Local Feature로부터 Global Feature를 추출하는 모델

1. 특징추출 \_

2. topology invariant

3. classification

1,2 번을 위해서 filter / sub-sampling (max pooling)

원래 커널의 계수를 정하고 convolution을 돌리지만,

CNN에서는 정하지 않고 특징을 추출할 수 있는 계수를 학습이 진행되면서 정하게 됨

강하고 global한 특징 추출에 초점을 맞춰야 함.

->convolution + max pooling 반복 거치게 됨

Convolution의 결과를 Feature map이라고 부름

강한 신호만 살리는 max pooling을 통해 topology invariance를 얻게 됨.

Convolution과 max pooling을 거치면서 feature map의 크기가 작아 짐.

전체를 대표할 수 있는 강인한 특징만 남게 됨.

마지막에 남게 되는 강인한 특징들만 가지고 classification을 진행하게 됨

위/아래 구조 =>GPU적용

<모델 형태 정하기>

1. Kernel의 개수

입력단 근처에 있는 Layer에서는 Feature map의 크기가 크기 떄문에

Filter의 개수를 적게하고,

입력단에서 멀어질수록 Feature map의 크기가 줄어들기 때문에

Filter의 개수를 많게 한다.

**연산시간. 각 단에서의 연산시간/량을 비교적 일정하게 유지하며 시스템의 균형을 맞추는 것.**

**각 Layer에서 Feature map의 개수와 pixel 수의 곱을 대략적으로 일정하게 유지한다.**

**(여기서 말하는 픽셀의 수가 뭐지?)**

**Feature map의 size? 맞는 듯.**

(각 Layer에서 연산시간) = (출력 Pixel의 수) x (Feature map의 개수) x (filter당 연산 시간)

2. Filter의 형태 **FIilter의 Size**

결과는 여러 개의 작은 크기의 필터를 중첩해서 사용하는 것이 좋다.

**중간 단계에 있는 Non linearity를 활용하여 원하는 특징을 더 돋보이도록 할 수 있다.??**

&

**작은 필터를 중첩해서 사용하는 것이 연산량도 적다.(그럴거 같긴 함)**

3. Stride 값

건너 뛸 픽셀의 개수 결정.

입력 영상의 크기가 큰 경우.

**연산량을 줄이기 위해 입력단과 가까운 쪽에만 적용.**

Stride를 1로 하면 모든 입력 영상에 대해 convolution을 적용하지만.

그리고 pooling을 하면서. 강한 신호를 고를 수 있지만.

Stride를 크게 하면 선택의 기회가 사라진다.

**통상적으로는 stride를 1로 하고 pooling을 통해 적절한 sub sampling을 거치는 게 좋다.**

큰 영상에 대해 CNN을 적용하는 경우,

입력 영상을 직접적으로 처리하는 1단계 Convolution Layer에서 stride를 크게 주기도 함.

4. Padding 지원 여부

Zero padding

-> Feature map size0 유지

+

->경계면의 정보 살릴 수 있음.

=>결과가 좋아짐.

**1.LeNet**

처음 CNN 도입

**2.AlexNet**

Convolution Layer 바로 뒤에 Convolution Layer 적용.

**3.ZF Net**

Convolution Layer의 크기를 늘림.

**GoogleNet**

**Inception Module** 도입.

**4.VGGNet**

Homogeneous structure

망의 시작부터 끝까지 동일하게 3x3 Convolution + 2x2 Max Pooling + 16 depth

=> 최적의 성능

분류 성능은 약간 떨어짐.

다중 전달 학습 과제에서 좋은 결과

단점 : 메모리 수와 파라미터의 수가 크다.

**5.ResNet**

층이 깊어질 때 학습을 쉽게 할 수 있도록 해주는

Residual Framework 개념 도입.

**ImageNet**

**세계 최대 영상 데이터 베이스.**

**ILSVRC**

ImageNet 영상 데이터 베이스를 기반으로 하여 성능 우열을 가리기 위한 대회.

**LeNet**

Convolution + sub sampling을 거치면서 global한 feature를 얻어간다.

의미 있는 global feature를 얻게 되면

FCNN 을 통해 classification 진행.

**5x5 convolution Kernel -> local Receptive Field 개념 적용**

**전체 이미지에 대해 같은 Kernel을 적용 => shared weight 개념 적용**

**가장 큰 자극만 취함(max pooling) =>sub-sampling 가능**

**LeNet-5**

입력 영상의 크기가 커짐.

Detail에 대한 고려가 훨씬 많아짐.->우수한 성능

Feature map이 다음 단계의 Feature map과 모두 연결되지는 않는다.

(6개의 Feature map -> 16개의 Feature map)

AlexNet

GPU를 사용함

**Overlapped pooling**

Pooling 하는 부분이 겹치도록 함

->에러율과 overfitting을 줄임

**Local Response Normalization**

Relu를 이용한 학습 효율화

**Overfitting 해결**

Data Augmentation

DropOut