1️⃣

Image Classification 1

강의 소개

우리는 오감 중 특히 시각에 의존하여 사물을 바라보고 이해하며 살아가고 있습니다.동일한 프로세스를 컴퓨터에 적용한 컴퓨터 비전입니다. 본 강의에서는 컴퓨터 비전 (CV)의 첫 시간으로 CV에 대해 짧게 소개하고, CV에서 가장 기본적인 task, image clasiification을 소개합니다.

Image Classification은 사진이 주어졌을 때 특정 카테고리로 분류하는 task입니다. 이번 강의에서는 먼저 기존의 머신러닝과 구분되는 딥러닝을 사용한 Image classification의 특징에 대해서 배웁니다. 다음으로 대표적인 CNN 모델인 AlexNet을 배우고 이에 대한 실습을 진행합니다. 끝으로 가장 유명한 classification 모델 중 하나인 VGGNet에 대해 배웁니다.

Further Reading

- VGGNet : https://arxiv.org/pdf/1409.1556.pdf

Course overview

Why is visual perception important?

Artificial Intelligence (AI)

```
The theory and development of computer systems able to perform tasks normally
requiring human intelligence, such as visual perception, speech recognition,
decision-making, and translation between languages.
  --from the Oxford dictionary
```

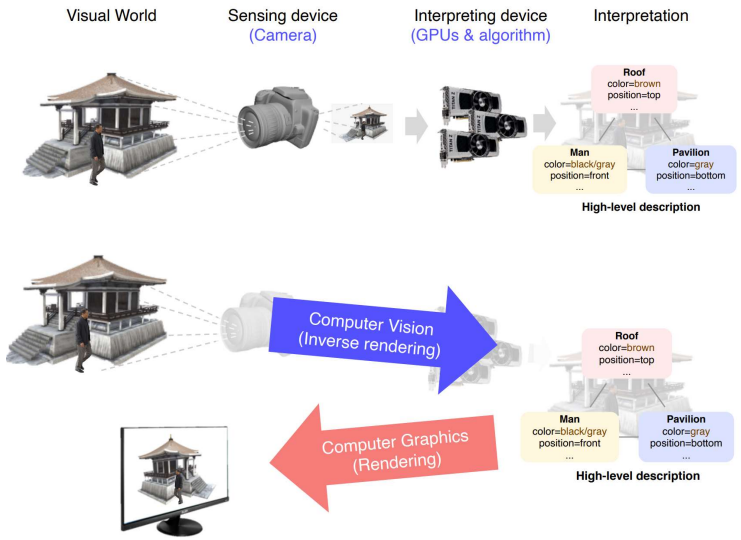
Perception to system?

- Developing machine perception is still open research area

Why is visual perception important?

- Understanding
  - 뇌 영역의 50% 이상 → visual information 처리
- Sensing
  - 정보의 75%는 눈으로부터

What is computer vision?



- Rendering → 정보를 통해 2D 이미지 표현

Visual perception & intelligence

- Input : visual data (image or video)

Class of visual perception

- Color perception

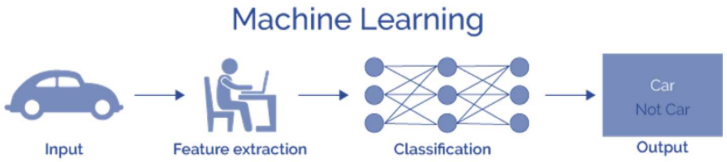
- Motion perception
- 3D perception
- Semantic-level perception
- Social perception (emotion perception)
- Visuomotor perception

Our visual perception is imperfect

- 사람도 똑바로 된 이미지를 '많이' 봤기(= bias) 때문에 거꾸로 된 이미지를 구별 가능

How to implement?

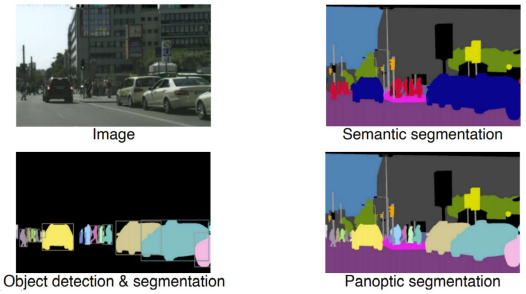
- Machine Learning
  - 사람이 feature extraction



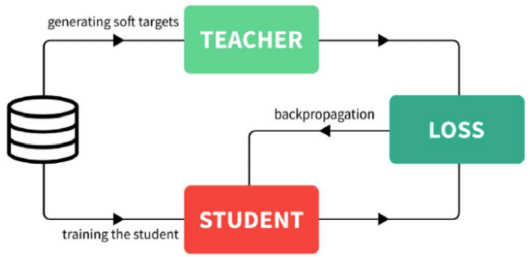
- Deep Learning
  - ▼ 📎 feature extraction
    - **원본 특징 들의 조합으로 새로운 특징을 생성하는 것이다.**
    - 고차원의 원본 feature 공간을 저차원의 새로운 feature 공간으로 투영시킨다. 새롭게 구성된 feature 공간은 보통은 원본 feature 공간의 선형 또는 비선형 결합이다.
    - 가장 대표적인 알고리즘으로 PCA(Principle Component Analysis)가 있다. PCA를 간단히 설명하면 각 변수(Feature)를 하나의 축으로 투영시켰을 때 분산이 가장 큰 축을 첫번째 주성분으로 선택하고 그 다음 큰 축을 두번째 주성분으로 선택하고 **데이터를 선형 변환하여 다차원을 축소하는 방법이다.**
    - 참고 : 기계학습/feature engineering - 인코덤, 생물정보 전문위키 (incodom.kr)
    - 읽어보기 : Feature Extraction Techniques. An end to end guide on how to reduce a… | by Pier Paolo Ippolito | Towards Data Science

What you will learn in this course

Fundamental image tasks



Data augmentation and knowledge distillation



- ▼ 📎 knowledge distillation
  - 크고 무거운 모델의 정보(knowledge)를 작고 가벼운 모델로 전달하여 작고 가벼운 모델이 더 정확한 추론을 하도록 학습시키는 방법론

- Knowledge distillation 의 목적은 "미리 잘 학습된 **큰 네트워크(Teacher network)**의 지식을 실제로 사용하고자 하는 **작은 네트워크(Student network)** 에게 전달하는 것" 입니다.
- 참고 : 1. Knowledge Distillation이란? :: Time Traveler (tistory.com)
- 참고 : 딥러닝 용어 정리, Knowledge distillation 설명과 이해 (tistory.com)

Multi-modal learning (vision + {text, sound, 3D})

- 다른 perception과 vision을 함께 학습

Conditional generative model

Neural network analysis by visualization

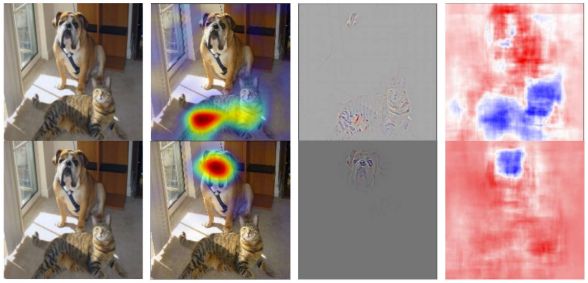
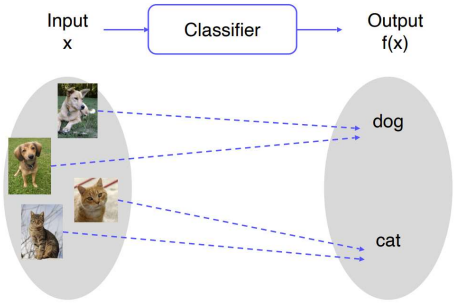


Image classification

What is classification?

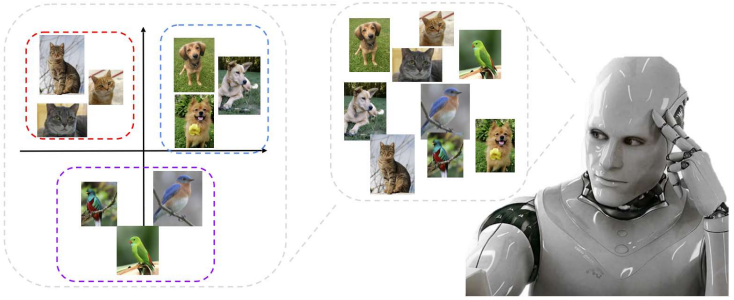
Classifier

- mapping function that maps an image to a **category level**



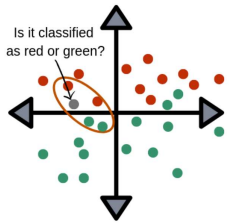
An ideal approach for image recognition

- 세상의 모든 이미지를 memorize → classification 문제를 k Nearest Neighbors (k-NN)으로 해결
- but, 불가능
  - ∵ 모든 데이터를 저장하는 것이 불가능
  - ∵ k-NN하려면 데이터(이미지)간의 유사도를 정의해야 → 유사도 정의 hard
  - ∵ Time complexity (ex. linear search) → O(n) (n=infinite)
  - ∵ Memory complexity → O(n) (n=infinite)



k Nearest Neighbors (k-NN)

- query data 근방의 k개의 data를 보고 query data를 classify

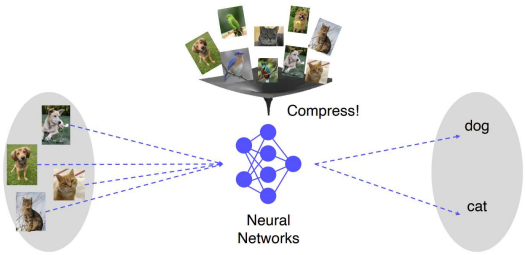


▼ 📎 k-NN

- 새로운 데이터가 입력되었을 때, 기존의 데이터와 새로운 데이터를 비교함으로써 새로운 데이터와 가장 인접한 데이터 k개를 선정한다. 이어서, k값에 의해 결정된 분류를 입력된 데이터의 분류로 확정한다. 즉, 새로 입력된 데이터와 기존 데이터를 비교함으로써 새로운 데이터를 유사하게 판단된 기존 데이터로 분류한다.
- cf) k는 보통 홀수를 많이 사용
- 참고 : kNN(k Nearest Neighbor) 알고리즘 (tistory.com)

Convolutional Neural Networks (CNN)

- 모든 data를 neural network에 compress



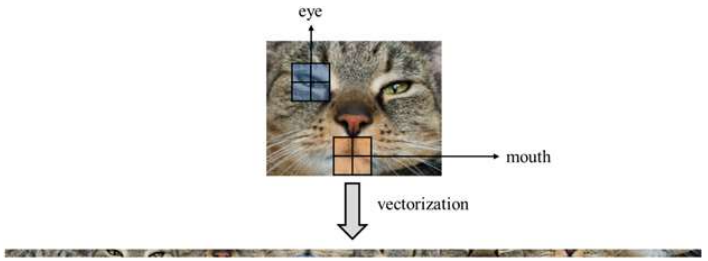
visualization of single fully connected layer networks

▼ 📎 fully connected layer

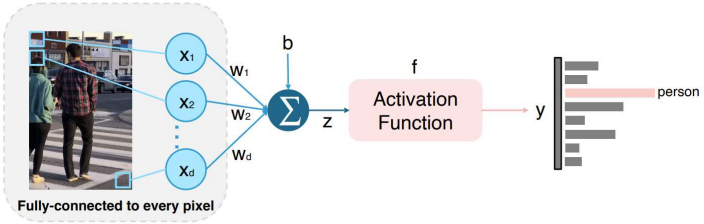
- 한층의 모든 뉴런이 다음층이 모든 뉴런과 연결된 상태로, 2차원의 배열 형태 이미지를 1차원의 평탄화 작업을 통해 이미지를 분류하는데 사용되는 계층입니다.
  - 1. 2차원 배열 형태의 이미지를 1차원 배열로 평탄화
  - 2. 활성화 함수(Relu, Leaky Relu, Tanh,등)뉴런을 활성화
  - 3. 분류기(Softmax) 함수로 분류
- 참고 + 읽어보기 : [딥러닝 레이어] FC(Fully Connected Layers)이란? : 네이버 블로그 (naver.com)

▼ 📎 FNN VS. CNN

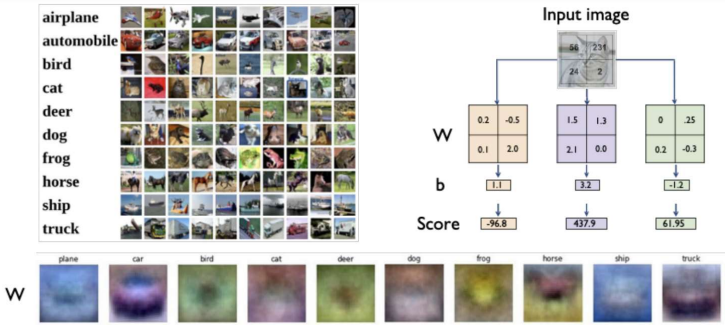
- CNN이 나오기 이전 이미지 인식은 2차원으로 이미지(채널까지 포함 3차원)를 1차원 배열로 바꾼 뒤 **FNN (Fully- connected multi layered Neural Network)** 신경망으로 학습시키는 방법 이었다.
- FNN의 문제점은 인접 픽셀간의 상관관계가 무시된다는 것이다. FNN은 벡터 형태로 표현된 데이터를 입력 받기 때문에 이미지를 반드시 벡터화 해야 한다. 그러나 이미지 데이터는 일반적으로 인접한 픽셀간의 상관관계가 매우 높기 때문에 이미지를 **벡터화 (vectorization)하는 과정에서 정보 손실이 발생한다.**



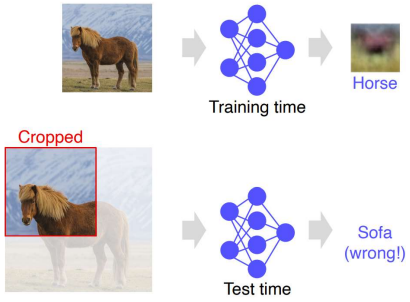
- CNN은 이미지의 형태를 보존하도록 행렬 형태의 데이터를 입력 받기 때문에 이미지를 벡터화 하는 과정에서 발생하는 **정보 손실을 방지**할 수 있다.
- 참고 : CNN (Convolutional Neural Network) 개념 : 네이버 블로그 (naver.com)



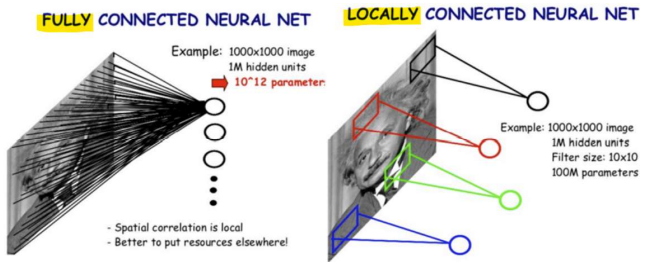
- 문제 1️⃣ 평균 영상(이미지) 외에는 표현 불가능



- 문제 2️⃣ test time 때 문제 발생
  - crop 된 사진이 input으로 들어오면, 이런 패턴은 학습한 적이 없으므로 틀린 output 내놓음



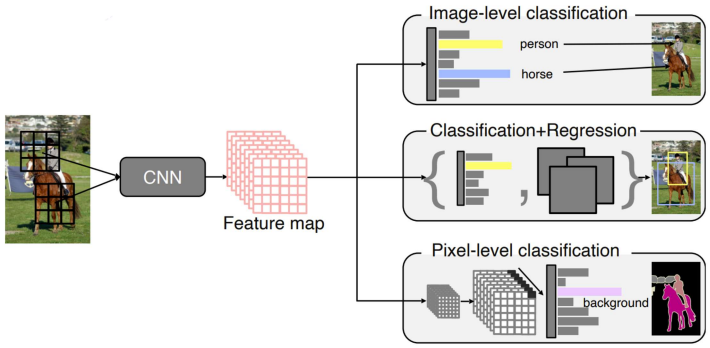
Convolution neural networks are ~~fully~~ `locally` connected neural networks



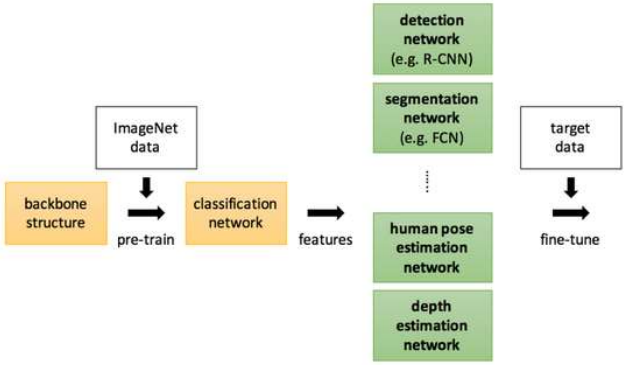
- Local feature learning
  - feature 1개 얻기 위해서
    - fully connected → 패턴 바뀌면 틀릴 확률 ↑
    - locally connected
- Parameter sharing
  - hidden node(filter) 재활용 가능 → parameter 개수 ↓ + overfitting 방지 (∵ parameter 개수 多 이면, overfitting)

backbone으로 사용되는 CNN

- CNN → feature extraction 역할
- target network head
  - image-level classification
  - classification + regression → object detection
    - ▼ 📎 box regression
      - **predicted box**가 **ground truth box**와 유사하도록 학습하는 것
      - 참고 : Bounding box regression (tistory.com)
  - pixel-level classification → segmentation



▼ backbone



- Backbone은 등뼈라는 뜻인데, 즉 척추는 뇌와 몸의 각 부위의 신경을 이어주는 역할을 한다.
- 입력: 뇌를 통해, 출력: 팔, 다리 라고 생각하면 backbone은 입력이 처음 들어와서 출력에 관련된 모듈에 처리된 입력을 보내주는 역할이라고 생각할 수 있다.
- 개체를 검출하든 영역들을 나누든 Neural Network는 **입력 이미지로부터 다양한 feature를 추출**해야 한다. 그 역할을 **backbone 네트워크**가 한다.

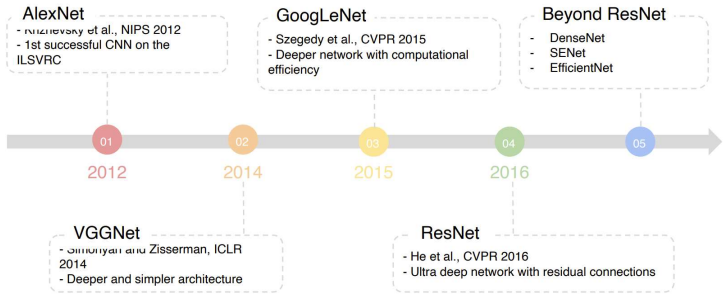
- 참고 : 딥러닝에서 Backbone Network란? : 네이버 블로그 (naver.com)

▼ classification + regression → object detection

- convolution network를 통해 **classification** + box **regression**(localization)을 수행한다
- 참고 : [AI/딥러닝] 진정한 딥러닝을 위한 3가지 분류 (Classification, Object Detection, Image Segmentation) 2탄 (tistory.com)

CNN architectures for image classification 1

History



AlexNet

LeNet-5

- by Yann LeCun
- `Conv` - `Pool` - `Conv` - `Pool` - `FC` - `FC`
- Convolution : 5x5 filters with stride 1
- Pooling : 2x2 max pooling with stride 2

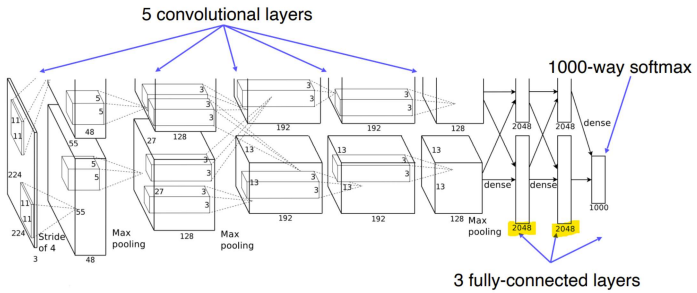
▼ pooling layer

- 1. input size를 줄임(Down Sampling).
  - 텐서의 크기를 줄이는 역할을 한다.

- 2. overfitting을 조절
  - input size가 줄어드는 것은 그만큼 쓸데없는 parameter의 수가 줄어드는 것이라고 생각할 수 있다. 훈련데이터에만 높은 성능을 보이는 과적합(overfitting)을 줄일 수 있다.
- 3. 특징을 잘 뽑아냄.
  - pooling을 했을 때, 특정한 모양을 더 잘 인식할 수 있음. (?)
- 4. 지역적 이동에 노이즈를 줌으로써 일반화 성능을 올려준다.
  - max pooling의 경우 주어진 픽셀중 큰것만 뽑기때문에 모양이 조금 달라지는 특성을 가지고 있다
- 참고 : [CNN] pooling이란? (tf.keras.layers.MaxPool2D) (tistory.com)

AlexNet

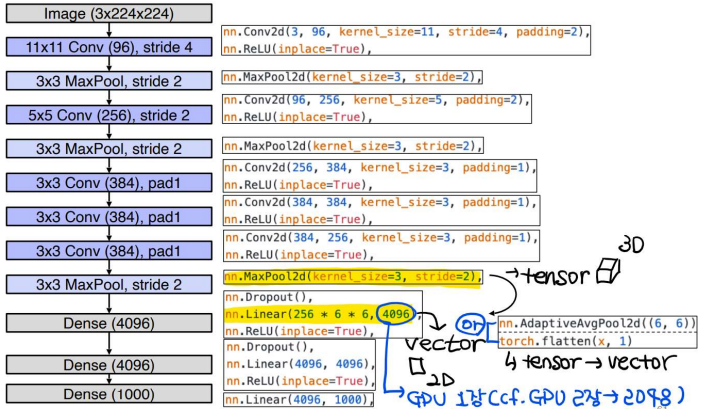
- **LeNet-5와의 차이점**
  - **bigger** (7 hidden layers + 605k neurons + 60 million parameters)
  - trained with **ImageNet**
  - **ReLU** (activation function) + **dropout** (regularization technique)
- **overall architecture**
  - `Conv` - `Pool` - `LRN` - `Conv` - `Pool` - `LRN` - `Conv` - `Conv` - `Conv` - `Pool` - `FC` - `FC` - `FC`



  - (위의 그림) GPU 2장으로 나눠서 사용 (단, LRN 사용 O)
    - 중간중간 activation map cross 일어남

▼ activation map

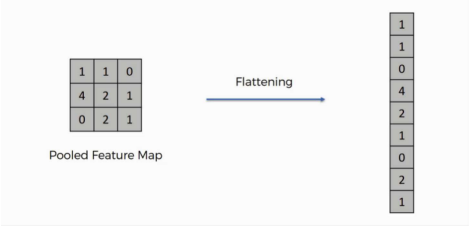
- 하나의 **convolution filter**가 순차적으로 input 데이터를 거치게 되면, 하나의 **map 형태의 결과값**이 나오게 되는데, 이를 feature map이라고 합니다.
- 이때, feature map에서 activation function을 적용한 결과를 activation map이라고 합니다.
- 참고 : 1. Activation Map :: Time Traveler (tistory.com)
- 참고 : [Deep Learning] 헷갈리는 기본 용어 모음집 — Constructing Future (tistory.com)



  - (위의 그림) GPU 1장을 사용 (단, LRN 사용 X)

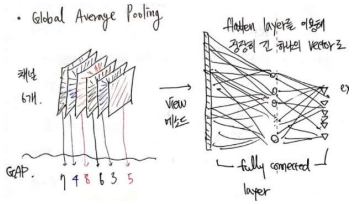
▼ tensor → vector ?

- CNN에서 Convolution Layer와 Pooling Layer를 반복적으로 거치면서 주요 특징만 추출되는데 이 때 추출된 주요 특징은 2차원 데이터로 이루어져 있지만 fc layer(Dense)와 같이 분류를 위한 학습 레이어에서는 1차원 데이터로 바꾸어서 학습이 되어야 한다. **이때 Flatten Layer가 2차원 데이터를 1차원 데이터로 바꾸는 역할을 한다.**



- 참고 : Tensorflow 2.0 - 각 Layer별 역할 개념 및 파라미터 파악 (tistory.com)

▼ 📎 average pooling VS. flatten ?



- `Flatten Layer`를 이용해서 입력받은 값을 굉장히 긴 하나의 벡터로 만든 다음, 그 벡터를 FC Layer에 넣는 방식으로 하나하나 매핑해서 클래스를 분류했습니다. 이 과정에서 **공간적 정보도 많이 잃어**버리는데다가, 굉장히 많은 파라미터, 즉 **가중치가 많이 필요**하고, VGGNet의 경우 이 부분이 전체 계산량의 85%를 차지했습니다. 컨볼루션 레이어를 아무리 쌓아도 FC Layer 하나를 못 따라갈 정도입니다.
- `Global Average Pooling layer`는 **분류할 클래스 수만큼 feature map을 마지막에 생성**합니다. 위 그림에선 feature map이 6개니까 분류할 클래스 수가 6개라고 가정합시다. 그럼 그 feature map 안에 있는 특징값들의 평균을 구해서 각각의 출력 노드에 바로 입력하는 방식입니다. 위 그림에서는 각 feature map의 평균이 7, 4, 8, 6, 3, 5가 나왔습니다.
  - 즉, 단순히 **i번째 feature map의 평균값을 구해서 i번째 출력 노드에 입력하는 것**
    입니다.
- GAP의 장점

  1. Location 정보를 FC Layer보다 적게 잃는다.

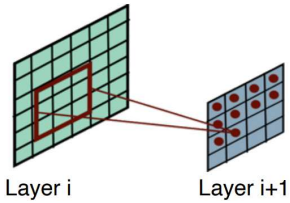
  2. 파라미터를 차지하지 않아 계산 속도가 빠르다.

  3. 파라미터가 많아지지 않기 때문에 오버피팅을 방지한다.

  4. feature map 안의 값들의 평균을 사용하기 때문에 global context 정보를 가진다.

- 참고 : GAP (Global Average Pooling) : 전역 평균 풀링 (tistory.com)

- **deprecated components**
  - 1️⃣ Local Response Normalization (LRN) → Batch normalization 사용
    - ▼ 📎 LRN (local response normalization)
      - `ReLU`는 양수의 방향으로는 입력의 값을 그대로 사용합니다. 그렇게 되면, Conv나 Pooling 시 매우 높은 하나의 픽셀값이 주변의 픽셀에 영향을 미치게 됩니다. → overfitting 가능 (∵ training data에만 feature가 크게 반응했으므로)
      - 이런 부분을 방지(= 한 filter에서만 과도하게 activate 되는 것을 방지)하기 위해서 다른 activation map의 같은 위치에 있는 픽셀끼리 `normalization`을 진행합니다.
      - 참고 : LRN(Local Response Normalization) 이란 무엇인가?(feat. AlexNet) :: Taegu (tistory.com)
      - 참고 : AlexNet: ImageNet Classification wtih Deep Convolutional Neural Networks Curaai00's Deep Learning Blog (tistory.com)
      - 읽어보기 : AlexNet (tistory.com)
  - 2️⃣ 11x11 convolution filter → 크기가 큰 filter 사용 X
    - The filter size ↑ → the input size of the image ↑
      - LeNet: 28x28
      - AlexNet: 227x227
    - Larger size **filters** are used to cover **a wider range of the input image**
  - 3️⃣ Receptive field in CNN
    - (가정) KxK Conv + stride 1 + PxP pooling layer
      - → input size = (P+K-1) X (P+K-1)
      - 일반적인 conv output layer 구하는 공식과 동일
      - (아래 그림) Layer i (input layer) ↔ Layer i+1 (pooling layer)



▼ 📎 Receptive field

- Receptive Field는 **filter가 한번에 보는 영역**으로 생각하면 됩니다. 일반적인 3x3 filter size는 receptive field가 3x3 이미지 입니다. 그리고 이러한 layer를 두개, 세개 쌓으면 receptive filed가 5x5, 7x7 이런식으로 늘어나게 됩니다. Receptive field가 늘어난다는 것은 output을 계산할때 사용하는 정보의 양이 많다는 것 입니다.
- 정보의 양이 늘어나면, 성능이 좋아질 확률도 높아지지만, 학습해야 할 양이 많아서 연산량이 증가하게 되는 단점도 있습니다. 이 **Receptive field를 높이기 위해서 filter의 크기를 키우거나, layer를 늘릴 수 있습니다. 또는 pooling 등을 사용하는 것도 receptive field를 높일 수 있습니다.** Poling의 경우 연산량 까지 감소할 수 있지만 정보의 손실을 가져올 수도 있죠.
- 참고 : https://dataplay.tistory.com/29

▼ 📎 output layer size 수식

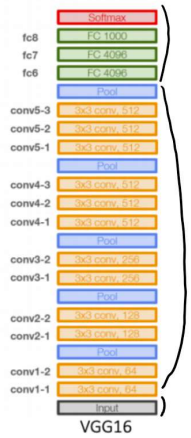
$$(OH, OW) = \left(\frac{H + 2P - FH}{S} + 1, \frac{W + 2P - FW}{S} + 1\right)$$

- (H, W) : 입력 크기 (input size)
- (FH, FW) : 필터 크기 (filter/kernel size)
- S : 스트라이드 (stride)
- P : 패딩 (padding)
- (OH, OW) : 출력 크기 (output size)

- 참고 : 06. 합성곱 신경망 - Convolutional Neural Networks (tistory.com)

VGGNet

- Deeper architecture → 16, 19 layers
- Simpler architecture
  - LRN X
  - 3x3 Convolution + 2x2 max pooling (11x11 Convolution X)
- Better performance
  - AlexNet 보다 성능이 좋음 (2nd in ILSVRC14)
- Better generalization
  - 미리 학습된 중간 feature를 다른 task에 적용가능하도록 만듦 (?) ⇒ generalization ↑
- overall architecture



VGG16

`output`

- 3 fully-connected (FC) layers

`Key design choices`

- 3x3 convolution filters with stride 1
- 2x2 max pooling operations

⇒ Using **many** **3x3 conv layers** instead of a **small** number of **larger conv filters**

- Keeping receptive field sizes large enough (∵ 작은 kernel size의 conv layer도 많이 쌓으면, receptive field 크기 大)
- Deeper with more non-linearities
- Fewer parameters

`Input`

- 224x224 RGB images (same with AlexNet)
- Subtracting mean RGB values of training images ⇒ normalize

▼ 📎 subtracting RGB mean 하는 이유

- **Mean subtraction**은 가장 일반적인 전처리 형식이다. 이 방법은 데이터의 모든 개별*피쳐*에 **그 평균을 빼는 것**이다. 이에 대한 기하학적 해석은 데이터의 **모든 차원에 대한 분포의 중심을 원점으로 이동**하는 것이다.
- **Unnormalized**의 경우, 앞뒤로 왔다 갔다 하면서 수많은 단계를 거쳐 최적값에 도달하게 된다. 또한, 학습률(Learning Rate)를 작게 설정 해야 한다.
- 반면에, **Normalied**의 경우 어디서 시작하든 쉽게 최적값에 도달할 수 있으며, 학습률을 상대적으로 높여서 사용할 수 있게 때문에 **빠르게 훈련시킬 수 있다.**
- 참고 : cs231n: Setting up the data and the model (tistory.com)

`Other details`

- **ReLU** for non-linearity
- No local response normalization (LRN X)

Reference

Course overview

- Thompson, Margaret Thatcher: A New Illusion, Perception 1980
- Kirillov et al., Panoptic Segmentation, CVPR 2019
- Gordon et al., Depth from Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras, ICCV 2019
- Huang et al., Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization, ICCV 2017
- Selvaraju et al., Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, ICCV 2017

CNN architectures for image classification 1

- Lecun et al., Gradient-based Learning Applied to Document Recognition, Proceedings of the IEEE 1998
- Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
- Simonyan and Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015