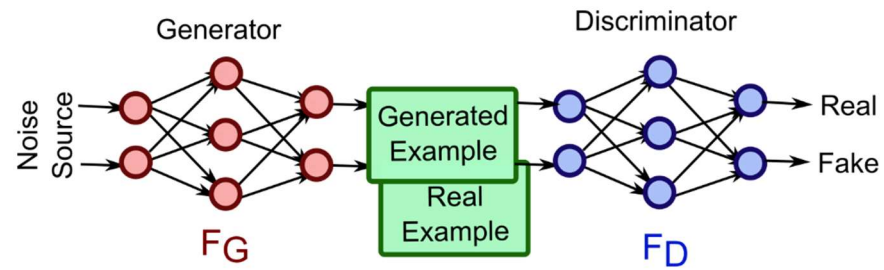
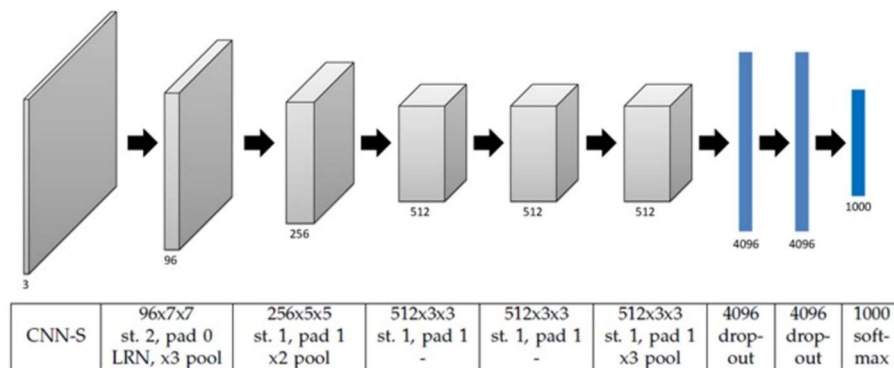


10주차 조별보고서	
작성일: 2019년 11월 08일	작성자: 김영연
조 모임 일시: 2019년 11월 07일	모임장소: 학교 앞 카페
참석자: 김영연, 이재은, 이충현, 위성조, 최진성	조원: 김영연, 이재은, 이충현, 위성조, 최진성
구 분	내 용
학습 범위와 내용	<p>4.5 생성 모델</p> <p>4.6 딥러닝은 왜 강력한가?</p>
논의 내용	<p>Q1</p> <p>GAN의 학습은 생성자와 분류자가 서로 경쟁하며 학습하는 모델입니다. 생성자나 분류자 둘 중 하나가 성능이 특히 뛰어난 경우에도 학습이 잘 되는지 또한 학습이 잘 되지 않는다면 이를 해결할 방법이 무엇인지 궁금합니다.</p>
	<p>A1</p> <p>생성자나 분류자 둘 중 하나가 성능이 특히 뛰어나다면, 학습이 제대로 이루어지지 않는다. GAN의 학습이 잘되기 위해서는 비슷한 수준의 생성자와 분류자가 함께 조금씩 발전해야, 균형 있게 서로 훈련을 주고받기 때문이다. 다시 말해, 두 모델간 실력차가 발생하는 경우에 훈련이 한쪽에 치우쳐 성능이 제약된다.</p> <p>이러한 부분을 해결하기 위해서 나타난 해결책은 바로 DCGAN(Deep Convolutional GAN)이다. DCGAN은 기존 GAN에 존재했</p>

던 fully-connected DNN구조를 CNN 구조로 대체한 것으로, 생성자와 분류자의 세부적인 구조가 달라진다.



[기존 fully-connected로 되어있는 GAN의 구조]



Chatfield et al. *Return of the Devil in the Details: Delving Deep into Convolutional Nets*

[fully-connected 구조 대신에 convolution, pooling, padding을 활용한 CNN의 구조]

GAN의 한계를 극복하기 위해 시도된 여러 방법들을 통한 DCGAN의 구조는 아래와 같다.

- Discriminator 에서는 모든 pooling layers 를 strided convolutions 로 바꾸고, Generator 에서는 pooling layers 를 fractional-strided convolutions 으로 바꾼다.
- Generator 와 Discriminator 에 batch-normalization 을 사용한다. 논문에서는 이를 통해 deep generators 의 초기 실패를 막는다고 하였다. 그러나 모든 layer 에 다 적용하면 sample oscillation 과 model instability 의 문제가 발생하여 Generator output layer 와 Discriminator input layer 에는 적용하지 않았다고 한다.
- Fully-connected hidden layers 를 삭제한다.
- Generator 에서 모든 활성화 함수를 Relu 를 쓰되, 마지막 결과에서만 Tanh 를 사용한다.
- Discriminator 에서는 모든 활성화 함수를 LeakyRelu 를 쓴다.

Q2

GAN의 최종 목적은 분류 모델이 최종적으로 정확한 분류를 하지 못 할 정도의 생성 모델 성능을 내는 것인데, GAN 과정에서 분류 모델의 성능 향상은 없나요? 만약에 분류 모델의 성능 향상이 생성 모델의 성능 향상보다 더 뛰어나다면 사용자가 임의로 중지할 때 까지, 혹은 제한된 횟수가 다 될 때 까지 훈련을 하는 것인가요?

A2

GAN의 경우 분류기와 생성기의 목적함수가 서로 연관되어 있으며, 모델 학습 시, 생성기와 분류기 모두의 그래디언트를 구해 분별기는 그래디언트를 더해 주고(최대화 문제), 생성기는 그래디언트를 빼 주면서(최소화 문제) 학습하므로, GAN은 생성기와 분류기 모두 학습 과정을 거치게 된다. 또한 분류기의 성능이 생성 모델의 성능 향상보다 더 뛰어나더라도, GAN은 분별기의 목적함수 값이 1/2가 될때까지 학습하게 된다.

Q3

생성 모델이 아닌 분류 모델의 성능 향상을 목표로, 생성 모델과 분류 모델을 동시에 가지고 있는 신경망은 없을까요?

A3

분류 모델은 샘플의 카테고리만의 예측하는데 관심이 있는 모델이다. 쉽게 말하면 특정 데이터의 카테고리를 분류하는 모델이다. 사후확률만 구하는데 x 라는 샘플이 있을 때 이 샘플의 카테고리가 y 일 확률, 즉 사후 확률 $P(y|x)$ 만을 추정하면 된다.

$$P(Y|X) \propto P(X|Y)P(Y)$$

생성 모델은 분류 모델과 달리 x 가 발생할 확률인 $P(x)$ 나 카테고리 y 에서 x 가 발생할 확률 $P(x|y)$ 를 명시적으로 계산한다. 베이지 공식을 적용하여 $P(y|x)$ 를 구해 분류 모델 역할을 수행할 수도 있다.

$$f(x) = \operatorname{argmax}_y P(X|Y)P(Y)$$

위의 내용을 토대로, 베이지 공식을 적용하면 $P(y|x)$ 를 구해 생성 모델이 분류 모델 역할을 수행할 수도 있다고 생각한다. 대표적인 예가 DCGAN 이다.

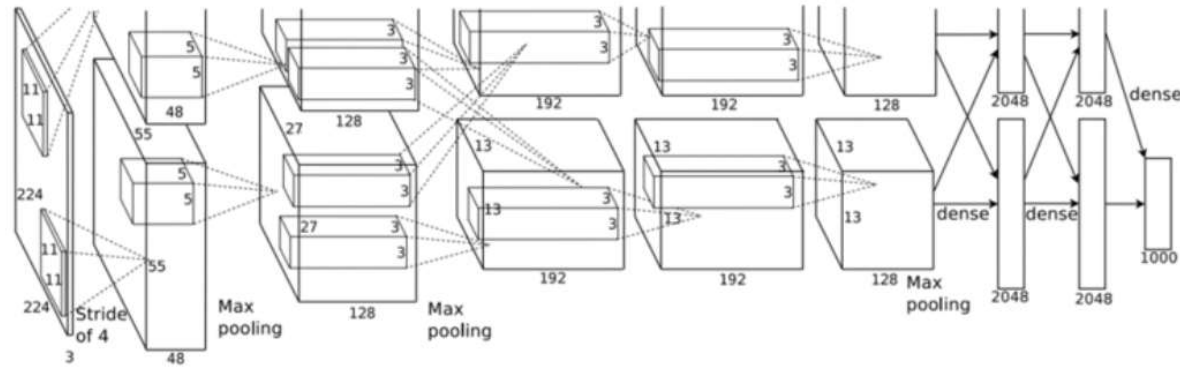
<DCGAN 을 이해하기 쉬운 글>

<https://medium.com/@jongdae.lim/%EA%B8%B0%EA%B3%84-%ED%95%99%EC%8A%B5-machine-learning-%EC%9D%80-%EC%A6%90%EA%B2%81%EB%8B%A4-part-7-2435b4a55ccd>

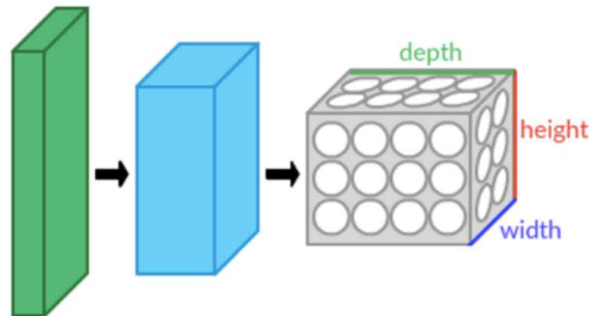
Q4

AlexNet 구조를 통해서 윈도우 크기를 연산하는 과정에 대해 자세히 알고 싶습니다.

A4



AlexNet 은 다음과 같이 5개의 convolution layers와 3개의 fully-connected layers로 구성되어 있으며 마지막 FC layer는 1000개의 카테고리로 분류하기 위해 활성 함수로 softmax 함수를 사용하고 있다.



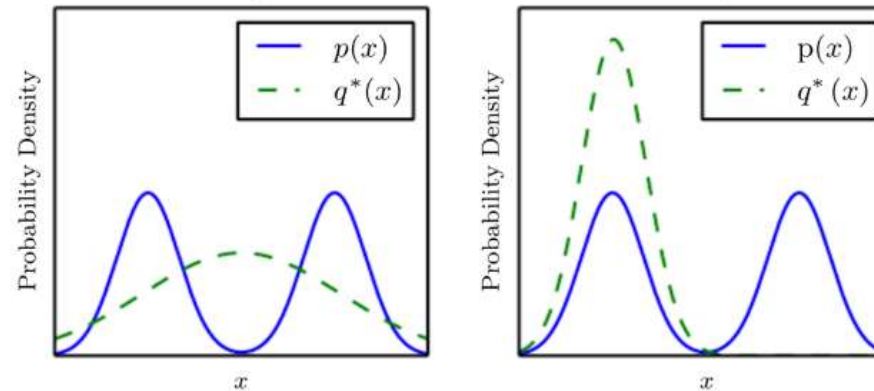
	<p>AlexNet의 블록을 표현하는 그림은 위와 같고 227*227의 3개의 채널 이미지를 사용한다.</p> <p>AlexNet의 첫 번째 convolution layer의 kernel 크기는 11*11*3이며 stride를 4로 적용하며 96개의 feature map을 생성하여 55*55*96의 출력을 가지게 된다.</p> <p>첫 번째 convolution layer를 거치면서, GPU-1에서는 주로 컬러와 상관없는 정보를 추출하기 위한 kernel이 학습되고, GPU-2에서는 주로 컬러와 관련된 정보를 추출하기 위함 kernel이 학습된다.</p> <p>첫 번째 convolution layer 다음은 max pooling으로 27*27*96으로 사이즈를 줄인다.</p> <p>두 번째 convolution layer는 5*5 사이즈의 kernel로 연산하여 27*27*256의 이미지를 출력하고, 그 다음에 max pooling으로 13*13*256으로 사이즈를 줄인다.</p> <p>세 번째 convolution layer는 3*3 사이즈의 kernel로 연산하여 13*13*384의 이미지를 출력한다. 이때는 두 GPU가 서로의 연산 결과를 섞어서 연산한다.</p> <p>네 번째 convolution layer는 3*3 사이즈의 kernel로 연산하여 13*13*384의 이미지를 출력한다.</p> <p>다섯 번째 convolution layer는 3*3 사이즈의 kernel로 연산하여 13*13*256의 이미지를 출력하고, max pooling으로 6*6*256으로 사이즈를 줄입니다.</p> <p>Convolution layer의 연산이 끝나면 두 GPU의 결과를 합쳐 두 층의 fully connected layer로 연산하고, 마지막 softmax를 통해 1000개의 클래스로 구분된다.</p>
질문 내용	조별 모임을 통해 서로의 질문 사항을 모두 해결하였습니다.
기타	

첨부 개별 레포트

이름	이충현	학번	201402665
구분	내용		
학습 범위	기계학습 4장 4.5 생성모델 4.6 딥러닝은 왜 강력한가		
학습 내용	<p><4.5 생성모델></p> <ul style="list-style-type: none"> ● 분별모델은 학습단계에서 수식으로 표현하면 $P(y x)$로 추정한다. 예측단계에서는 $f:x \rightarrow y$ 이고 지도학습이다. 생성모델은 학습단계에서 수식으로 표현하면 $P(x)$또는 $P(x y)$또는 $P(x,y)$으로 추정된다. 예측단계에서는 $x \rightarrow y, y \rightarrow x, f \rightarrow x, y$ 으로 나타낼수 있고 비지도 학습이다. ● 실제 상황에서 생성모델은 현대 기계 학습은 주로 딥러닝 모델을 사용하여 확률 분포를 암시적으로 표현한다. GAN(generative adversarial network), VAE(variational autoencoder), RNN 등등 ● GAN 의 아이디어 = 생성기 G(가짜 샘플) 분별기 D(진위구별)의 대립 구조, 결국 가짜가 진짜처럼 보이게 할수록 성능이 뛰어나다. 최종 목표는 G 가 만들어내는 샘플을 D 가 구별하지 못하는 수준까지의 학습이다. <div data-bbox="851 908 1543 1243"> <p>그림 4-34 GAN의 원리</p> </div> <ul style="list-style-type: none"> ● GAN 의 목적함수 게임이론 타입의 목적함수로 두 명의 플레이어(GG 와 DD)가 싸우면서 서로 균형점(nash equilibrium)을 찾아가도록 하는 방식이다. 		

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log \{1 - D(G(z))\}]$$

- GAN의 이론적 배경은 탄탄하지만, 실제 적용에는 많은 문제가 있다. 대표적으로 학습이 어렵다는 점을 꼽을 수 있다. MODEL COLLAPSING은 우리가 학습시키려는 모형이 실제 데이터의 분포를 모두 커버하지 못하고 다양성을 잃어버리는 현상이다. 그저 손실(loss)만을 줄이려고 학습을 하기 때문에 G가 전체 데이터 분포를 찾지 못하고, 아래 그림처럼 한번에 하나의 *mode*에만 강하게 몰리게 되는 경우이다.



- 개선된 GAN은 특징 매칭, 가상 배치 정규화, 미니배치 분별 등 기법 적용. $P(x)$ 대신 $P(x, y)$ 을 추정하여 레이블이 있는 샘플을 생성->준지도 학습에 활용한다.
DCGAN은 DMLP 대신 CNN을 사용한다. CNN과 데이터 흐름이 반대이다.

<4.6 딥러닝은 왜 강력한가>

- 고전적인 방법에서는 분할, 특징 추출, 분류를 따로 구현한 다음 이어 붙음, 데이터셋 적어도 괜찮다. 전이 학습 가능하다(특정 구간만 재학습).
현대 방법에서는 딥러닝은 전체 과정을 동시에 최적화(통째 학습이라 부른다.)하므로 데이터 셋, 파라미터가 많이 필요하다.
- 계층적 특징에서 깊은 신경망에서는 층의 역할이 잘 구분된다. 반면 얇은 신경망은 하나, 두개의 은닉층이 여러 형태의 특징을 모두 담당한다.

질문 내용	
-------	--

이름	김영연	학번	201500629
구분	내용		
학습 범위	4.5 생성 모델 4.6 GAN은 왜 강력한가?		
학습 내용	<ul style="list-style-type: none"> ● 4.5 생성모델 ● 분별 모델과 생성모델 비교 <ul style="list-style-type: none"> ->분별 모델: 카테고리만을 예측하는데 관심이 있는 모델 <li style="padding-left: 40px;">x라는 샘플이 있을 때 이 샘플의 카테고리가 y인 확률, 즉 사후 확률 $P(y x)$만을 추정하면 됨 -> 생성 모델: 분별 모델과 달리 x가 발생한 확률인 $P(x)$나 카테고리 y에서 x가 발생할 확률 $P(x y)$를 명시적으로 계산하고 이 확률 정보를 이용해서 새로운 샘플 생성 가능 ■ GAN <ul style="list-style-type: none"> -> GAN의 구성: GAN은 회귀생성 모델로서 분류를 담당하는 모델(판별자 D)과 회귀생성을 담당하는 두 개의 모델(생성자 G)로 구성되어 있다. <li style="padding-left: 40px;">두 모델은 생성자 G와 판별자 D가 서로의 성능을 개선해 적대적으로 경쟁해 나가는 모델이다. <li style="padding-left: 40px;">수학적으로 생성자G는 원데이터의 확률분포를 알아내기 위해 노력하며, 학습이 종료된 후에는 원데이터의 확률분포를 따르는 새로운 데이터를 만들어 내게 된다. 		

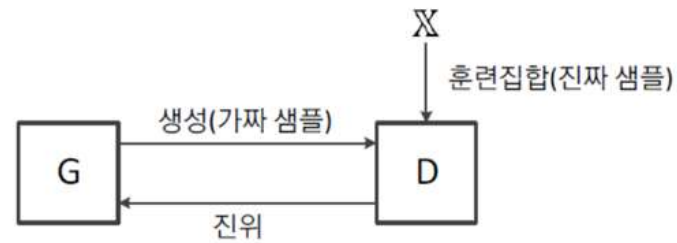


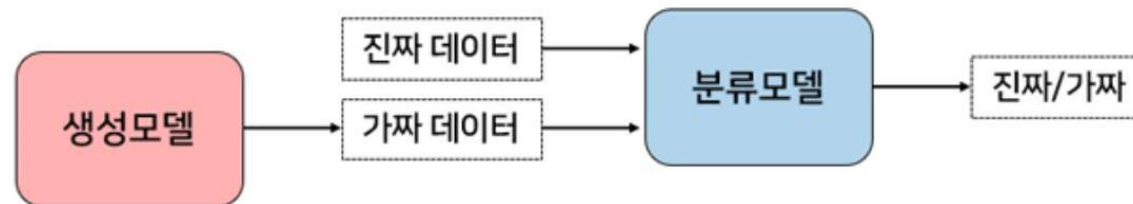
그림 4-34 GAN의 원리

- 적대적 구조: GAN과 같은 적대적 학습에서는 분류 모델을 먼저 학습시킨 후 생성 모델을 학습시키는 과정을 서로 주

고 받으며 반복한다. 분류 모델의 학습은 크게 두가지로 나눌 수 있다.

첫번째는 진짜 데이터를 입력해서 네트워크가 해당 데이터를 진짜로 분류시키도록 학습시키는 과정이다

두번째는 첫번째와 다르게 생성 모델에서 생성한 가짜 데이터를 입력하여 해당 데이터를 가짜로 분류하도록 학습시키는 과정이다.



- 4.6 딥러닝은 왜 강력한가?

-전체과정을 동시에 최적화: 통째학습

	-깊이의 중요성 -계층적 특징
질문 내용	GAN의 학습은 생성자와 분류자가 서로 경쟁하며 학습하는 모델이다. 생성자나 분류자 둘 중 하나가 성능이 특히 뛰어난 경우에도 학습이 잘 되는지 궁금하다. 또한 학습이 잘 되지 않는다면 이를 해결할 방법이 무엇인지 궁금하다.

이름	위성조	학번	201402033
구분	내용		
학습 범위	4.1 딥러닝의 등장 4.2 깊은 다층 퍼셉트론 4.3 컨볼루션 신경망 4.4 컨볼루션 신경망 사례연구		
학습 내용	분별(Generative) 모델 : 지도 학습. 주어진 데이터 X가 Y일 확률 학습 및 예측 생성(Discriminant) 모델 : 클래스가 주어졌을 경우 X를 생성		

표 4-1 분별 모델과 생성 모델

모델	학습 단계가 할 일	예측 단계가 할 일	지도 여부
분별 모델	$P(y x)$ 추정	$f: \mathbf{x} \mapsto y$	지도 학습
생성 모델	$P(\mathbf{x})$ 또는 $P(\mathbf{x} y)$, $P(\mathbf{x}, y)$ 추정	$f: \text{씨앗} \mapsto \mathbf{x}$ 또는 $f: \text{씨앗 } y \mapsto \mathbf{x}$, $f: \text{씨앗} \mapsto \mathbf{x}, y$	비지도 학습

실제 상황에서 자연계 전체에 대한 데이터 발생 분포 $P_{data}(x)$ 를 알아낼 수 없으며, $P_{data}(x)$ 를 모방하는 모델의 확률 분포 $P_{model}(x; \theta)$ 를 명시적으로 추정하는 것도 불가능하다. 현대 기계학습에서는 주로 딥 러닝 모델을 사용하여 확률 분포를 암시적으로 표현한다.

분별 모델과 생성 모델을 서로 경쟁시키며 네트워크를 학습시킨다.

생성기가 만들어내는 샘플을 분별기가 판단할 수 없는 수준까지 네트워크를 학습

GAN의 목적함수

목적함수 J_D, J_G 로 로그우도를 사용

분별기 D의 목적함수 J_D

$$\hat{\theta}_D = \operatorname{argmax}_{\theta_D} J_D(\theta_D)$$

$$\begin{aligned} \text{이때 } J_D(\theta_D) &= \log(f_D(\mathbf{x}^{\text{real}})) + \log(1 - f_D(\mathbf{x}^{\text{fake}})) \\ &= \log(f_D(\mathbf{x}^{\text{real}})) + \log(1 - f_D(f_G(\mathbf{z}))) \end{aligned}$$

생성기 G의 목적함수 J_G

$$\hat{\theta}_G = \operatorname{argmin}_{\theta_G} J_G(\theta_G)$$

$$\text{이때 } J_G(\theta_G) = \log(1 - f_D(f_G(\mathbf{z})))$$

J_D 와 J_G 가 서로 상반관계에 있으므로, 결과는 1/2로 수렴한다.

Model collapsing

모델이 전체 데이터에 fitting 되지 않고, 일부 데이터에 대해서만 fitting 되는 현상

GAN의 학습

분별기의 경우, 최대화 문제이므로 그래디언트를 더해 주고,

생성기의 경우, 최소화 문제이므로 그래디언트를 빼 준다.

GAN은 CNN과 데이터 흐름이 반대.

벡터를 입력받아 영상을 출력(컨볼루션 연산이 진행될 때마다 채널의 수를 줄여나가는 방식)

딥 러닝이 강력한 이유

고전적인 방법에서는 분할, 특징 추출, 분류를 따로 구현한 다음 이어 붙였으며, 각 과정이 사람의 직관에 의존하므로 성능이 떨어진다.

	<p>어지며, 인식 대상마다 새로 설계해야 했음.</p> <p>(데이터 양이 적은 경우에도 분류 모델을 만들 수 있다는 장점이 있음)</p> <p>딥 러닝은 전체 과정을 동시에 최적화 함 (통째 학습이라고 부름)</p> <p>딥 러닝의 앞 부분(저급 특징을 추출하는 부분)은 다른 분류에서도 활용할 수 있음(전이학습)</p> <p>고전적인 방법의 경우 특징공간에 문제가 생기면 문제점을 대상으로 fine tuning을 하면 되지만,</p> <p>딥 러닝은 전체를 재학습 시켜야 하는 단점도 있음</p> <p>은닉층의 깊이가 깊을수록 표현력이 좋으며, 성능이 좋음.</p> <p>깊은 신경망에서는 각 층의 역할이 잘 구분되나, 얇은 신경망은 한 층이 여러 형태의 특징을 모두 담당함</p>
질문 내용	

이름	이재은	학번	201502469
구분	내용		
학습 범위	4.5절: 생성모델 GAN 4.6절: 딥러닝이 강력한 이유		

#중간고사 주요 문제 오답노트

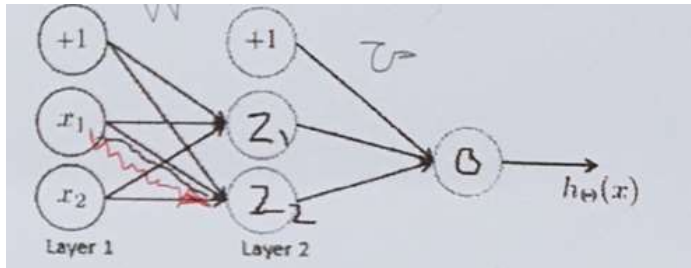
[문제 1.C] Fisher의 Linear discriminant analysis는 데이터의 차원을 효과적으로 낮추는데 이용된다.

틀린 이유: 선형분류기라는 개념만 떠올라, 데이터의 차원을 효과적으로 낮춘다는 설명이 Manifold의 설명이 아닐까 의문점이 들어서 답을 적지 않았다.

내용: Fisher's linear discriminant (FLD)는 데이터를 여러 변수들의 선형결합으로 표현하였을 때 서로 다른 그룹을 잘 구분할 수 있게 해 주는 coefficient를 찾는 방법으로 선형분류법이다. 다시 말해, classification의 일종으로 n차원의 데이터를 직선상으로 투사함으로써 차원을 낮추는 방법이므로 데이터의 차원을 효과적으로 낮춘다는 말은 옳다.

학습 내용

[문제 3] 오류역전파 기법을 사용한다 가정할 때, 가중치 $W_{1,2}$ 의 update 방정식을 유도하라. (단, 활성화함수는 계수 1의 sigmoid 함수이다.)



$$\begin{aligned}
\frac{\partial J(w, v)}{\partial w_{1,2}} &= \frac{\partial (0.5(y-0)^2)}{\partial w_{1,2}} = -(y-0) \frac{\partial 0}{\partial w_{1,2}} = (0-y) \frac{\partial \tau(0sum)}{\partial w_{1,2}} \\
&= (0-y) \tau'(0sum) \frac{\partial 0sum}{\partial w_{1,2}} = (0-y) \tau'(0sum) \frac{\partial 0sum}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_{1,2}} \quad (\because \text{chain rule}) \\
&= (0-y) \tau'(0sum) v_{2,1} \frac{\partial z_2}{\partial w_{1,2}} = (0-y) \tau'(0sum) v_{2,1} \frac{\partial \tau(zsum_2)}{\partial w_{1,2}} \\
&= (0-y) \tau'(0sum) v_{2,1} \cdot \tau'(zsum_2) \cdot \frac{\partial zsum_2}{\partial w_{1,2}} = (0-y) \tau'(0sum) v_{2,1} \cdot \tau'(zsum_2) x_1
\end{aligned}$$

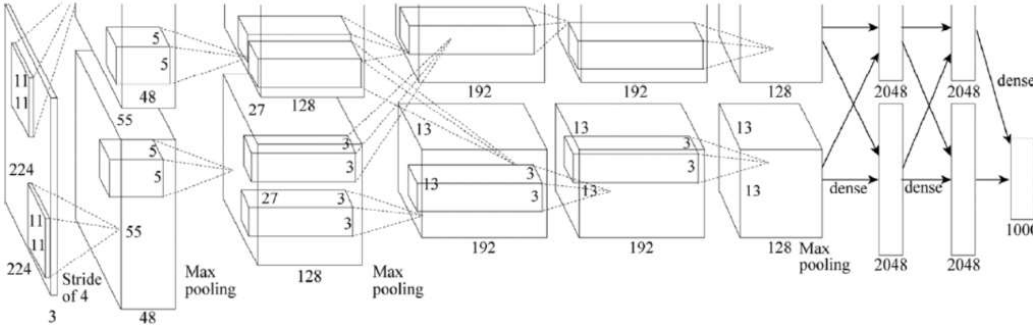
#기계학습 4.3

-패턴 인식에서 분류에 쓰이는 모델은 Generative model(생성 모델)과 Discriminative model(분별 모델) 2 가지이다. Generative 모델은 sample data set 을 생성하는 모델이다. 다시 말해서 학습하는 데이터의 분포를 학습하게 된다. 반면에 판별 모델은 샘플을 생성하지 못하며, 주어진 X로 Y가 무엇인지 학습하는 것이다.

-GAN: 비지도학습에 사용되는 인공 지능 알고리즘으로, 최대한 진짜 같은 데이터를 생성하려는 생성 모델과 진짜와 가짜를 판별하려는 분별 모델이 각각 존재하여 서로 적대적으로 학습합니다.

-개선된 GAN: 특징 매치, 가상 배치 정규화, 미니배치 분별 등 기법 적용, $P(x)$ 대신 $P(x,y)$ 를 추정하여 레이블이 있는 샘플을 생성

- 딥러닝은 왜 강력한가? 전체 과정을 동시에 최적화, 깊은 신경망, 계층적 구조

질문 내용	 <p>그림 4-21 AlexNet 구조[Krizhevsky2012]</p> <p>AlexNet 구조를 통해서 윈도우 크기를 연산하는 과정에 대해 자세히 알고 싶습니다.</p>
-------	--

이름	최진성	학번	201403474
구분	내용		
학습 범위	기계학습 4장 딥 러닝 4.5 생성 모델 4.6 딥러닝은 왜 강력한가?		
학습 내용	기계학습 4장 딥 러닝 4.5 생성 모델 Sample data set을 생성할 수 있는 모델 – 이제까지의 훈련 집합을 토대로 비슷한 훈련 모델을 구축함. 분별 모델 vs 생성 모델 분별 모델 : 지도 학습, 학습단계는 $P(y x)$ 추정, 사후 확률을 측정하는 용도, 입력된 데이터가 어떤 기준에 의하여 분류되는지 판별.		

생성 모델 : 비지도 학습, 학습 단계는 $P(x)$ or $P(x|y)$ or $P(x, y)$ 추정, 사후에서 사전을 추정하여 새로운 훈련 샘플 제작, Parameter가 주어졌을 때 생성되는 모델

$P_{data}(x)$ -> 자연계에 내재한 데이터 발생 분포 -> 일반적으로 알아낼 수 없음.

$P_{model}(x;\theta)$ -> 자연계 데이터 발생 분포 $P_{data}(x)$ 를 모방하는 확률 분포 모델 -> 명시적인 추정 불가능. -> 딥러닝 모델을 이용하여 암시적으로나마 표현함

GAN

Generative Adversarial Network

현재 모델을 생성하는 생성 모델들 중 가장 높은 효과를 보여주는 모델.

생성 모델에서 데이터를 생성하여 수집한 데이터와 섞어놓은 후 분류 모델이 인위적 생성인지 자연적인 데이터인지 구분, 최종 목표는 분류 모델이 분류를 하기 힘들 정도로 판단을 힘들게 하는 것.

최초의 GAN은 G와 D를 DMLP로 구현.

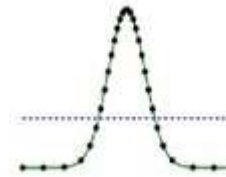
* 난수 생성기 R에서 얻은 난수 r 을 통해 가짜 데이터를 만들고 그것을 분류 모델에 입력시키는 형식.

* GAN의 목적함수는 MSE 대신 로그우도를 사용한다.

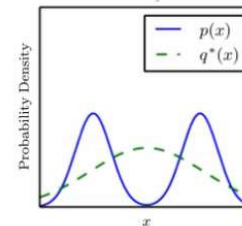
Model Collapsing

GAN에서 자연 데이터와 인공 데이터의 분류 확률이 같이 50%로 수렴하지 않는 경우.

이상적인 수렴 그래프



Model Collapsing.이 나기 쉬운 환경의 그래프



개선된 GAN

기존의 GAN에 특징 매칭, 가상 배치 정규화, 미니 배치 분별 등의 기법을 적용함.

$P(x)$ 대신 $P(x,y)$ 를 추정하여 레이블이 있는 샘플을 생성 -> 준지도 학습에 활용.

* 준지도 학습 : 목표값이 표시된 데이터와 표시되지 않는 데이터를 모두 훈련에 사용하는 경우.

DCGAN -> Deep Convolution GAN

DMLP 대신 CNN을 사용하며, CNN과 데이터 흐름이 반대.

즉, 벡터를 입력 받아서 $3 * 64 * 64$ 영상을 출력한다.

4.6 딥러닝은 왜 강력한가?

고전적 방법 : 데이터를 분할, 특징추출, 분류를 따로 구현한 다음 이어 붙인다

	<p>단, 사람의 직관에 따르므로 성능적 한계가 명확하며, 인식 대상, 혹은 종류가 달라지면 그 데이터에 맞춰서 새로 설계 필요.</p> <p>딥러닝 방법 : 전체 과정을 동시에 최적화(-> 통째 학습)</p> <p>*특징 추출에서 저급 특징, 고급 특징을, 그 후 분류.</p> <p>깊이의 중요성.</p> <div data-bbox="730 539 1715 778" data-label="Figure"> </div> <p>실선 : 20개 노드를 가진 은닉층 하나의 신경망 점선 : 각각 10개 노드를 가진 은닉층 두 개의 신경망. -> 더 정교한 분할</p> <p>계층적 특징</p> <p>깊은 신경망에서는 층의 역할이 잘 구분 됨.(역할에 따라 층을 나누는 경우가 많기 때문)</p> <p>얕은 신경망은 하나 또는 두 개의 은닉층이 여러 형태의 특징을 모두 담당.</p>
질문 내용	<p>1. GAN의 최종 목적은 분류 모델이 최종적으로 정확한 분류를 하지 못 할 정도의 생성 모델 성능을 내는 것인데, GAN 과정에서 분류 모델의 성능 향상은 없는 것인가? 만약에 분류 모델의 성능 향상이 생성 모델의 성능 향상보다 더 뛰어나다면 사용자가 임의로 중지할 때 까지, 혹은 제한된 횟수가 다 될 때 까지 훈련을 하는 것인가?</p> <p>2. 생성 모델이 아닌 분류 모델의 성능 향상을 목표로, 생성 모델과 분류 모델을 동시에 가지고 있는 신경망은 없을까?</p>

