

# CUAI 딥러닝 논문 리뷰 스터디 CV 1팀

2025.03.25

발표자 : 박준우

## 스터디원 소개 및 만남 인증



스터디원 1 : 박준우

스터디원 2 : 김태환

스터디원 3 : 송재호

스터디원 4 : 이해찬

스터디원 5 : 김성민

목차

스터디 소개

2주차 스터디 내용 소개

다음 스터디 계획

## 스터디 소개 (1)

주차	논문
1주차	1. ImageNet Classification with Deep Convolutional Neural Networks (AlexNet) 2. Going deeper with convolutions (Inception-v1)
2주차	3. Rethinking the Inception Architecture for Computer Vision (Inception-v2~3) 4. Deep Residual Learning for Image Recognition (ResNet)
3주차	5. Squeeze and Excitation Network (SENet) 6. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
4주차	7. You Only Look Once: Unified, Real-Time Object Detection 8. Generative Adversarial Networks (GAN)
5주차	9. Rich feature hierarchies for accurate object detection and semantic segmentation 10. Attention Is All You Need

- CV 분야에서 foundational한 논문들 10편 선정하고 매주 두 편씩 읽음
- 두 명씩 조를 짜 각 조가 발표할 논문을 정함
- (김태환님은 따로 리뷰할 논문 준비/설명)

## 스터디 소개 (2)

### 논문 리뷰

☰ 박준우 ☑ 이해찬 ☑ 송재호 ☑ 김성민 ☑ 김태환 ☑ 전체 +

☰ ↓ ↶ 🔍 ... 새로 만들기 ▾

👤 이름	Aa 논문 제목	🕒 주차	+ ...
P Park Junu	📄 ImageNet Classification with Deep Convolutional Neural Networks	1주차	
P Park Junu	📄 Going Deeper with Convolutions	1주차	
👤 김태환	📄 Personalize Anything for Free with Diffusion Transformer	1주차	
👤 송재호	📄 Going Deeper with Convolutions	1주차	
👤 송재호	📄 ImageNet Classification with Deep Convolutional Neural Networks	1주차	
👤 성민	📄 ImageNet Classification with Deep Convolutional Neural Networks %	1주차	
H Haechan Lee	📄 Going deeper with convolutions	1주차	
H Haechan Lee	📄 ImageNet Classification with Deep Convolutional Neural Networks	1주차	
+ 새 페이지			

- 각자 논문을 읽고 노선에 정리 및 공유
- 각 조당 발표자는 한 명씩 당일에 랜덤으로 정함

## 2주차 스터디 내용 소개

김성민, 이해찬: **ImageNet Classification with Deep Convolutional Neural Networks (AlexNet)**

박준우, 송재호: **Going Deeper with Convolutions (GoogLeNet)**

김태환: **Personalize Anything for Free with Diffusion Transformer**

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

ILSVRC-2014에서 우승을 한 **GoogLeNet**에 관한 논문

AlexNet (ILSVRC-2012) 이후로 Deep Convolutional Layer에 대한 가능성을 확인할 수 있었음. (층을 더 깊게 쌓으면 더 복잡한 특징 학습 가능)

### 3 Motivation and High Level Considerations

The most straightforward way of improving the performance of deep neural networks is by increasing their size. This includes both increasing the depth – the number of levels – of the network and its width: the number of units at each level. This is as an easy and safe way of training higher quality models, especially given the availability of a large amount of labeled training data. However this

“그럼 층을 어떻게 더 깊게 만들 수 있을까?” 라는 질문에서 출발

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Problem 1: 층을 넓고 깊게 만들 때 생기는 문제

1) 파라미터의 수가 늘어나면서 **overfitting**의 문제가 생김

+ 한정된 데이터: ILSVRC는 아주 세밀하게 분류된 데이터

(현실적으로 구하기 어려움)



(a) Siberian husky



(b) Eskimo dog

2) 컴퓨팅 자원이 기하급수적으로 늘어나게 되고 낭비됨

두 개의 연속된 Convolution Layer에서 filter 수를 균일하게 증가시키면 계산량은 제곱으로 증가함

(Ex. 두 Layer의 filter 수를 각각 두 배로 늘리면 계산량은 4배( $2^2$ )로 증가)

만약 추가된 용량이 비효율적으로 사용된다면(Ex. 대부분의 가중치가 거의 0에 가까움), 상당한 컴퓨팅 자원이 낭비됨



## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Solution 1. Use Sparsely connected architecture

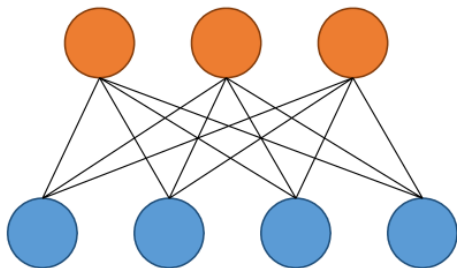
Fully(Densely) Connected: 모든 뉴런이 서로 연결됨 = 많은 수의 파라미터 필요

Sparsely Connected: 일부 뉴런들만 서로 연결됨 = 더 적은 수의 파라미터 필요

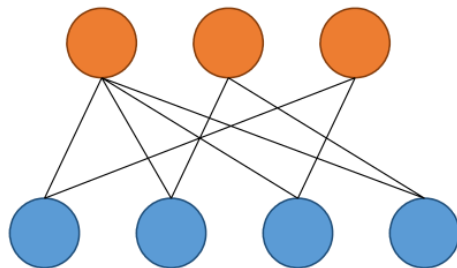
-> 과적합, 컴퓨팅 자원 해결

Convolution Layer, FC Layer 모두 sparsity를 구현하여 훨씬 적은 수의 파라미터를 사용하도록 architecture를 변경함.

Densely connected



Sparsely connected



## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### **Problem 2: 현재 컴퓨팅 디바이스(CPU, GPU)는 dense matrix operation(밀집 행렬 연산)에 최적화돼 있음.**

현대 CPU, GPU는 규칙적인 패턴의 대규모 행렬 연산에 최적화 돼 있음. 그러나 Sparse 행렬은 0이 아닌 값들이 불규칙하게 분포해 있음.

대표적으로

#### **1. Overhead of Lookups**

Sparse Matrices는 연산할 때마다 값이 "0인지 아닌지" "어디에 있는지"를 확인하는 과정이 필요한데 조회 과정이 실제 연산보다 더 많은 시간을 소모할 수 있음.

#### **2. Cache Misses**

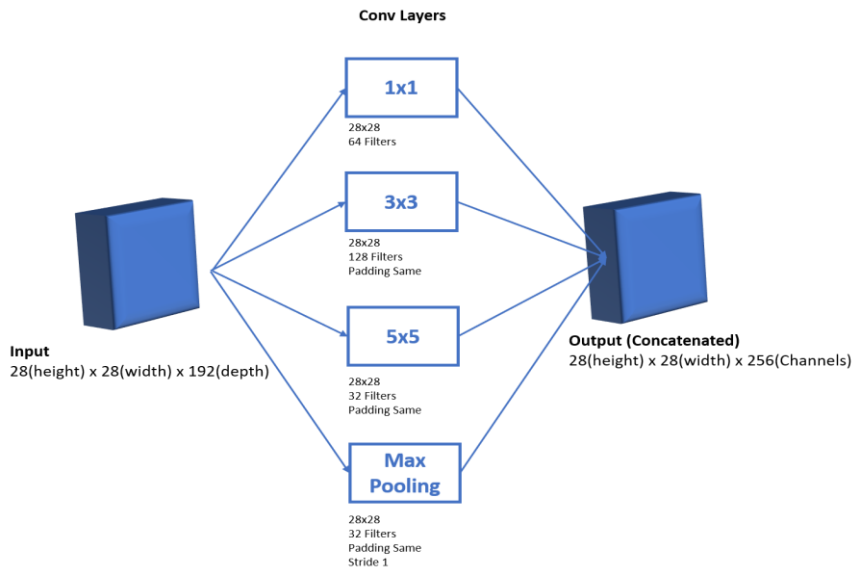
Dense Matrices와 달리 Sparse Matrices는 데이터가 불연속적으로 저장돼 캐시 효율이 낮고 메모리 접근 시간이 증가함.

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Solution 2. Inception Module

Idea: 하드웨어의 Dense matrix 연산의 효율성과 sparse connectivity의 이점을 동시에 얻어보자

다양한 filter size와 pooling을 병렬적으로 적용하고 결과를 합침(Concatenate)



## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Solution 2. Inception Module

**Sparse architecture:** 1x1, 3x3, 5x5 필터, 그리고 풀링(Pooling)을 통해서만 입력과 출력이 연결

이때

- 1x1: 주로 채널 간 정보를 혼합하고 채널의 수 축소
- 3x3: 작은 영역 특징 추출
- 5x5: 상대적으로 큰 영역 특징 추출

**Dense matrix:**

각각의 filter 내부는 밀집 행렬

따라서 filter를 통해 계산할 때 밀집 행렬 곱셈(dense matrix calculation)을 수행할 수 있음.

Dense matrix는 GPU에 최적화돼 있기 때문에 계산 효율적임

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Problem 3: Inception Module을 그냥 사용하니 (Naïve version) 연산량 매우 증가함

1. filter 크기가 클수록(3x3, 5x5) 연산량이 제곱으로 늘어남
2. 입력 채널 수가 클수록 (모든 채널에 filter를 적용해야 되기 때문에) 연산량이 증가함

Ex. 입력 채널이 256, 출력 채널이 256일 때 파라미터 수 =  $5 \times 5 \times 256 \times 256$  = 약 1,600,000  
(매우 커짐)

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Solution 3. 1x1 convolution layer를 사용해 연산량을 줄여보자.

3x3, 5x5 와 같은 kernel 을 사용할 때는 먼저 1x1 convolution layer를 통과하게 하게 함  
채널 수가 감소하기 때문에 파라미터 수가 현저히 감소할 뿐더러  
성능 저하 없이 오히려 모델의 표현력이 향상됨 (1x1 convolution layer + ReLU)

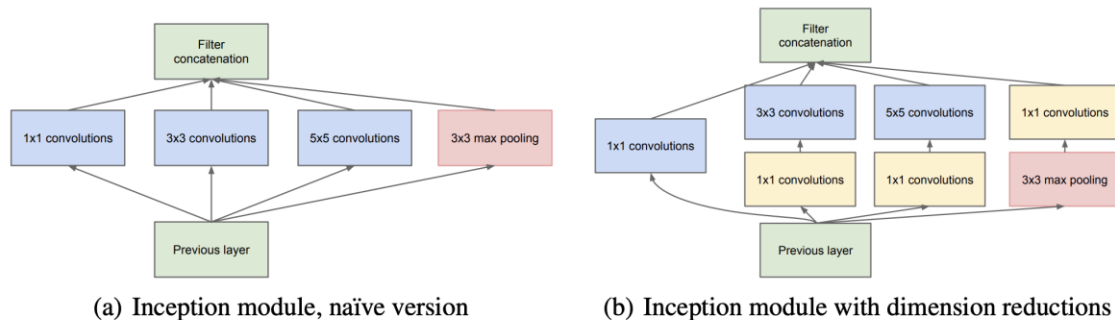


Figure 2: Inception module

## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### GoogLeNet with Inception Module

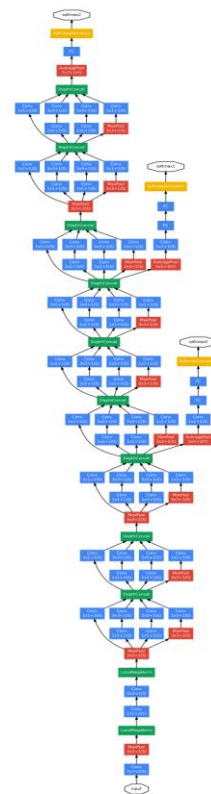


Figure 3: GoogLeNet network with all the bells and whistles

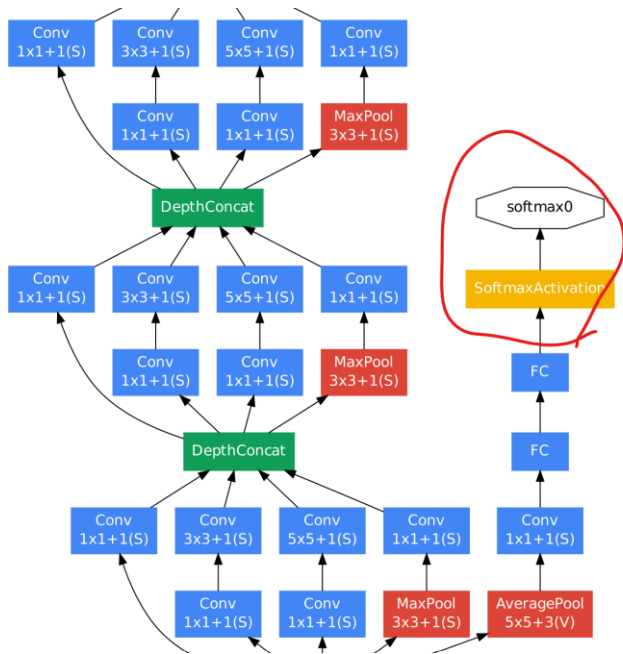
## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### Auxiliary(보조적인) Classifier

네트워크의 깊이가 긴 만큼 기울기 소실  
(Vanishing Gradient) 문제가 있을 수 있음.

VG: 학습 과정에서 기울기가 점점 작아져 앞쪽  
레이어의 가중치가 제대로 업데이트되지 않는  
현상

보조 분류기를 도입하여 중간 단계에서도  
손실을 계산하고 기울기 생성. 기울기가 네트워크  
의 앞까지 더 잘 전달되도록 도움으로써  
VG 문제 완화





## 2주차 스터디 내용 소개: Going Deeper with Convolutions

### **Others**

- Multibox
- Ensemble
- Global Average Pooling
- ...

## 다음 스터디 계획

1. Rethinking the Inception Architecture for Computer Vision (Inception-v2, Inception-v3)
2. Deep Residual Learning for Image Recognition (ResNet)

감사합니다