0. GoogLeNet의 특징

이 모델의 주요 특징은 연산을 하는 데 소모되는 자원의 사용 효율이 개선되었다는 것이다. 즉, 정교한 설계 덕에 네트워크의 depth와 width를 늘려도 연산량이 증가하지 않고 유지된다는 뜻이다. 이때, Google 팀에서는 성능을 최적화하기 위해 Hebbian principle과 multi-scale processing을 적용하였고, 이 구조를 GoogLeNet이라 부른다고 한다. GoogLeNet은 22개의 layer를 가지며, 코드네임은 Inception이다.

1. Introduction: 딥러닝의 발전 동향, GoogLeNet이 나오게 된 배경, 설계 주안점

딥러닝의 발전 동향: 새로운 아이디어, 알고리즘, 개선된 신경망 구조로 인해 CNN 분야의 큰 발전

GoogLeNet이 나오게 된 배경: 적은 parameter로 연산량을 줄여도 정확도를 높이기 위함

설계 주안점: 유연한 구조, Inception module

- “deep”의 의미? "Inception module"의 형태로 새로운 차원의 구조를 도입한다는 뜻 & 네트워크 깊이의 증가

2. GoogLeNet의 구조

Convolution layer가 쌓이고 그 뒤에 1개 이상의 Fully connected layer가 따라오는 구조

layer의 수와 크기를 늘리고, dropout를 이용해 overfitting을 방지하는 구조

convolution layer에서 1x1 convolution layer를 추가하고, ReLU activation이 따라오는 구조

3. GoogLeNet이 나오게 된 배경

deep neural network의 성능을 높이려면 level의 수를 늘리거나(depth ↑), 각 level의 unit 수 증가(width ↑)시켜야 함

- P1: parameter의 수가 증가하고, 컴퓨터 자원의 사용량이 늘어남

- S1: dense한 fully-connected 구조 -> sparsely connected 구조(연관 관계가 높은 것들만 연결)

- P2: 균일하지 않은 Sparse data 구조를 다룰 때 오히려 비효율적

- S2: **Sparse matrix를 clustering 하여 상대적으로 dense한 submatrix를 만듦**

4. Inception module의 구조

CNN에서 각 요소를 최적의 local sparce structure로 근사화하고, 이를 dense component로 바꾸는 방법을 찾는 것

- 입력 이미지의 작은 부분에서 비슷한 특징을 가진 뉴런들이 모여 있는 경우, 이를 처리하기 위해 1x1 convolution을 사용할 수 있음

- 반대로 입력 이미지의 어떤 부분에서는 넓은 크기의 filter가 있어야 특징을 효과적으로 추출할 수 있으므로, 3x3, 5x5 convolution 연산도 병렬적으로 수행 (네트워크가 깊어지면 깊어질수록 더 큰 크기의 feature를 추출해야 하기에 3x3, 5x5 filter의 수가 늘어나야 함)

P: 3x3, 5x5 convolution filter를 사용하면 연산량이 증가함

S: 3x3, 5x5 앞에 1x1를 두어 차원을 축소하고 연산량을 낮춤

결과: 낮은 layer에서는 기본적인 CNN 모델을 사용(Inception module 없이 Conv -> MaxPool 등), 높은 layer에서만 Inception module을 사용

장점: 낮은 연산량으로 각 level의 unit 수를 증가시킬 수 있음, 다양한 크기의 filter를 통해 다양한 특징을 동시에 추출 가능

5. Auxiliary Classifier

P: 모델의 깊이가 깊어질수록, 기울기가 0으로 수렴하는 vanishing gradient problem이 발생 가능

S: 중간에 Auxiliary Classifier를 통해 결과를 출력하여 추가적인 back propagation을 일으킴 -> gradient 전달

6. GAP(Global average pooling)

이전 layer에서 추출된 feature map을 각각 평균 낸 것을 이어서 1차원 벡터로 만들어주는 과정

가중치가 필요하지 않고, fine tuning을 하기 쉽게 만들어 줌