다. 근데 지금 가는 중에 한 프록시로 몰리는 현상이 있는데 해결중이다.