케이스 스터디

1. 왜 케이스 스터디를 하나요?

효과적인 합성곱 신경망의 사례들

하나의 컴퓨터 비전 작업에서 잘 작동한 구조가 다른 작업에도 유용하고 잘 작동한다.

* LeNet - 5
* AlexNet
* VGG
* ResNet
* Inception

2. 고전적인 네트워크들

1) LeNet - 5

목적 – 흑백으로 된 손글씨 인식

적은 변수를 가진다 (60,000개 정도)

요즘과는 다르게 풀링층 뒤에 비선형함수를 적용한다

신경망의 왼쪽에서 오른쪽으로 이동할 때 높이와 너비가 감소한다 (32->28->14->10->5)

채널의 수는 증가한다 (1->6->16)

2) AlexNet

목적 - 이미지를 1000개에 해당하는 클래스로 분류하는 것

LeNet과 유사하지만 훨씬 큰 크기를 가진다

많은 매개변수를 가진다 (6천만개 정도)

활성화 함수로 ReLU를 사용한다

3) VGG -16

간결한 구조를 가진다

많은 하이퍼파라미터를 가지는 대신 stride가 1인 3\*3 필터만을 사용해 동일합성곱을 하고 max pooling layer에서는 2\*2를 사용한다

산출값의 높이와 넓이는 매 최대 풀링 마다 1/2씩 줄어들며, 채널의 수는 두배 혹은 세배로 늘어나게 만든다

단점 - 훈련시킬 변수의 개수가 많아 네트워크의 크기가 커진다

3. ResNet

아주 깊은 신경망을 훈련시키기 어려운 이유 : 경사 소실 / 경사 폭발

스킵 연결 : 한 층의 활성값을 가지고 훨씬 깊은 층에 적용하는 방식

a^[l] -> linear -> relu -> a^[l+1] -> linear -> relu -> a^[l+2]

ResNet 에서는 z^{[l+2]} 에 비선형성을 적용해주기 전에 a^{[l]} 을 더하고 이것을 다시 비선형성을 적용한다 (short cut = skip connection)

a^{[l]}의 정보는 short cut을 따라서 신경망의 더 깊은 곳으로 갈 수 있다

잔여 블록 : a^{[l]}을 더해서 다시 활성화 함수에 넣는 부분까지

경험적으로 층의 개수를 늘릴 수록 훈련 오류는 감소하다가 다시 증가한다. 하지만 이론 상으로는 신경망이 깊어질 수록 훈련 세트에서 오류는 계속 낮아져야 한다. 하지만 ResNet 에서는 훈련오류가 계속 감소하는 성능을 가질 수 있다.

4. 왜 ResNet이 잘 작동할까요?

신경망의 깊이가 깊어져도 훈련 세트를 다루는데 지장이 없다

ResNet에서는 큰 신경망에서 두개의 층을 더 추가하고 지름길을 연결해준다. 여기서 활성화함수 g는 ReLU이다.

항등식의 의미는 신경망으로 하여금 스킵 연결을 통해 두 층이 없는 더 간단한 항등식을 학습하여, 두 층 없이도 더 좋은 성능을 낼 수 있게 만든다는 것이다.

5. Network 속의 Network

합성곱 신경망을 구축할 때 1 x 1 합성곱은 매우 유용하다.

채널의 수를 원하는대로 줄이거나 늘릴 수 있게 만들어준다.

195 개의 입력숫자가 32개의 1 x 1 필터와 합성곱을 하여 32 개의 출력 숫자가 된다. 즉, 이는 입력 채널의 수만큼 유닛을 입력으로 받아서, 이들을 하나로 묶는 연산과정 통해, 출력채널의 수만큼 출력을 하는 작은 신경망 네트워크로 간주할 수 있다. 따라서 네트워크 안의 네트워크라고도 한다. 이처럼 1x1 합성곱 연산을 통해 비선형성을 하나 더 추가해 복합한 함수를 학습시킬 수 있고, 채널수를 조절해줄 수 있다.

6. Inception 네트워크의 아이디어

필터의 크기나 풀링을 결정하는 대신 전부다 적용해서 출력들을 합친뒤 네트워크로 하려금 스스로 변수나 필터 크기의 조합을 학습하게 만드는 것이다.

인셉션 네트워크의 문제는 계산 비용이다. 이를 1 x 1 합성 곱으로 해결 할 수 있다. – 병목 층

* 1 x 1 합성곱: 28 x 28 x 16 x 1 x 1 x 192 =  약 240 만개
* 5 x 5 합성곱: 28 x 28 x 32 x 5 x 5 x 16 = 약 1000 만개

학습에 필요한 계산 비용이 1/10 수준으로 크게 줄어든 것을 알 수 있다. 병목층을 사용시 표현의 크기가 줄어들어 성능에 영향을 지장을 줄지 걱정 될 수도 있는데, 적절하게 구현시 표현의 크기를 줄임과 동시에 성능에 큰 지장 없이 많은 수의 계산을 줄일수 있다.

7. Inception 네트워크

인셉션 모듈 – 활성값이나 이전 층의 출력을 입력값으로 받는다.

인셉션 네트워크 – 인셉션 블록들의 반복으로 이루어졌다

인셉션 네트워크는 구글넷이라고도 한다