Some thing i learn

Deep Learning & Natural Language Processing

2022 May 1

Yonsei UNiversity mirae

김호중

Table Contents

[Deep Learning 4](#_Toc102332713)

[Supervised learning 4](#_Toc102332714)

[K nearest neighbor 4](#_Toc102332715)

[Naïve bayes 4](#_Toc102332716)

[Image Features 6](#_Toc102332717)

[Histogram of Oriented Gradients 6](#_Toc102332718)

[Bag of words 7](#_Toc102332719)

[Impurity 7](#_Toc102332720)

[Decision Tree 7](#_Toc102332721)

[k-Clustering 7](#_Toc102332722)

[Principal Components Analysis 7](#_Toc102332723)

[Deep learning 8](#_Toc102332724)

[Cross validation 8](#_Toc102332725)

[Boosting 8](#_Toc102332726)

[Bootstrap 9](#_Toc102332727)

[Filter 9](#_Toc102332728)

[CNN 9](#_Toc102332729)

[Drop out 10](#_Toc102332730)

[GAN 10](#_Toc102332731)

[Gan(Generative Adversarial Network) 10](#_Toc102332732)

[Global optimality 11](#_Toc102332733)

[DCGAN 12](#_Toc102332734)

[CGAN 12](#_Toc102332735)

[LS GAN 12](#_Toc102332736)

[WGAN 13](#_Toc102332737)

[WGAN-GP 14](#_Toc102332738)

[Cycle Gan 14](#_Toc102332739)

[Style GAN 15](#_Toc102332740)

[Linear Algebra 15](#_Toc102332741)

[Linear 15](#_Toc102332742)

[대수구조(Algebraic Structure) 16](#_Toc102332743)

[군(Group) 16](#_Toc102332744)

[역원(Exsitence of inverse element) 17](#_Toc102332745)

[닫힘(Closure) 17](#_Toc102332746)

[환(Ring) 17](#_Toc102332747)

[체(field) 18](#_Toc102332748)

[선형생성 18](#_Toc102332749)

[차원 18](#_Toc102332750)

[PCA(Principal Component Analysis) 19](#_Toc102332751)

[Natural Language Processing 19](#_Toc102332752)

[Basic NLP 19](#_Toc102332753)

[Tokenization 19](#_Toc102332754)

[Cleaning & Normalization 20](#_Toc102332755)

[Lemmatization 20](#_Toc102332756)

[Integer Encoding 20](#_Toc102332757)

[One-Hot-Encoding 20](#_Toc102332758)

[N-gram 20](#_Toc102332759)

[CNN IN NLP 20](#_Toc102332760)

[RNN 21](#_Toc102332761)

[LSTM 21](#_Toc102332762)

[GRU 21](#_Toc102332763)

[Attention is All You Need 21](#_Toc102332764)

[BERT 21](#_Toc102332765)

[GPT 22](#_Toc102332766)

[Task Oriented Dialogue System 22](#_Toc102332767)

[End to End 22](#_Toc102332768)

[NLU 23](#_Toc102332769)

[DST 23](#_Toc102332770)

[DP 23](#_Toc102332771)

[NLG 23](#_Toc102332772)

[TRADE 23](#_Toc102332773)

[Slot gate 23](#_Toc102332774)

[Pointer Generator 23](#_Toc102332775)

[Multi turn problem 23](#_Toc102332776)

[Zero shot DST 24](#_Toc102332777)

[Challenge 24](#_Toc102332778)

[Reinforcement Learning 24](#_Toc102332779)

[Notation 24](#_Toc102332780)

[MDP 24](#_Toc102332781)

[State 24](#_Toc102332782)

[Action 25](#_Toc102332783)

[Reward 25](#_Toc102332784)

[Q Learning 25](#_Toc102332785)

[Emotion Recognition in Conversation 25](#_Toc102332786)

[Emotion 25](#_Toc102332787)

[Dialogue GCN 25](#_Toc102332788)

[MELD 25](#_Toc102332789)

# Deep Learning

## Supervised learning

분류와 회귀로 나뉜다. 분류는 말그대로 데이터를 클래스별로 분류하는 문제를 말한다. 스팸메일을 예시로 들 수 있는데 메일을 받았을 때에 메일을 정상적인 메일과 스팸메일로 구분하여 이메일의 안정성을 높여준다. 회귀는 데이터의 특징을 기반으로 하여 연속되어 나타내어지는 값을 예측하는 데에 사용된다. 집값예측 타이타닉 보험 등이 그 예시이다.

## K nearest neighbor

KNN은 supervised learning중 하나로 데이터가 주어졌을 때 근처의 데이터를 기반으로 어느 class에 들어갈지 판단한다. Hyperparameter k를 기반으로 하여 distance metric을 통해 이웃들을 뽑는다. KNN은 데이터가 들어오기 전 별도의 조정과정이 필요 없어 lazy model 혹은 instance model이라고 불립니다. 두번째 hyperparameter로서 distance metric을 정해 주어야한다. distance metric에는 Euclidean distance, Manhattan distance, Mahala Nobis distance 가 있으며 Mahala Nobis의 공분산 행렬이 항등 행렬일때 Euclidean distance와 일치합니다. NN은 몇 가지 단점이 존재하는데 첫째로 n개의 데이터를 모두 살펴보아야 하기에 O(n)의 복잡도를 가집니다. 둘째로 NN은 물체의 유사성인식 과 부합하지 않는 경우가 있습니다. 물체의 특성이 아닌 픽셀을 기준으로 차이를 비교한다는 점은 물체의 특성 뿐만 아니라 배경 색등에 큰 영향을 받음을 의미합니다. 위와 같은 문제는 특성기반 알고리즘이 아닌 픽셀을 기준으로 detection하는 알고리즘에서도 발생합니다.

## Naïve bayes

Bayes기반 classifier로서 조건부확률 기반 알고리즘이다. P(c|x0=p(c|x1)\* p(c|x2) )\* p(c|x3) )\* p(c|x4)….위와 같은 수식을 가지며 각각의 특징 x에대하여 발생할 확률의 곱으로 표현한다. 데이터 양이 많아 지거나 고려해야 할 특징이 많아지는 경우 효과적인 알고리즘이다. Multi class에서 빠르게 예측이 가능하며 추가적으로 독립조건이 붙는다면 logistic regression보다 효과적이다. 단점으로는 특정 class가 train data에 존재하지 않고 test data에만 존재한다면 zero frequency가 발생한다. 또한 독립이라는 조건이 쉽게 발생하지 않아 사용에 유의해야 된다.

Support vector machine

이진선형분류모델로 분류를 위해 기준선을 정의한다. 마진은 기준선과 support vector 사이의 거리로 마진이 최대가 되는 선이 최고의 기준선이다. Support vector machine은 2,3차원 이상 평면에서 데이터를 분류하며 결정경계와 인접한 포인트를 support vector이라고 한다. 이상치에 대해서 얼마나 허용을 할 것 인가에 따라 hard margin soft margin으로 나뉜다. 아웃 라이너에 대한 기준 빽빽하게 잡아 support vector과 기준선의 거리가 좁아지는 것을 hard margin이라고하며 이 반대의 경우가 soft margin이다. Hard margin의경우 overfitting의 문제가 발생할 수 있고 soft margin의 경우 underfitting의 문제가 발생할 수 있어 둘사이를 잘 조절하는 것이 중요하다. SVM에서 선형으로 분리할 수 없는 경우 kernel을 사용하여 높은 차원의 데이터로 변환시켜 분리한다.

Stochastic Gradient Descent

NN의 가중치를 조정할 때에 gradient descent를 사용한다. NN의 결과값과 실제 값의 차이를 최소화하기 위하여 미분을 이용하여 가중치를 업데이트 해주게 되는데 loss function의 값이 최소가 되는 점에 도달하기 위하여 식에서 정의된 방향으로 이동한다. Loss function을 도출해 내기 위하여 전체 train set을 사용하는 것을 Batch gradient descent라고하며 전체 데이터를 사용하기에 계산 량이 많다. 이를 해결하기 위하여 stochastic Gradient Descent를 사용하는데 이는 전체데이터가 아닌 일부 데이터를 추출하여 업데이트를 진행하기에 계산 속도가 훨씬 빠르다. Local minima에 빠질 가능성이 batch gradient descent에 비하여 낮지만 여전히 이상한 곳(saddle point)에 수렴할 확률이 존재한다.

Batch size

Batch size는 중요한 hyper parameter중 하나로 작은 size를 사용할수록 generalization 측면에 유리하다. Batch size에 따른 learning rate를 측정해 보았을 시에 작은batch일수록 안정적인 학습이 가능했으며 작은 batch사이즈가 보다 높은 test accuracy를 얻어냈다. (Batch 사이즈에 대한 조절은 모델에 따라 유동적으로 하는 것이 모델성능을 높일 수 있는 길이다)

Over fitting

Train data를 너무 잘 학습해서 발생하는 문제로 test data에 적응을 하지 못하지만 train data에서 높은 정확도를 보여줄 때 사용한다. 일반적으로 모델 complexity가 증가할수록 train test set에서 정확도가 증가하지만 임계점을 넘어가면 train data에 대한 정확도는 증가 하지만 test data에 대한 정확도는 감소한다. model capacity를 줄이거나 l1, l2 regularization, Drop out등의 방법으로 해결할 수 있다. 이와 반대되는 개념으로 underfitting이 있다. 모델이 tarin data를 너무 엉성하게 학습하여 발생하는 일로 모델의 complexity를 높여주면 된다.

Image classification

이미지를 분류하는 작업들을 말한다. 컴퓨터 비전의 주요한 부분들 중 하나이다. 인간의 프로세스와 컴퓨터의 프로세스 사이의 차이점이 있어 사람은 고양이를 물체로서 인식할 수 있지만 컴퓨터에게는 그저 0-255사이의 수의 나열에 불과하다. 같은 고양이 사진임에도 약간의 변화를 주는 것으로 수열이 뒤틀렸다 할 정도로 달라지게 된다. 예를 들어 같은 고양이사진에 명암을 조절하거나 고양이를 좌우반전 혹은 좌표를 이동해주는 것만으로도 픽셀 값은 크게 변한다. 심지어 고양이의 자세 종 등에도 큰 영향을 받는다.

Linear Classifier

NN의 기반 알고리즘이며 다양한 방면에서 활용 가능하다. 가중치 벡터에 입력x를 내적 하여 bias값을 더해 스코어를 산출해낸다. 고차원 공간으로 갔을 때 linear classifier은 클래스를 구분해주는 선이 된다. 데이터가 선형적으로 분리되지 않을 시에 사용이 힘들다는 단점이 존재하며 이를 보완하기위해 w값을 최적화 시키는 것이 중요하다.

Soft max

확률 분포를 사용하여 score에 의미를 부여할 수 있다. 보통 출력층에서 사용되며 은닉층에서의 사용은 기울기 소실 등의 문제로 학습의 효용성이 떨어질 수 있다. 점수 벡터를 클래스 별 확률로 변환하기 위해 사용되는데 벡터에 지수를 취한이후에 정규화 상수로 나누어 합이 1이되게한다. 3개이상으로 분류하는 다중 class에서 사용된다. 예를 들어 책 사진이 주어졌을 때 문학서적, 전공서적, 교양서적 등에 대해 각각 몇%확률을 가지고 있는지 알려준다.

Random process

랜덤벡터를 무한한 차원으로 확장하고자 할 때 사용한다. 주사위의 시행 횟수를 (0-inf) 로 하여 이를 t1,t2,t3라고하자 각각의 t1,t2,t3에대하여 주사위의 결과 그래프를 S라 하였을 때 S(t1),S(t2),S(t3)는 각각 확률밀도 함수를 가지게 되며 각각의 함수는 같은 값을 가질 수도 다른 값을 가질 수도 있다. 랜덤변수 index I에 대하여 표본공간 사이의 차원으로 mapping한다. 이때 I는 randomness가 없으며 샘플 공간에서 원소를 추출할 시에 함수와 변수가 달라진다. (이후에 나오는 내용들은 아직 이해가 되지 않는다.)

* 나중에 공부할 것들

## Image Features

CNN이전의 방식으로 linear classifier을 이미지에 직접 적용하는 것이 아닌 color Histogram, Histogram of Oriented gradients, Bag of words등의 형태로서 linear classifier을 실행해 왔다. Color Histogram은 이미지의 컬러를 픽셀로부터 파악한 뒤에 전체 color의 분포를 기반으로 feature을 추출하는 방식이다. 이미지의 다양한 색상이 얼마나 자주 존재하는지를 알 수 있으며 색상변화 감지에 유리하다.

## Histogram of Oriented Gradients

Histogram of Oriented Gradients는 이미지 경계의 기울기 벡터크기와 방향을 Histogram으로 나타낸다. 보행자 검출을 위해 만들어진 feature descriptor이다. 검출하고자 하는 영역을 잘라 이를 window라고 하며 Sobel filter를 사용하여 gx, gy값을 구하여 기울기를 도출해낸다. 보통 cell = 8\*8 pixel 로 사용하고 픽셀을 기준으로 구해진 기울기벡터를 히스토그램으로 나타낸다. 이후에 normalization을 통하여 밝기에 의한 경계 값 기울기 민감성을 줄여줍니다.

## Bag of words

Statistical Language Model 중에 하나로 확률 분포를 기반으로 하여 언어를 이해하도록 하는 모델이다. 자연어처리에서 주로 사용되며 단어들의 순서를 무시하고 빈도를 기준으로 하는 데이터 표현방법이다. 단순히 빈도수를 측정하는 모델이기에 문서에서 중요도가 높은 단어가 높은 빈도를 가진다면 다행이지만 그렇지 않은 경우 즉 모델의 빈도와 중요도가 비례관계가 아닐 경우에 문제가 발생한다. 또한 학습된 데이터에 지나치게 의존하기 때문에 overfitting의 가능성이 매우 크다. 위와 같은 문제점을 해결하고자 N-Gram등의 방법 개발되었다.

## Impurity

분리된 데이터 안 이상치가 얼마나 섞여 있는지를 뜻한다. 소수와 그 밖의 수로 나누었을 때 소수무리에서 1,4,6 등의 수가 나오는 경우 impurity 즉 불순도가 높다고 하며 이와 반대로 잘 정렬된 데이터를 purity가 높다고 한다. Impurity를 수치적으로 나타낼 시에 entropy를 사용하는데 엔트로피가 1에 가까울수록 불순도가 높다는 뜻이고 0에 가까울수록 낮다는 뜻이다.

## Decision Tree

Classification과 Regression 모두가능 한 지도학습 모델로서 알고리즘과 유사한 형태를 띈다. 질문 or 조건에 따라 데이터를 나누며 한번의 질문에 데이터가 양분된다. 트리와 마찬가지로 Root Node, Terminal Node, Leaf Node 등의 용어를 사용한다. Over fitting을 막기위하며 pruning을 사용하며 가지치기라는 뜻을 가지고 있다. 모델의 깊이 노드의 수 등에 제한 조건을 거는 방법으로 최적화를 해준다. Entropy 개념을 활용하여 최적의 상황을 알 수 있다. Decision Tree에서 파생된 Random forest 라는 모델이 있다. 적은 수의 feature은 Decision Tree에서 해결 가능하지만 수십 수백개의 feature를 계산하는 경우 overfitting의 발생가능성이 크며 최소한의 학습을 달성하기 위하여(underfitting이 발생하지 않기 위하여) pruning 또한 통하지 않는다. 때문에 여러 Tree를 만들고 ensemble하여 모델을 구축한다. Classification문제일 경우 가장 많은 값 Regression일경우 평균값을 기준으로 모델의 최종 값을 결정한다.

## k-Clustering

근처에 있는 데이터 즉 비슷한 feature를 가진 데이터끼리 묶는 것을 Clustering이라고 한다. 비지도 학습 중 하나이며 k개의 기준점(centroid)로부터 군집을 형성해준다. 초기에 Centroid를 잡는 방법으로 k-mean, random selection 등의 방법이 있다. Centroid가 결정되었을 시에 주변의 데이터들을 확인하며 가장 가까운 Centroid를 확인한 후 클러스터로 할당된다. 위와 같은 방식으로 인하여 k-mean으로 Centroid을 정할 시에 local minimum이 발생할 수 있다.

## Principal Components Analysis

다차원의 데이터일수록 모델의 loss가 증가하고 overfitting이 발생할 가능성이 커진다. PCA는 고차원의 데이터를 저차원의 데이터로 축소시켜 위와 같은 문제를 해결하는 방법 중 하나이다. 많은 feature을 기반으로 모델을 학습하는 경우에 모든 feature가 전부 같은 중요도를 지니고 있지는 않다. 이러한 특징을 이용하여 중요도를 기준으로 하여 몇몇feature만 사용하여 모델을 학습시키는 것이 PCA이다. PCA는 모델의 성능향상 뿐만 아니라 시각화(4차원이상의 데이터는 인간의 시각으로 쉽게 인식 불가능하다.), noise remove(결과 값과 연관 없는 feature를 제거함으로써 노이즈를 줄일 수 있다.), 복잡도 개선(feature수가 줄어듦에 따라 연산과정과 그에 따른 memory소요가 줄어든다.) 등을 위하여 사용된다. PCA의 동작원리를 알아보자 데이터의 분포를 통하여 기준점을 설정하고 원점을 지나는 선과 데이터 분포사이의 거리를 Sum of Squared를 통하여 구한후에 선의 위치를 최적화해준다. 만들어진 선에 대해서 기존의 축을 담으면 차원축소가 완료된다.

## Deep learning

X->y로의 과정에 있는 f(w, x)에 대한 값을 찾는 문제이다. 이때 함수의 구조와 복잡도를 정하는 것이 모델링이며 어떤 Architecture를 사용할 것인지 activation함수는 무엇을 사용할 것인지 등을 정한다. 모델링이 완료가 되면 가중치 w에 대한 업데이트가 진행이 된다. 업데이트란 목표로 하는 값에 다가갈 수 있도록 w값을 조정하는 것을 의미한다. 우리는 이렇게 최적의w에 대한 값을 찾아 감으로써 모델을 학습해가는데 최적의 w값을 찾기 위하여 loss function을 사용한다. 입력 x에 대해서 추정 값y와 실제 관측 값 y(2) 사이의 오차를 residual이라고 하며 residual을 활용하여 loss function을 정의한다. Residual의 제곱 혹은 절댓값을 씌울 수도 있으며 이 밖에도 다양한 활용이 가능하다. Loss function에따라 parameter가 수렴하지 못할 수도 굴곡이 심해 쉽게 발산할 수 있다. 즉 관측데이터에 따른 목표설정을 자유롭게 해주는 하나의 기능이라고 할 수 있다. Ranking loss (image retrieval), yolo (weight least squares loss), classification (cress entropy loss, balanced cross entropy loss).

## Cross validation

모델을 평가하는 방법 중에 하나로서 기존의 train data를 사용하여 훈련하고 test data로 평가하는 방식이 아닌 train data를 잘게 쪼개어 그중 하나를 test data로 만든다 이과정을 쪼개어진 횟수만큼 반복하여 모델을 평가하는 방식이다. 좀더 자세히 알아보면 k개의 데이터로 분류를 한 후 하나는 test data 나머지는 train data 로 활용한다. 위 과정을 k번 시행하며 각각의 성능의 평균을 최종성능으로 한다. 위와 같은 방식을 k-fold-validation이라고 한다. 데이터가 불균형한 경우를 대비하여 stratified k-fold 방식을 사용하기도 한다. Feature에 대해서 균등한 분배를 위하여 각각의 train set과 test set에 균형 있게 분배하는 방식이다. 예를 들어 사기에 대한 data 10개 그 밖의 정상 처리 데이터 90개를 k=5일 때 분배한다면 test data =(사기 \*2, 정상 \*18)과 같은 방식으로 분배가 된다.

## Boosting

Ensemble중 하나로 여러 모델의 연결을 통하여 학습을 진행한다. 첫 모델의 예측 결과를 기반으로 하여 다음 모델의 가중치를 업데이트 해주는 방식으로 진행되며 학습을 병렬적으로 진행하여 최종적으로 투표를 받는 bagging과는 달리 최종학습을 마친 모델의 결과를 최종결과로 한다. Bagging에 비하여 뛰어난 성능을 보이지만 이와 비례하여 overfitting이 발생할 가능성 또한 높아진다. Ada boost, gradient boost, NG boost 등의 방법이 존재하며 그중 NG boost는 확률적 예측을 가능하게 해준다는 특징을 가지고 있다.

* 나중에 공부할 것들

## Bootstrap

표본에서 표본을 복원 추출하고 각표본에 대한 통계량을 다시 계산하는 것으로 위와 같은 과정을 bootstrap라고 한다. N개의 표본 중 하나를 복원 추출하고 이를 k번 반복하며 k번 반복한 값들의 평균을 구한다 위과정을 또 R번 반복하여 bootstrap에 대한 신뢰도를 구한다. parameter에 대하여 정확도를 추정할 수 있으며 이를 기반으로 평균, 표준편차와 같은 총량 값을 추측할 수 있지만 parameter개별 값은 알 수 없다.

## Filter

필터는 CNN의 가중치로 이미지의 특징을 찾아 내기 위하여 고안된 방식이다. 다른 말로 kernel이라고도 하며 일반적으로 정 사각 행렬의 형태로 정의된다. 필터의 이동크기를 Stride라고 하며 필터는 이미지를 Stride 값만큼 이동시키며 Convolution을 진행한다. 입력데이터가 Channel을 같는 경우 channel수에 관계없이 각Channel의 convolution값을 통해 feature map을 합성하여 최종 값으로 한다. 여기서 Channel이란 기본적으로 RGB에대한 값이며 하나의 이미지가 컬러라면 이는 Red, Green, Blue Channel로 이루어져 있다. 이런 특성 때문에 depth라고도 한다. Filter을 사용하면 입력데이터의 값에 비해 Feature map가 작아지는데 이를 방지하기위해 padding을한다. padding이란 입력데이터 외부의 값을 특정한 수로 채우는 것을 말하며 보통 CNN에서는 0으로 채우는 Zero padding을 사용한다.

## CNN

CNN은 입력데이터를 1차원으로 한정하는 FCNN과 달리 데이터차원을 유지시켜 차원 축소로부터 이어지는 데이터의 손실을 예방할 수 있다. 복수의 filter를 사용하여 이미지의 feature을 추출하고 학습하는 능력이 뛰어나며 filter의 공유로 FCNN보다 필요 parameter의 수가 압도적으로 적은 모습을 보여준다. CNN에 대해 알아 보기위해 convolution에 대해 먼저 알아보자. 이산변수에 대하여 convolution은 변수 X, W에 대하여 하나의 변수를 Y축에 대칭 시키고 n만큼 이동 시킨 후에 곱을 하하는 것 의미한다. CNN은 cross-correlation을 사용하여 대칭이동에 대한 과정을 없애고 n에대한 이동만 남기는 것이 보통이다. CNN의 구조를 보면 Convolution 연산을 시행한후에 모델 중간중간에 있는 polling이라는 과정을 통하는데 Multi-layer perceptron 이전에 데이터를 1차원데이터의 형태로 변형시켜야 하기 때문이다. pooling 방법에는 Max-pooling, Average-pooling, Min-pooling 이 있다. 하나하나 살펴보면 Max-pooling의 경우 행렬의 특정영역안에서의 값 중 가장 큰 값을 기준으로 하여 결과값을 낸다. Average, Min-pooling의 경우도 같은 방식으로 적당한 기준으로 정 사각 행렬을 나누어 나누어진 부분에서 각각 평균값, 최솟값을 도출해낸다. 일반적으로 CNN에서는 Max-pooling를 사용하며 pooling은 stride와 같은 크기로 설정이 된다. Convolution과 Max-polling을 반복적으로 진행하며 Feature Extraction하는 과정이 굉장히 중요하며 이를 위하여 Filter, Stride, padding을 잘 조절해 주어야한다.

## Drop out

Hidden layer가 많아질 경우 학습능력이 올라간 만큼 overfitting에 빠지게 쉽다. 또한 학습시간이 지나치게 길어 지기도 한다. 이럴 때 사용하는 것이 Drop out이며 말그대로 Hidden layer의 node를 떨어뜨리는 것이다. Drop out을 통하여 voting 효과를 기대할 수 있는데 일정부분에 대하여 overfitting시켜 평균값을 얻어내는 것으로 regularization 효과를 얻을 수 있다. 둘째로 Co-adaptation을 피할 수 있다. 특정 Node에 대하여 bias나 weight값이 지나치게 클 경우 이 값이 전파되어 최종적으로 나오는 결과에 대해서 악영향을 준다. Drop out을 사용하는 경우에 bias와 weight에 대하여 전파력이 낮아 결론적으로 강력한 모델을 생성 할 수 있다.

# GAN

정말 개판으로 짠 대본 및 정리 본

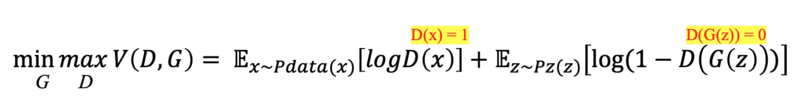
## Gan(Generative Adversarial Network)

Gan은 적대적 생성 모델로서 cv쪽에서는 이미지 생성을 목적으로 하며 데이터가 부족할경우

GAN에 대해서 가장 쉽게 이해하는 문장이 바로 경찰과 도둑입니다. 도둑은 위조지폐를 만들어 경찰을 속이려 노력하며 경찰은 위조지폐를 감별하기 위하여 노력합니다. 지폐를 만들고 감별함에 따라 감별기술, 위조기술의 능력치가 상승할 것이며 이는 인공지능 모델에서의 학습을 의미하죠. GAN에서 경찰은 Discriminator 도둑은 Generator라고 불리며 각각 D(X), D(G(Z)) 로 표현됩니다. 여기에서 Z는 노이즈를 의미하며 GAN은 확률공간 상의 노이즈를 뽑아 이미지를 생성합니다. 수식에 대해 살펴보기에 앞서 증명에 사용하는 개념에 대하여 가볍게 살펴보겠습니다. 먼저 엔트로피입니다. 어떤 상태에서의 불확실성,평균 정보량을 의미하며 확률이 낮고 불확실한 정보일수록 엔트로피가 높다고 표현합니다. 다음으로 kl divergence입니다. 쿨 백 라이블러 발산은 두 확률 분포 사이의 차이를 계산할 때 사용하는 함수로서 정보 엔트로피를 이용해 비교를 진행 합니다. 다음으로는 JSD(Jensen shannon divergence) 입니다 kLD의 경우 Symmetric하지 않다는 단점이 존대합니다. JSD는 아래 수식을 보시면 알겠지만 JSD의 경우 Symmetric며 두 확률 분포 사이의 distance 를 나타내 줍니다.

이세상의 모든 데이터는 다차원 특징공간의 한점으로 표현됩니다 만들고자 하는 이미지에는 통계적인 평균이 존재하며 모델은 이를 수치적으로 표현하는데요 생성모델 G의 목표는 이미지 데이터의 분포를 근사하는 것입니다. 생성 모델이 학습되어 감에 따라 원본 이미지의 분포구조와 유사해지 지게 되죠. 조금 더 자세히 알아보면 아래 그림에서 파란 점선은 discriminative function 검은 점선은 real data distribution 초록색선은 generative function 을 의미합니다. 노이즈 z에서 x로의 mapping 이 G(z)이며 초기에 학습되지 않은 경우를 볼때는 생성자와 원본이미지의 분포에 큰차이가 있지만 학습을 진행함에 따라 두분포사이의 차이가 줄어들고 판별모델의 성능 또한 1/2에 수렴합니다.

이제 GAN의 수식을 살펴보겠습니다.



V(D,G)는 GAN의 loss함수로 Ex-pdata(x)는 임의의 원본데이터 중 일부를 샘플링한 값을 의미하며 E(z-pz(z))는 노이즈 값의 추출을 의미합니다. D, 구별자는 위와 같이 D(G(z))=0과 D(x)=1을 목표로 하며 D에대한 미분으로 ascending하게 됩니다. 반면에 G는 D(x)의 결과에 관계없이 오른쪽 식인 1-D(G(z))를 최소화하는 것을 목표로 하며 G에 대한 미분을 통해 ascending합니다.



## Global optimality

다음으로 생성자와 판별자가 수렴하는 곳을 찾기위해 유도과정을 살펴보겠습니다. 먼저 D(x)의 경우 G값이 고정되었다는 가정하에 아래의 식에서 최적의 값을 가집니다. 유도해보면 기본V(D,G)에서 적분형태로 식을 변형시킨후에 P(X)와 D(X)를 치환하고 정리하여 유도합니다. 앞에서 살펴 보았듯 모델은 P g=p data을 따라가며 이때 global optimum이 정의가 됩니다. -log4 를 취해 줌으로서 Kl divergence를 사용하고 이를 다시 jsd로 정리함으로서 나타냅니다 kl과 jsd를 사용하는 이유는 단순히 증명의 편의 성을 위하여 사용하며 kld의 경우 거리를 구함에 있어 사용하기 까다롭습니다. 때문에 jsd를 사용하여 정리해주는 것이지요. 결과적으로 위와 같은 과정을 통하여 global optima 값은 -log 4을 얻을 수 있습니다.

GAN의 문제점

Vanilla GAN에는 몇가지 문제가 존재하는데 첫번째로 gradient기반의 학습과정에서 문제가 생긴다. 두모델들 D,G는동시에 협력하지 않아 균등한 상태유지가 힘들다. 간단하게 생각해 보면 알 수있다. D에 대한 업데이트와 G에대한 업데이트를 동시에 진행할 경우 아래와 같이 분안정한 진동을 낳게된다. 둘째로 고차원 공간상의 자유로움을 누리지 못한다. 데이터셋은 여러 특징이 분포하며 이는 고차원 공간으로의 확장을 의미하게 된다. 하지만 D와 G의 경우 저차원 공간의 manifold에 놓여지게 되며 이는 두 분포를 구별하는 완벽한 식별자가 존재함을 의미한다. 마지막으로 식별자가 극단적인 값을 가지게 되는 경우이다. 완벽한 식별자를 가진다면 비용함수의 gradient는 0에 수렴하여 학습은 매우 느려지거나 멈추게 되며 식별자의 성능이 나쁠경우 생성자는 정확한 피드백을 하지 않고 비용함수는 현실을 반영하지 못한다는 문제가 있습니다. 이 밖에도 GAN의 모델 평가가 어렵다는 점 지나치게 학습을 하는 경우가 자주 발생한다는 점이 있습니다.

## DCGAN

고해상도 이미지를 제작하는 깊은 차원의 모델을 만들고자 고안된 아키텍쳐로 GAN의 불완전한 부분을 보완한 모델이다. 이름에서 알 수 있듯이 convolution 구조를 GAN에 합한 구조로 상당히 강력한 성능을 뽐냅니다. DCGAN의 특징으로는 poolinglayers 대신에 stride convolution &fractional strided convolutions 을사용 , conv features 위의 FC layers 를 제거하고 Global average pooling으로 모델 안정성을 향상시킴. BN을 적용시킴 ,Relu를 사용함 등이 있다. 대부분의 상황에서 학습이 되는 강견함과 특정파트를 학습한 사실을 보여 줄 수 있는점에서 굉장한 관심을 끌었던 모델입니다.Generator의 collapse를 방지하기 위하여 image de duplication과정을 진행했으며 이를통해 precision은 높아지고 FP비율이 0.01이하로 떨어지는 결과를 내보였다.latent 에서 이동을 살펴보면 급작스럽지 않고 부드럽게 변화하며 이미지 생성에 대해 의미있는 변화 객체를 추가하거나 제거하는 등의 변화를 줄 수 있다. 아래 사진을 보면 TV가 생기기도 하고 커튼이 생기기도 하는등의 변화를 확인 할 수 있다. 또한 블랙박스라고 비판받던 점을 backpropagation을 시각화 하여 해결해 주었고 이를 통해 D의 인식과정을 알 수 있었다. DC GAN의 경우 GAN의 불완정성이 보완되었지만 여전히 oscliiating ,mode collapse등의 문제다 발생할 수 있다. DC GAN을 이용한 결과를 살펴보면 아래와 같이 어느정도 그럴듯한 이미지를 내보내고 있다. 포켓몬 사진을 입입력으로 주었을 때 어느정도 그럴듯한 이미지를 보여주죠 물론 위 모델에서 epoch가 일정수준일 넘어갈 때 흰색화면만 보여주는 문제점이 있어 조정이 필요합니다.

## CGAN

기존의 GAN에 Condition이 추가된 형태로 MNIST 클라스 라벨 조건에 맞춰 생성할 수 있다. CGAN의 loss함수는 다음과 같다. 기존의 GAN에 추가적인 정보 Y를 집어 넣는 것으로 표현이 되는데 이때 y는 보조적인 정부일수도 있고 클래스라벨이나 다른 유형으로부터의 데이터 또한 가능하다. G에서 노이즈와 y를 합동은닉하며 GNA 플레임워크는 이러한 은닉표현에 상당히 자유롭다.

minGmaxDV(G,D)=𝔼x∼pdata(x)[logD(x|y)]+𝔼z∼pz(z)[log(1−D(G(z|y))]

아래는 MNIST 이미지들에 대해 학습한 결과입니다.

## LS GAN

기존의 GAN이 가지고 있던 문제점중 하나가 바로 실제이미지에 비해 이미지의 질이 떨어지는 질적문제이다. 원인으로는 loss function으로 sigmoid를 사용했다는 점이며 LSGAN에서는 이러한 점을 해결하고자 loss function에 least square를 적용시켰다. 이로 인하여 가짜 샘플들을 결정경계에 가깝게 이동시켜 주었다. Ls GAN의 공식을 살펴보면 아래와 같은 형태이다.

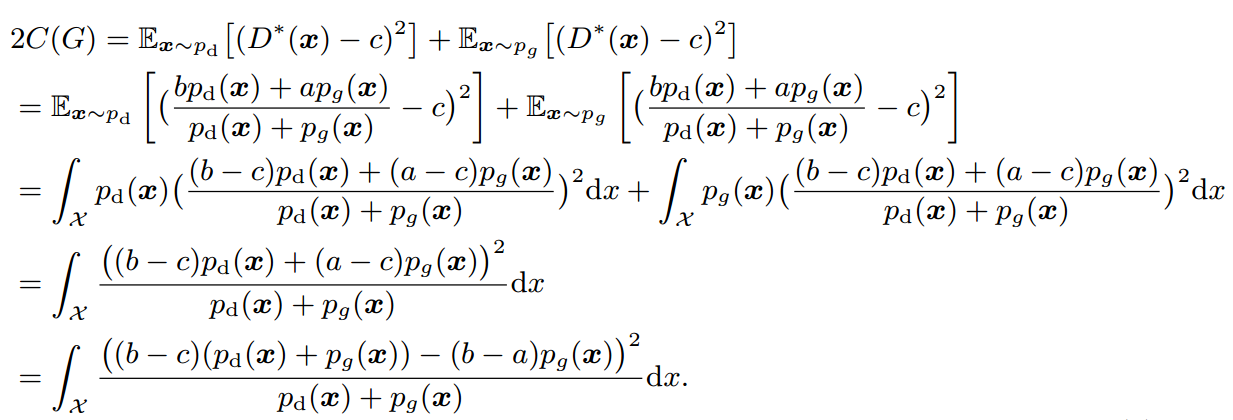
minDVLSGAN(D)=1/2𝔼x∼pdata(x)[log(D(x)−b)2]+1/2𝔼z∼pz(z)[log(D(G(z))−a)2]

minGVLSGAN(G)=1/2𝔼z∼pz(z)[log(D(G(z))−c)2]

기존의 GAN에서 조금 변형된 식을 가지는데 a는 가자 데이터 라벨 b는 실제데이터 라벨 c는 G입장에서 D가 가짜 데이터를 보고 진짜라고 믿기를 원하는 값이다. Gan과 마찬가지로 optima를 구해보면 아래와 같은 과정을 거치게 됩니다. 수식의 유도과정의 경우 Vanilla GAN과 유사한 느낌으로 이후에 한번 읽어 보시면 이해가 되실 것 입니다.  
minDVLSGAN(D)=12Ex∼pdata (x)[(D(x)−b)\*\*2]+12Ez∼pz(z)[(D(G(z))−a)\*\*2

minGVLSGAN(G)=12Ex∼pdata(x)[(D(x)−c)\*\*2]+12Ez∼pz(z)[(D(G(z))−c)\*\*2]

D∗(x)=(bpdata(x)+apg(x))/(pdata+pg(x))



실험으로는 Gaussian Mixture distribution dataset에 대한 것 이다. 위실험에서 Gaussian kernel density 추정의 결과를 보여준다. Regular GAN이 step 15K부터 mode collapse에 빠지는 것을 볼 수 있다. 반면에 LSGAN은 성공적으로 Gaussian mixture distribution은 학습했다.

위와 같은 실험을 기반으로 LS GAN이 GAN보다 강력한 성능을 가지고 있으며 다른 모델에 비하여 안정성이 굉장이 높음을 알 수 있습니다.

## WGAN

Wasserstrin Gan으로 기존의 비용함수를 Wasserstein Distance로 설정하여 최적화를 진행하는 신경망을 말한다. 기존의 gan과 다르게 비용함수를 min max문제에서 max문제로 변환됩니다. Fw라는 특정한 함수가 만들어진다. 가장 핵심적인 waserstrin Distance(earth movers distance)에 대하여 알아보자 확률분포 p\_r에서 p\_g로의 질량이동을 통해 거리를 알아내는 방법으로 질량을 이동하는 방법인 결합확률 분포세트중 가장 작은 값을 선택한다. 이러한 방법은 결합확률 분포의 수가 많아 질수록 모델이 계산하기 어려워진다. 따라서 우리는 이러한 문제점을 해결하기위해 kantorovixh Rubinstein duality를 사용하여 max문제로 치환을 하게된다. 새로운 wasserstein방법은 항상 k lipschitz continuous를 만족해야하며 식별자는 실제 샘플과 가짜 샘플을 구별하는 지표가 아니라 k Lipschitz continuous를 학습하게 됩니다. 모델의 발전가능성이지만 제약이기도한 k Lipschitz contunity는 논문에서 제시한 트릭으로 어느정도 보완이된다. 가중치를 -0.01-0.01사이로 고정시켜 Lipschitz continuty를 유지하는것이다.

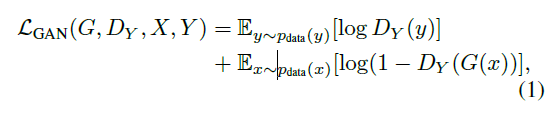
## WGAN-GP

WGAN에서 critic 의 weight clipping이 발생시키는 문제 undesired behavior를 해결하기위해 고안된 방법이다 . 최적화에 문제가 발생할 수있으며 최적화가 잘되더라도 critic가 상상도 못할값을 가질 수도 있다. 때문에 Gradient penalty를 사용하여 해결하는데 아래 실험결과를 보면 weight clipping와 gradient 의 안정성을 비교해 보았을 때 weight clipping을 사용 하였을 시에는 vanish explode 등의 문제가 발생하며 gradient penalty 방법에서는 문제가 발생하지 않았다. 수식을 보면 아래와 같은 형태이다. 랜덤샘플 x-P\_x에대하여 gradient norm 에대한 패널티와 함계 soft한 버전의 제약을 준다 .weight clipping 보다 학습속도가 빠르고 샘풀의 품질이 좋아지는데 이를 증명하기위해 CIFAR-10을 활용한 실험결과이다.RMSProp를 사용하냐 Adam을 사용하냐에 따라 learning rate를 조정하며 같은 optima라도 W GAN의 수렴속도가 훨씬 빠르며 weight clipping 높은 점수를 보여준다.

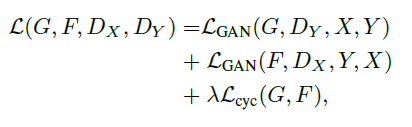
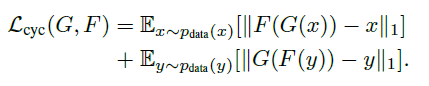
## Cycle Gan

기존의 GAN모델의 경우 각각의 인풋에 대해 의미있는 방식으로 엮여진다고 말할 수 없다. D의 경우 인풋과 관계없는 데이터임에도 불구하고 진짜인지 가짜인지 만을 판단하기 때문에 통과가 가능하다. Translation이 consistent을 가지도록 cycle gan을 만들었다. 한쌍으로 묶이지 않은 데이터를 학습하기 위해 cycle을 사용하는데 말을 얼룩말로 바꾸거나 계절을 바꾸는 등 content는 유지한채로 몇가지 특성을 바꾸는 것을 가능하게 해준다

Loss 함수들을 살펴보면 먼저Adversarial의 경우 아래와 같은 형태를 띈다.



G는 함수를 최소화시키고자하고 D는 함수를 최대화 시키고자 합니다. 다음으로 cycle consistency loss입니다. Mode collapse문제를 예방하기위해 가능한 매핑함수의 공간을 줄이기 위해 다음과 같은 식이 유도됩니다.



다음으로 L\_identity입니다. 추가적인 loss요소로서 색상 구성을 보존해야 할 때 주로 사용되며 생성자 G에 이미지y가 들어왔을 때 그대로 내보내는 것으로 학습을 노리며 Style이나 다른분야에서의 적용이 가능합니다. 수식은 다음과 같습니다.

****

Cycle GAN에도 불완전한 부분이 존재하는데요 그중 대표적인 것이 바로 말과 얼룩말의 전환파트입니다. 말이미지 트레이닝 데이터에서는 사람이 타고있는 데이터가 많이 존재하지만 얼룩말 이지지에서는 사람이 타고 있는 사진이 거의 존재하지않아 이미지 셋분포의 특징에서 차이가 나기에 모델의 불완전성이 야기됩니다.

## Style GAN

PG GAN은 Style GAN의 선조가 되는 모델입니다. 간단하게 알아보고 가면 latent vextor z가 모델에 바로 입력이 되며 Gan은 latent space의 분포를 그대로 따라가게 됩니다. 또한 PG GAN은 시간이 지남에따라 고해상도 이미지를 만들어 내지만 그학습시간이 지나치게 길다는 단점이 존재합니다. 또한 특정 style 조절이 어렵다는 단점이 존재 했는데요 이러한 단점을 해결한 것이 바로 Style GAN입니다. Style Gan에 대해 살펴보기전 disentangle에 대해 설명드리겠습니다. Disentangle는 각 스타일들이 잘구분되어있는 상태를 말하며 축에 따른 이동으로 각각의 특징을 조절 예측 할 수 있는 상태를 말합니다.이제 Style GAN의 핵심인 mapping network를 살펴 보겠습니다. 완전하지 않은 데이터 분포 a를 가우시안 디스트리뷰선을 활용하여 구의형태로 뽑아 낼 경우 여러특성이 얽힌 entangle 상태가 됩니다. Style Gan에서는 이러한 문제를 해결하기 위해서 latent vector z를 별도의 과정을 통해 disentangle된 w로 변화시켜 사용합니다. 다음으로는 AdIN layer입니다.Adain layer는 feed forward방식의 학습에 강한성능을 보여주며 style과 content에 대한 정보를 VGG encoder를 통해 추출할 수 있습니다. 식을 보면 바로 이해가 되실텐데요 그냥 단순한 평균과 분산으로 보아도됩니다. 실제 모델에서는 normalizrtion ,shpae조정, style 적용등의 역할을 합니다. 학습데이터 셋에 따라 correlation이 발생할 수 있는데요. 일정 style이 a가등장할 때 또하나의 style b가 공통인자로서 연속적으로 등장할경우 모델이 a,b에 대해 overfitting할 가능성이 높아집니다. 이러한 문제를 해결하기 위해 style mixing방법을 사용하는데요 w에대하여 비율을 적용시켜 다양한 style synthesis network 학습을 진행합니다. 아래 표가 그 학습의 결과입니다. Style Gan 1에서는 특정 style을 조정할때에 추가적인 noise 가 섞여 들어가는 것을 볼 수 있습니다.

# Linear Algebra

## Linear

선형대수학은 선형함수에 대한 대수학으로 Linear에 대한 정의는 다음과 같다,

1. L(cx)=cL(x)

2. L(x+y)=L(x)+L(y)

위 두조건을 만족할 시에 선형성을 가진다고 하며 regression 에서는 조금 다른 정의를 가지는데 이는 다음과 같다. 행렬 a와 p의 결합으로 벡터 q가 만들어진다고 했을때 벡터 p가 Linear하다면 벡터q는 Linear하다. 예를 들어보면 이차함수 ax^2+bx+c에서 우리에게 x는 중요하지 않다. 기본적으로 행렬의 형태를 띄는 input값이며 regression에서 중요한 것은 a,b,c의 값이다. ax^2+bx+c을 행렬과 벡터의 관점에서 바라보면 행렬a [x1^2,x2,1;x2^2,x2,1;……]와 벡터 [a,b,c]의 결합으로 만들어진 식이다. 앞서 말했듯이 행렬 a의 상태는 우리의 관심사가 아니다. 벡터의 형태가 linear하다면 ax^2+bx+c또한 linear하다. 조금 더 이어서 생각해 보면 abx+c는 linear은 선형일까? 그래프의 형태로 보았을 때는 틀림없이 직선의 형태를 띈다. 하지만 행렬과 벡터의 결합관점에서 해석해보면 ab는 linear을 만족하지 않으므로 abx+c또한 linear하지 않는다.위와 같이 기하학적인 선형과 regression에서 말하는 선형의 의미는 다르다. 일반적으로 3차함수, 4차함수는 선형적이지 않다 말하지만 위의 정의를 바탕으로 해석했을 때는 linear하다. 같은 맥학으로 a^2x와 abx와 같은 형태의 식은 그래프상으로 linear하지만 regression에서는 무의미하다.

가산성과 동질성에 대해 조금 더 살펴보자 1번식을 만족할 때 동질성,Homogeneity을 만족한다고 하며 2번식을 만족할 때 가산성 ,Additivity하다고 한다. 선형대수학에서 정의하는 선형성을 보면 regression과 같이 기존의 linear 함수라고 생각했던 것들이 linear하지 않다는 것을 깨닫게 된다. f(x)=ax+b라고 해보자 1번의 정리에 대입하면 f(tx)=tax+b=tf(x)이며 tf(x)=tax+bt이므로 Homogeneity를 만족하지 못한다. 왜 이렇게 정의되었을까?(사실 잘 모르겠다) Homogeneity와 additivity를 만족할 때 즉 linear할 때 우리는 input에 대한 outpt의 예측을 할 수 있게 된다. 임의의 input과 output값에 대해 알고있을 시에 input data에 곱과 합 연산을 적용하여 만들 수 있는 데이터를 집어 넣는다면 우리는 output에 해당하는 데이터 값을 추측할 수 있게 된다.

# Algebraic Structure

## 군(Group)

집합G에 대하여 결합법칙(Associative law), 항등원(Existence of neutral element), 역원(Existence of inverse element), 닫힘(Closure) 4가지 연산조건을 만족할 때 군이라고 한다. 하나하나 살펴보자 (편의상+기호를 사용하지만 이는 새로 정의한 기호이며 기존의 더하기와는 다른 이항기호이다.)

결합법칙(Associative law)

결합법칙(Associative law)의 경우 우리가 알고 있는것과 같이 (a+b)+c=a+(b+c)가 만족하는 경우를 의미한다.

항등원(Existence of neutral element)

항등원(Existence of neutral element)의경우 G의 임의의 원소a와 연산+를 수행하였을 때 a+k=a,k+a=a를 만족하는 k가 존재함을 의미하며 이때의 k가 집합 G의 항등원이다. 예를 하나 들어보면 실수공간을 집합으로 하며 \*연산을 정의했을 때 만들어지는 군 A의 항등원은 무엇일까? 정의를 따라 생각해보자 실수공간에서 임의의 원소 R을 뽑았을 때 무엇을 곱하더라도 자기자신이 나오는 수는 무엇일까? R\*1=R이므로 1이 바로 A의 항등원이된다.

## 역원(Exsitence of inverse element)

항등원에 역이되는 개념으로서 G의 임의의 원소 a에 대한 항등원이 정의 되었을 때 원소 a와 다른 원소 a-에 대한 연산결과가 항등원k가 되게하는 원소 a-를 의미한다. 식으로 표현을 해보았을 때 a+a-=k를 만족할 때 역원이 존재한다고 한다. 마찬가지로 예를 들어보자 실수공간을 집합으로 하며 연산\*를 통해 군 B를 만들었을 때 임의의 실수 t에대해 t\*t-=1이 되게하는 t-=1/t이며 군 B는 역원이 존재한다. 만약 집합을 실수공간이 아닌 정수체계로 택했을 경우 군B는 역원을 가지지 못하게 된다. 정수체계에서 1/t를 만족하는 원소는 존재하지 않기 때문이며 집합의 특성과 연산에 대한 정확한 파악이 중요하다.

## 닫힘(Closure)

집합 G의 임의의 원소를 연산했을 때 나오는 결과 또한 집합G에 속해야 한다는 뜻이며 G의 원소 a,b를 연산+ 했을 때 즉 a+b의 결과 또한 집합 G에 속해야 한다는 뜻이다. 예를 들어보면 실수공간에서의 집합에서 뽑은 두개의 원소를 연산+했을 때의 결과물은 실수공간을 벗어나서는 안된다는 뜻이다. 정수공간에서 뽑음 두개의 원소 p와q를 서로 나누어 보았을 때를 생각해보자 p/q가 정수일 때 즉 정수공간에서 정수공간으로 움직일 때는 닫힘의 성질을 만족하지만 p/q가 실수의 체계위로 움직일 때는 닫힘의 성질을 만족하지 않는다.

군의 종류

아벨군(abelian group): 군의 네가지 특성(결합,항등,역원,닫힘)을 만족하며 추가적으로 교환법칙이 성립할 경우 아벨군이라 칭한다. 기존의 +연산과 실수체계에 대한 집합을 군으로 하는 G에 대해 교환법칙이 성립하므로 아벨군이라 칭할 수 있다.

순환군(Cyclic group); 한원소로 군의 모든 원소를 표시할 수 있는 군을 의미하며, 이때의 해당원소를 생성원(generator)이라한다.(솔직히 처음 듣는다.)

부분군(sub Group): 군 G의 부분집합으로 군G와 같은 연산구조를 가진다.

가군(module Group): 어떤 환의 원소에 대한 곱셈이 주어지며 분배법칙이 존재하는 아벨군이다. 환의 한원소와 아벨군의 곱셈연산의 결합으로 만들어지는 대수구조이며 벡터공간 또한 가군이다.(벡터공간은 체에서 원소를 가져오며 환은 체의 하위구조이다 따라서 벡터공간은 가군이다.)

## 환(Ring)

어떤 집합 R에 대하여 결합법칙이 존재하는 두개의 이항연산 +,\*이 정의 되어있고 이하 조건을 만족할 때 환이라고 한다.군과는 다르게 2가지의 연산을 포함하며 채의 하위구조이다. 덧셈에 대하여 아벨군 곱셈에 대하여 반군을 이루고 분배법칙이 성립하는 대수구조이며 조금 더 자세히 살펴보면 다음과 같다.

덧셈에 대해 아벨군을 만족한다.(군의 4가지 특징에 더하여 교환법칙이 성립한다.)

임의의 두원소의 연산을 통해 나온 결과값은 집합G로 환원된다.

1. 모든원소에 대하여 교환법칙이 성립 2. 모든원소에 대하여 각각의 항등원이 존재

3. 모든원소에 대하여 각각의 역원이 존재

곱셈에 대하여 이하조건을 만족한다.

1. 임의의 두원소의 연산을 통해 나온 결과값은 집합G로 환원된다. 2. 모든원소에 대해 결합법칙이 성립한다.

덧셈과 곱셈에 대해

모든 원소에 대하여 분배법칙이 성립한다.

환의 종류

가환환; 곱셈이 교환법칙을 만족하는 환

나눗셈환: 0이 아닌 모든 원소가 역원을 가지며 원소의 개수가 둘이상인 환(0의 역원은 가질 수 없다.)(추가적으로 항등원의 성질 또한 만족하며 원소의 개수가 1개일경우 의미가없다.)

## 체(field)

환의 상위구조로서 특정한 성질을 만족하며 2개의 연산이 정의된 가환환이며 나눗셈환이다.(환의 상위개념이기 때문에 당연한 이야기이다.) 사칙연산이 자유로이 시행될 수 있는 대수구조이며 산술의 잘 알려진 규칙들을 만족한다. 유리수의 집합 Q, 실수의 집합R, 복소수의 집합 C가 체에 속한다. 정수의 집합 Z에 대해서는 만족하지 못하는데 이유는 위에서 언급했듯이 나누기 연산을 실행할 때에 정수체계 밖으로 넘어가기 때문이다. 다시한번 정리해보면 +와\*에 대하여 결합법칙 교환법칙 항등원 역원 4가지 성질을 모두 만족하는 대수구조를 체라한다.(\*연산에서 역원성질은 특별하게 0은 제외된다)

## 선형생성

선형결합은 벡터들의 스칼라배와 덧셈을 통해 새로운 벡터를 만드는 연산으로 벡터공간에서 가장 기본적인 연산이다. 벡터공간 V의 공집합이 아닌 부분집합 S={v1,v2,v3,….vn} 내의 벡터들의 가능한 모든 선형결합으로 이루어진 V의 부분벡터공간을 S의 선형생성 span(S)이라한다. 예를 들어 S={(1,0),(0,1)}에 대하여 span(s)={k(1,0),m(0,1)}=R^2이다.

## 차원

벡터공간 V의 부분집합 B가 선형 독립이고 V를 생성할 때 B를 V의 기저라 한다. 이때 B의 원소의 개수를 V의 차원 dim(V)라고한다. 기저에는 정규기저,직교기저가 존재하며 이둘을 합친 정규기저가 존재한다. 노름공간 V의 기저B를 정규기저라 하며 내적공간 V의 기저 B를 직교기저라 한다.

## PCA(Principal Component Analysis)

PCA는 n차원에 분포된 데이터들을 축소된 차원에 대입한다. 예를 들어 2차원 좌표평면 위에 데이터 (x,y),(x1,y1),(x2,y2),(x3,y3),…(xn,yn)이 존재 할 때 이 데이터 들의 분포를 데이터들의 분산이 가장 큰 방향벡터로서 분석해 주는 것이다. 영상인식분야에서 활용을 살펴보면 대표적인 예시로 eigenface가 있다. 45\*40의 얼굴 이미지들을 1800차원의 벡터로 치환하고 이에 대하여 PCA를 수행하면 데이터의 차원의 수와 동일한 수의 주성분 벡터를 얻을 수 있다.



위 방법으로 얻어진 주성분벡터를 이미지로 해석한 것이 eigenface라고 하며 분산이 큰 방향벡터 순으로 나열해 보았을 때 위와 같은 양상을 띈다. 초기의 이미지에서는 얼굴전체의 특징을 나타내지만 후기의 이미지일 경우 세부적인 특성을 나타내며 극 후반으로 갈 경우 noise에 가까운 정보를 나타낸다. noise에 대해 조금 더 알아보자 PCA는 데이터의 집합을 n개의 주성분 벡터로서 표현하는 방법이다.eigenface의 앞부분에서는 얼굴의 전체적인 특징 즉 얼굴의 공통적인 파트를 담당하며 후반으로 갈수록 세부적인 파트 사람과 사람을 구분할 수 있는 특징에 대해서 반영하게 된다. PCA를 통해 얻어진 주성분 벡터들은 서로 수직인 관계를 띈다(PCA는 2차원 데이터에서 2개의 서로 수직인 주성분 벡터를 3차원 데이터에서는 3개의 서로 수직인 주성분 벡터들을 반환한다.) 이 말은 주성분 벡터들이 n차원 공간을 생성하는 기저 역할을 할 수 있다는 걸 의미한다.어떤 데이터의 주성분벡터들의 일차결합으로 표현하는 것을 karhunen-loeve transform(KLT) 또는 Hotelling transform이라한다. PCA란 입력데이터들의 공분산 행렬에 대한 고유값분해로 볼 수 있다. 고유벡터가 주성분 벡터로서 데이터의 분포에서 분산이 큰방향을 나타내며 대응되는 고유값이 그 분산의 크기를 나타낸다.

# Natural Language Processing

## Basic NLP

### Tokenization

데이터가 필요에 맞는 형태로 정리되어 있지 않다면 전처리를 통해 데이터를 맞춰주어야 합니다. 토큰의 기준을 단어로 한다면 단어 토큰화라고 부르며 띄어쓰기 혹은 특정한 기준에 따라 단어를 구분합니다. 우리가 원하는 토큰의 형태를 얻기 위해서 직접 코드를 짜서 해결해도 되겠지만 기존에 만들어진 토크나이저를 활용하여 토큰화를 시키는 것 또한 좋다. 토큰화 시에 구두점 혹은 특수문자를 단순 제외해서는 안되며 줄임 말 혹은 단어 내에 띄어쓰기가 있는 경우 또한 주의 해야한다.

### Cleaning & Normalization

토큰화 이전에 모델의 성능을 끌어올리기 위한 전처리 과정으로 토큰화 작업에 방해되는 요소들을 없애고 노이즈를 줄이는 작업이다. 규칙에 기반하여 표기가 다른 단어들을 통합하거나 대소문자 통합, 등장빈도가 적거나 길이가 짧은 단어 제거 등의 역할을 한다. 같은 의미를 가진 다른 단어들을 통합시키는 것만으로도 유의미한 결과를 얻을 수 있다.

### Lemmatization

표제어 추출은 단어들에서 표제어를 뽑아내는 과정으로 is, are, am에서 be를 뽑아내는 과정이라 생각하면 편하다. 단어를 형태학적으로 분해하여 어간, 접사로 구분하여 진행한다. 문맥을 고려하여도 품사의 정보가 보존되지 않을 수 있다.

### Integer Encoding

Model에 집어넣기 앞서 데이터를 수로 변환시켜주는 작업이 필요하다. 이를 위해 텍스트를 수로 바꾸는 여러가지 방법이 존재하는데 그중 Integer Encoding은 단어에 정수를 부여하는 방법으로 빈도수에 따라 vocab을 만들고 빈도수가 높은 순서대로 단어에 인덱스를 부여한다.

### One-Hot-Encoding

원 핫 인코딩은 단어 집합의 크기를 벡터의 차원으로 하고 표현하고 싶은 단어의 인덱스에 1의 값을 부여 다른 인덱스에는 0을 부여하는 표현 방식입니다. 정수 인코딩 후에 표현하고자 하는 단어의 고유한 정수를 인덱스로하여 해당 위치에 1을 부여한다. 단어의 개수 즉 보캡의 사이즈가 커질수록 벡터의 크기 또한 커진다는 단점이 존재한다. 예를 들어 100개의 단어를 표현하고자 한다면 벡터의 크기는 100이되며 n개의 단어를 표현하고자 한다면 벡터의 크기는 n이 된다. 또한 단어의 유사도를 표현하지 못한다는 단점이 존재한다. 단어와 단어 사이의 유사도를 표현하지 않고 단순하게 인덱스를 기반으로 인코딩하기에 부족한 점이 많다.

### N-gram

카운트 기반한 통계적 접근을 사용하는 SLM의 일부로서 일부단어를 고려하여 접근한다. N개의 연속적인 단어 나열을 통해 다음 단어를 통계적으로 접근하며 n=1일 때 유니그램 2일 때 바이그램등 n의 수에 따라 이름이 달라진다. 단어 몇 개를 선택하여 접근하다 보니 전체적인 문맥을 파악하지 못하는 경우가 생기며 n을 정할 때 trade off문제에 직면하는 것 또한 단점이다.

## CNN IN NLP

NN에 Convolution을 적용한 모델로서 필터라는 가중치를 사용하며 pooling, stride, padding등의 개념이 존재한다. 이미지분야에서 강력한 성능을 보여주며 NLP task에서도 사용된다.

Filter: CNN에서의 가중치 CNN은 matrix형태의 가중치를 가지며 input값과 convolution연산을 통해 특징을 뽑아낸다.

Stride: filter를 얼마나 이동시킬지에 대한 파라미터 값 앞뒤 위아래로 움직일 수 있다.

Pooling: convolution된 값들 중 특정 기준을 만족하는 값을 뽑아낸다.

Padding: matrix의 크기조절을 위한 파라미터로 convolution연산후에 줄어드는 matrix 크기를 고정하기위한 파라미터 변수이다.

NLP에서 강력한 성능을 보여주는 모델이지만 dialogue system에서는 발화의 길이가 고정되지 않은 점으로 인해서 높은 성능을 보여주기 어렵다.

## RNN

RNN의 기본개념은 이전 노드의 값을 학습에 사용하여 시퀀스 데이터에 대한 성능을 높이고자 함에 있다. 입력 값이 히든 노드를 통해 weighted Sum되며 output으로 나오는 일반적인 모델들과는 달리 RNN은 순차적으로 연산이 이루어지며 이전의 입력 값으로 입력되어 계산을 마친 값 혹은 히든 노드의 값을 새로운 입력 값 연산 수행시에 사용한다. 히든 노드의 값을 사용할지 결과 값을 사용할지에 따라 Elman-Type 와 Joran-Type RNN으로 나누어 진다. RNN에는 가중치 값에 이상이 생기는 gradient vanishing문제가 존재한다.

### LSTM

기존 RNN모델의 Gradient vanishing문제를 해결하기 위해 고안되었으며 입력 게이트, 삭제 게이트, 셸 상태로 이루어진다. 게이트 메커니즘을 통해 데이터의 시퀀스를 관리하며 그 모습을 현재 메모리와 이전 메모리 게이트 구조를 통해 알 수 있다. 셸 상태를 보면 알 수 있는데 삭제게이트를 통해 시그모이드를 거친 값이 이전 셀값과의 연산을 통해 이후의 셀로 나아간다.

### GRU

RNN 계열의 모델로 LSTM과 달리 update gate, reset gate만을 포함하고 있다. LSTM에 비해 적은 파라미터를 보유하고 있지만 보편적으로 더 안정된 성능을 보여준다.

## Attention is All You Need

Attention은 단어와 단어 사이의 상관관계를 고려하기위한 메커니즘으로 임베딩된 단어벡터에 WQ,WK,WV 행렬을 곱하여 Q,K,V 행렬을 생성한다. W값을 학습하며 더 나은 행렬을 만들기 위해 최적화 된다. Q와 KT의 내적을 차원의 루트값으로 스케일링한후 softmax를 취해주고 Z행렬을 곱하여 의미없는 값들을 날려버리는 것이 attention이다. Attention을 쌓아 만든 것이 Transformer로 이런 Transformer의 인코더 파트와 디코더 파트를 각각 쌓은 모델이 바로 BERT와 GPT이다.

### BERT

Transformer의 Encoder를 쌓은 모델 마스크예측과 다음문장 예측을 사용한다. 마스크 예측은 문장에 확률에 기반한 마스크를 랜덤으로 씌워 마스크 처리된 부분을 예측하는 파트이며 다음 문장 예측은 문장의 순서를 예측하는 학습이다. 토큰 임베딩 세그먼트 임베딩, 위치 임베딩을 사용하며 단어의 빈도수를 기반으로한 확률 알고리즘인 워드피스 알고리즘을 사용한다.

### GPT

GPT는 Transformer 기반의 디코더를 쌓은 모델이다. Byte pair Encoding과 Uni direction이 특징이다. 생성분야에서 기존의 모델들에 비해 뚜렷한 성능 향상을 보여준다.

Byte pair Encoding: 자주 사용하는 단어를 묶어서

Uni direction: Bert와는 다르게 단방향으로 인코딩한다.

# Task Oriented Dialogue System

상호간의 발화 중에 사용자의 목표를 인식하고 달성하기 위한 분야로 삼성의 Bixby나 애플의 SIRI를 예로 들 수 있다. TOD의 pipeline은 다음과 같이 구성되어 있다.

NLU: Natural Language Understanding

* Predefined된 slot에 mapping

DST: Dialogue State Tracking

* Each turn에 발화를 전발화와 결합하여 state를 tracking

DP: Dialogue Policy

* Dialogue state를 바탕으로 action을 결정

NLG: Natural language generation

* Action을 선택하고 generate한다.

## End to End

Credit assignment problem

* 마지막 유저의 feedback이 모듈에 제대로 전달되기 힘들다.

Process Interdependence problem

* 각 컴포넌트의 결과 값이 이후의 컴포넌트의 input으로 들어간다. 예를 들어 NLU의 결과값이 DST로 입력되고 DST의 결과 값이 DP이후로 NLG로 이어진다. 이러한 문제는 새로운 분야를 학습할 때 두드러지는데 각 component의 global optima를 수렴시켜 주어야한다.

## NLU

인간의 발화를 처리하는 단계로 입력된 자연어를 기반으로 slot classification을 수행한다. NLU Model은 3가지 task를 처리한다.

1. Domain Classification

-

2. Intent Detection

-

3. Slot Filling

-

## DST

## DP

DST로부터 도출된 상태에 대한 표현을 통하여 다음 스텝의 시스템 액션을 정해준다. 지도학습과 강화학습이 사용되며 최근 연구들은 대부분 이부분에 강화학습을 사용한다.

## NLG

## TRADE

### Slot gate

Slot gate g는 3가지 slot상태로 분리된다. 각각 don’t care, none, ptr이다. Ptr은 slot에 값이 매핑되었을 때 don’t care은 의미 없는 값일 때 none는 아직 값이 매핑되지 않았을 때를 의미한다.

### Pointer Generator

인코딩된 벡터를 어텐션하여 나온 값과 기존의 vocab의 분포를 weighted sum하여 새로운 예측분포를 만들어 낸다. Copy된 단어에 비하여 vocab에서의 단어값을 통해 p,1-p값을 결정한다.

### Multi turn problem

Predefine된 ontology 없이 parameter sharing을 통하여 학습하지 못한 slog, value pair에서 높은 성능을 끌어올렸다.

### Zero shot DST

학습되지 않은 도메인에 대하여 실험해 보았을 때 몇몇 경우에서 높은 성능을 보였는데 이는 같은 슬롯을 공유했을 때 발생했다. 각 도메인 마다 필요한 슬롯이 보통 다르지만 특정한 몇몇 캐이스에서 겹치는 것을 확인 할 수 있는데 이때 zero shot에서 높은 성능을 보였다.

## Challenge

Deep learning을 DST에 적용한 challenge로 feature function 12개를 사용하여 다이얼로그를 추적한다. Feature function에는 SLU score, Rank score, Affirm score, Negate score 등이 존재하며 SLU(Spoken language Understanding)로부터 이해한 정보를 기반으로 feature function에 입력하여 weighted sum한후 softmax를 취한다. Challenge의 결과로 12개의 feature function을 전부 사용했고 hidden node의 수가 20,10,2일 때 가장 좋은 score를 기록했다.

Dialogue state는 3가지 파트로 구성되어 있다. (Goal constraint, requested slots, search method) goal constraint는 사용자의 목표를 특정하는 것을 의미하며 ontology를 통해 특정할 수 있으며 사용자의 목표가 특정해지지 않을 때 즉 slot에 value가 아직 매핑 되지 않았을 때 None이라는 표현을 사용한다. Slot에 의미 없는 value가 매핑 된다면 don’t care사용

# Reinforcement Learning

## Notation

S state

A action

S set of all nonterminal states

S+ set of all states, including the terminal state

π policy, decision-making rule

π(a) action taken in state s under deterministic policy π

π(a|s) probability of taking action in state s under stochastic policy π

Vπ(s) value of state s under policy π

## MDP

### State

에이전트에게 주어진 상태로 다시 말해 에이전트가 관찰 가능한 상태의 집합이다.

### Action

### Reward

## Q Learning

# Emotion Recognition in Conversation

## Emotion

## Dialogue GCN

## MELD

**Reference**

Reinforcement Learning: An Introduction

Dialogue GCN: A Graph Convolution Neural Network for Emotion Recognition in Conversation

A Survey on Dialogue Systems: Recent Advances and New Frontiers