오늘 발표 할 주제는 deep residual learning for image recognition이라는 논문입니다.

논문에 대해 설명하자면 2015년에 RESNET에서 발표된 논문으로, 딥러닝의 성능을 향상 시킨 방법입니다.deep residual learning은 깊은 신경망에서 흔히 발생하는 기울기 소실 문제를 해결하기 위해 Residual Block을 도입한 모델입니다. 논문 발표 이후 3.57%의 error를 달성해서 ILSVRC 2015에서 1등을 차지했습니다.

deep residual learning의 탄생배경에 대해 소개하겠습니다. deep residual learning이 탄생 하기 이전에는 이전의 neural network은 degradation라는 문제를 갖고 있었습니다. 일반적으로 이전의 neural network는 x를 가지고 weight layer를 거치면서 학습을 한 후 나온 hx를 activation function을 적용후 결과를 도출을 하는 구조입니다. 하지만 문제는 Layer가 깊어질수록 train,test 에러가 증가하고, 성능이 떨어집니다. 이는 아래의 그래프를 통해 알수 있습니다. 56개의 레이어,20개의 레이어를 가지고 error를 측정한 결과 training error,test error부분에서 56개의 레이어 부분이 20개의 레이어보다 높은 error률을 기록했습니다. 다음으로 일정 이상의 layer의 수가 증가할때 vanishing/exploding gardients라는 문제가 발생한다고 합니다.이때 vanishing/exploding gardients라는 문제란 backpropagation 과정에서 기울기가 0으로 수렴되는 현상입니다. backpropagation에서 gradient를 계산할 때, 각 층의 activation fucntion의 미분 값이 곱해집니다.각 층을 거치면서 작은 값의 미분이 연속적으로 곱해지게 되면, 기울기 값은 지수적으로 감소하게 됩니다. 결과적으로, 네트워크가 깊어질수록 학습이 제대로 이루어지지 않으며, 이 상태를 기울기 소실,vanishing gradient이라고 합니다.이때, 논문에선 vanishing/expolding gradient가 일어나는 원인은 overfitting 문제는 아니라고 밝힙니다. 이러한 두 문제점을 해결하고자 RESNET에선 deep residual learning 이라는 알고리즘을 제시했습니다.

deep residual learning은 x를 identity mapping을 하면서 fx를 학습을 하게 하고 난뒤 x와fx 합치는 구조로 되어 있습니다.이때 논문에서는 shortcut이라는 아이디어를 제시합니다. 이전의 neural network는 x를 가지고 weight layer를 거치는데 weight layer가 깊어질수록 학습의 난이도가 높아져 네트워크의 난이도가 올라간다고 합니다. 이때 deep residual learning에서는 이전에는 실제로 내제한 mapping인 $H(x)$를 곧바로 학습하는 것은 어려우므로 대신 Fx로 mapping을 하면서 학습을 하게 합니다. 이로 인해 hx를 fx로 mapping을 하면 네트워크의 최적화난이도를 낮추게 만들수 있습니다. 이때 x를 identity mapping을 하는 과정을 shortcut연결이라 불립니다.

deep residual learning을 나타내는 수식은 다음과 같습니다. f(x,{w_i)는 fx를 학습하는 과정을 나타내고 이때 multiple convolution layers이라 불립니다. 이때 i는 weight layer의 층 변호를 의미합니다. w_s*X는 x를 identity mapping을 하는 과정이고 shortcut이라 불립니다. 이때 w_s은 x의 demension을 맞추는 과정이라고 합니다.

다음은 논문에 있는 related work에 대한 내용입니다.Residual Representations은 이미지 인식에서 입력 벡터 대신 **잔차 벡터(residual vectors)**를 인코딩하여 더 나은 성능을 얻는 방식입니다. 이 방식은 복잡한 문제를 간단히 풀어내고, 더 빠르게 결과를 도출할 수 있습니다.이럐 VLAD, Fisher Vector가 residual vector의 개념을 사용합니다.shortcut connection(지름길 연결)방법은 신경망에서 입력을 여러 층을 거치지 않고 바로 출력으로 연결합니다. 이 방식은 깊은 신경망에서 발생하는 문제(예: 기울기 소실)를 해결하고, ResNet처럼 더 깊은 네트워크에서도 성능을 향상시킵니다.

마지막으로 deep residual learning의 실험 및 측정 결과에 대해 설명하겠습니다. 18개의 레이어,34개의 레이어를 가진 plain network와 residual network를 가지고 실험을 진행했습니다. 이때 34개의 레이어의 가진 plain network와 residual network의 구조는 figure 4를 통해 알수있습니다. 특히 34개의 레이어의 residual network에서 table1에서 처럼 layer들이 필터의 개수를 기준으로 3개,4개,6개,3개씩 띄어져 있습니다.그리고 18개의 레이어의 residual network도 이전의 residual

network 처럼 구조를 이루고 있습니다.

ImageNet Classification분야에서 plain network와 residual network를 실험하고 비교한 결과 plain network는 레이어가 깊어질수록 error가 높아지는 반면 residual network는 레이어가 깊어질수록 error가 낮아진다고 합니다. 이러한 실험 결과는 figure 4,table2에서 확인할 수 있습니다.

그리고 두가지 방법이 있는데 Identity mapping ,Projection Shortcuts이 있습니다.실험에서 zero-padding을 이용하여 demension을 늘리고 identity mapping 사용하는 방법A,demension이 늘어날때마다 projection 사용하는 방법B,마지막으로모든 shortcut에 projection 사용하는 방법C으로 나눠 실험을 진행했습니다. 실험 결과, 성능이 C>B>A순으로 나타납니다. 원인은 A는 residual learning 이 이뤄지지 않기 때문에 B가 성능이 미세하게 더 좋고, C가 B보다 좋은 이유는 projection shortcut에 사용된 extra parameter때문입니다.하지만 이러한 A,B,C 들 사이의 작은 차이는 주요 문제가 projection shortcut 때문에 생기는 것이 아니라는 것을 뜻합니다. 이를 통해 projection shortcut이 degradation 문제를 해결하는데 필수적이지는 않다는 것을 확인할 수 있었습니다. 따라서 memory / time complexity와 model size를 줄이기 위해 이 논문에서는 C 옵션을 사용하지 않는다고 합니다. 특히 A인 Identity shortcut은 bottleneck 구조의 복잡성을 높이지 않는데에 매우 중요하다고 합니다.

Deeper Bootleneck Architectures에 대해 설명하겠습니다. 1x1 -> 3x3 -> 1x1 순서로 구성된 세 가지 필터로 이루어져 있습니다. 첫 번째 1x1 filter에서는 256의 dimension을 64개의 dimension으로 차원 축소를 하는 과정이고 이후 3x3 filter로 공간적인 특징 추출을 진행합니다. 이후 다시 1x1 filter로 256개의 dimension으로 확장하는 과정을 진행합니다. 1x1 filter의 개수만큼 channel이 생기는데, 여기서 64개이므로 channel이 강제적으로 64개로 축소가 된다. 축소하는 이유는 연산량을 줄이기 위해서이다. 이후, 3x3 convolution으로 공간적인 특징을 추출하고, 이후 원래 dimension인 256으로 확장합니다. 이는 연산량을 최소화하기 위해서 사용하는 방법이며 결과적으로 parameter의 수를 감소시키는 효과를 갖는다. 따라서 연산 시간 감소 성능을 얻을 수 있다고 합니다.

CIFAR10 and Analysis에서는 ImageNet Classification분야와 같이 이전까지의 아키텍처와 다르게 레이어가 깊을수록 성능이 항상하는 걸 table 6과 figure 6을 통해 알수 있습니다. 그러나 표에서 레이어가 1202개 일때 다른 레이어 개수와 비교할때 error가 높게 나옵니다. 이를 통해 과도하게 깊을수록 성능이 감소되는 걸 알 수있습니다.논문에서 이러한 원인을 overfitting으로 보고 있습니다.

---

identity mapping
Identity Mapping은 입력 값을 그대로 출력으로 전달하는 과정.
이를 통해 네트워크가 더 깊어져도 기울기 소실 문제를 방지하고, 성능을 높일 수 있음.

VLAD(Vocabulary of Local Aggregated Descriptors) 컴퓨터 비전과 이미지 검색 시스템에서 사용되는 방법으로, 이미지 내의 특징을 효과적으로 표현하고 검색 속도를 높이는 기술입니다. 주로 이미지 검색, 분류, 인식 분야에서 사용되며, **SIFT(Scale-Invariant Feature Transform)**나 **SURF(Speeded-Up Robust Features)**와 같은 지역적인 특징 추출 기법과 결합하여 사용됩니다.
로컬 특징 추출: 이미지에서 SIFT나 SURF와 같은 방법으로 로컬 특징들을 추출합니다.

클러스터링: 추출된 로컬 특징들을 클러스터링 알고리즘, 예를 들어 **K-평균(K-means)**을 통해 특정 중심점으로 그룹화합니다. 이러한 중심점들을 '시각적인 단어(visual words)'라고 부릅니다.
잔차 계산: 각 로컬 특징 벡터와 그것이 할당된 클러스터 중심점 간의 차이(잔차)를 계산합니다.
잔차 합산: 이미지 내의 모든 로컬 특징들에 대해 잔차를 합산하여 최종적으로 하나의 벡터로 요약합니다.
정규화: 결과 벡터를 정규화하여 고차원 공간에서의 표현을 개선합니다.
이 과정을 통해 VLAD는 이미지의 주요 특징을 요약한 고차원 벡터를 생성하며, 이를 통해 대규모 이미지 데이터베이스에서의 이미지 검색이 더 빠르고 정확해집니다.

Fisher Vector(FV) 알고리즘은 컴퓨터 비전과 머신러닝 분야에서 이미지 및 비디오의 특징을 효과적으로 표현하고 비교하기 위해 사용되는 고차원 벡터 표현 방법입니다. VLAD(Vector of Locally Aggregated Descriptors)와 유사하게, Fisher Vector도 지역적 특징을 집계하여 전체 이미지의 표현을 생성하지만, 통계적 모델링을 활용하여 더 풍부한 정보를 담고 있습니다.

로컬 특징 추출: 이미지에서 SIFT, SURF 또는 HOG(Histogram of Oriented Gradients)와 같은 방법을 사용해 로컬 특징을 추출합니다.
GMM 학습: 추출된 로컬 특징들을 사용해 가우시안 혼합 모델을 학습합니다. GMM은 특징 벡터의 분포를 몇 개의 가우시안 분포로 설명합니다.
잔차 계산: 각 로컬 특징이 GMM의 각 가우시안 컴포넌트에 대해 어떻게 변화하는지(잔차)를 계산합니다. 이를 통해 로컬 특징이 전체 이미지의 분포에서 얼마나 벗어나는지를 측정합니다.
잔차를 FV로 표현: 계산된 잔차를 Fisher Vector로 변환하여 이미지 전체를 설명할 수 있는 하나의 고차원 벡터로 요약합니다. 이 과정에서 두 가지 주요 정보를 사용합니다:
가우시안 평균의 변화(첫 번째 미분)
가우시안 분산의 변화(두 번째 미분)
정규화 및 차원 축소: 최종적으로 생성된 벡터는 정규화 및 차원 축소 과정을 거쳐 더 효율적으로 처리할 수 있도록 조정됩니다.

Downsampling이란 신호처리에서 말하는 용어로 sample의 개수를 줄이는 처리과정을 의미합니다. 딥러닝에서는 인코딩할때 data의 개수를 줄이는 처리과정

Residual vector는 통계학과 선형 대수학에서 자주 사용되는 용어입니다. 주로 회귀 분석에서 사용되며, 모델이 예측한 값과 실제 값 사이의 차이를 나타냅니다.

구체적으로, 잔차 벡터(residual vector)는 다음과 같이 정의됩니다:
회귀 분석에서: 모델이 예측한 값과 실제 관측 값 간의 차이로 정의됩니다. 즉, 잔차 벡터는 모델의 예측과 실제 데이터 간의 오차를 나타냅니다.
선형 시스템에서: 선형 시스템의 해결 과정에서 잔차 벡터는 시스템의 예측 결과와 실제 측정 결과

간의 차이를 나타낼 수 있습니다.

잔차 벡터를 분석함으로써 모델의 적합도나 시스템의 정확도를 평가할 수 있습니다.