Going Deeper with Convolutions

Author	Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna
Field	CV
Journal	CVPR
# Published Year	2014
Speaker	강동규 이민지
Summary	InceptionNet-v1
status	Finished!
link	https://arxiv.org/pdf/1409.4842.pdf

▼ Target Problem

> 📌 By increasing the depth and width efficiently, produce high accuracy

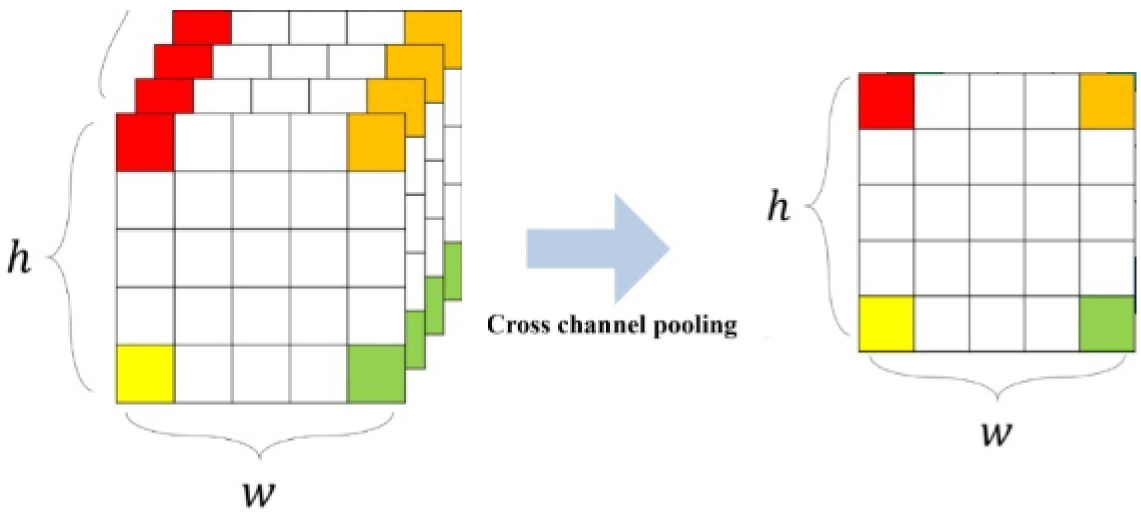
- Good Model → Efficient Model → Good at **performance** and **memory usage**
- Performance
    - Increase DEPTH and WIDTH

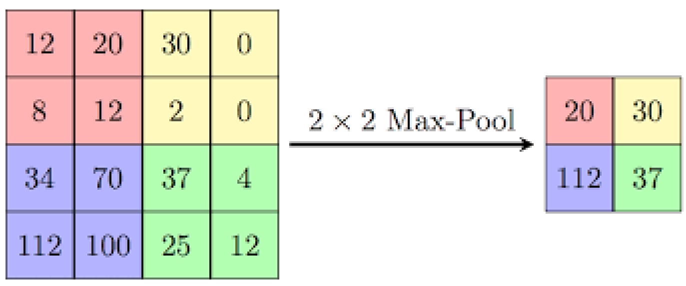
▼ Background & Related Work

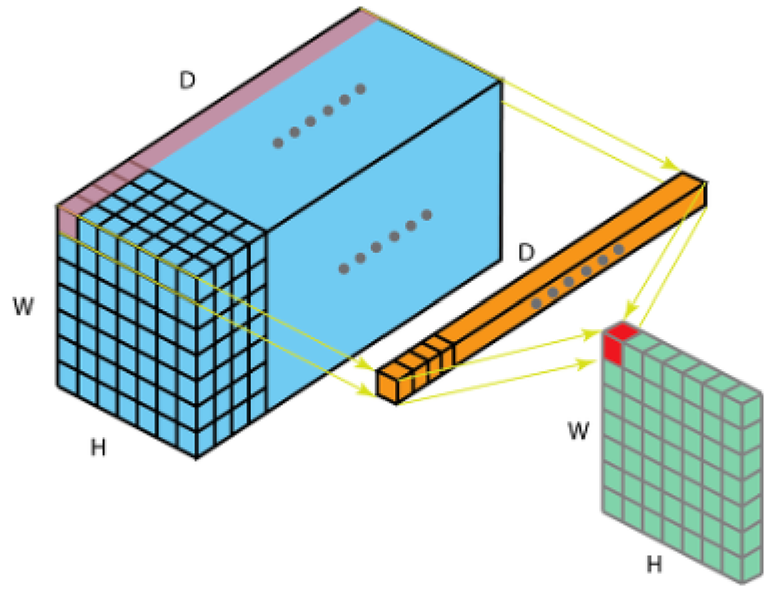
Network In Network (NIN)

1x1 convolution

what is CCCP(cascaded cross channel parametric pooling)?





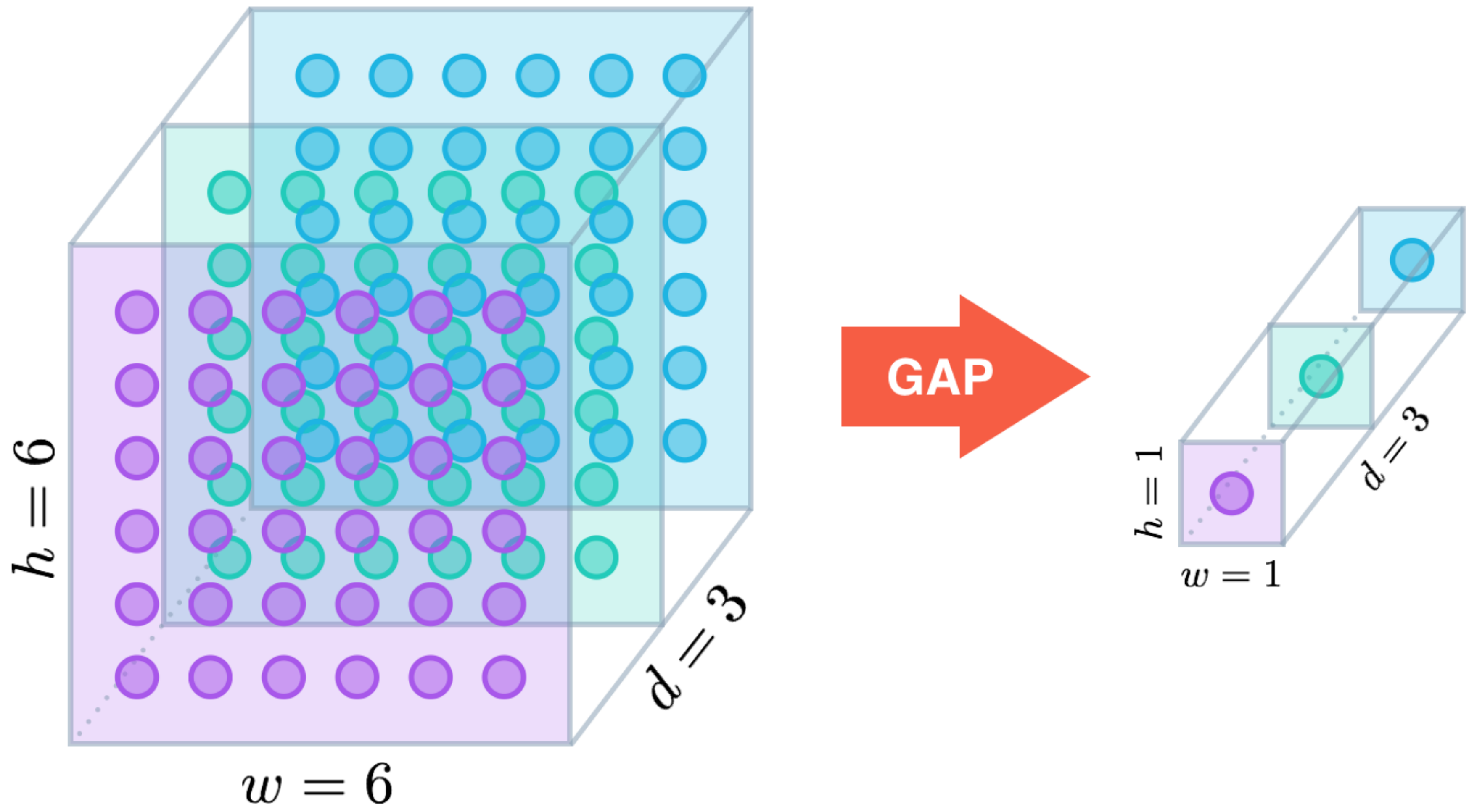


- CCCP
  - Max pooling
  - Reduce channels

⇒ CCCP is equal to 1x1 convolution kernel!

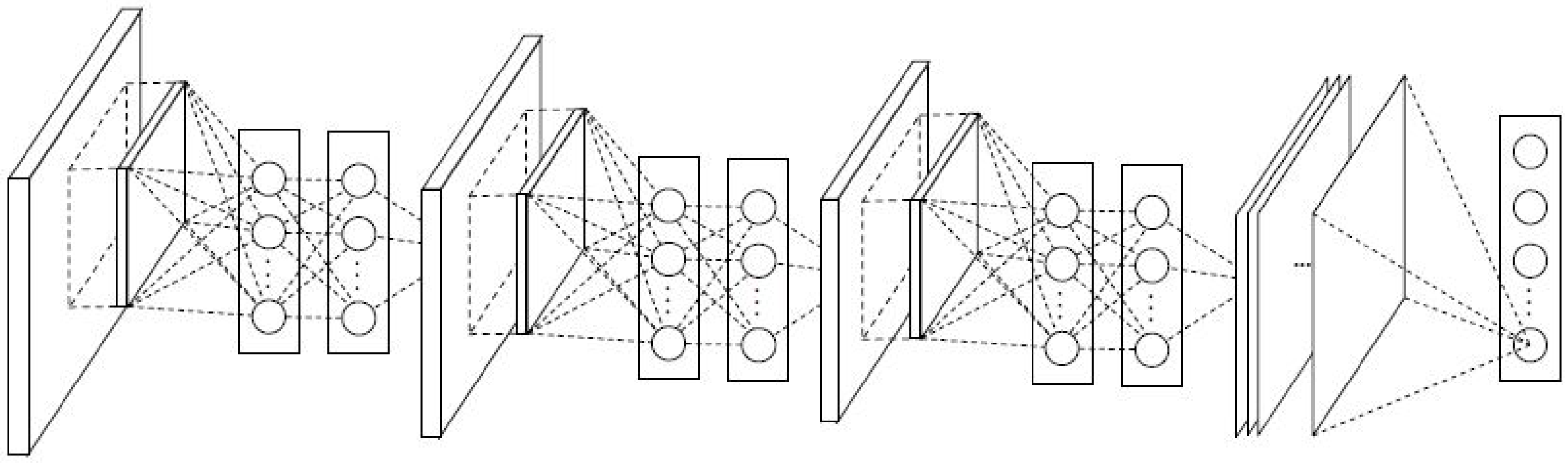
This derived 1X1 convolution is later used in GoogLeNet and several papers due to the effect of channel reduction.

Global average pooling



- Global average pooling
  - Calculate the average of each feature map
  - Each feature map represent the characteristics
  - Don't have parameters
- FC layer
  - All information in all feature map is connected
  - Difficult to know why each class was selected

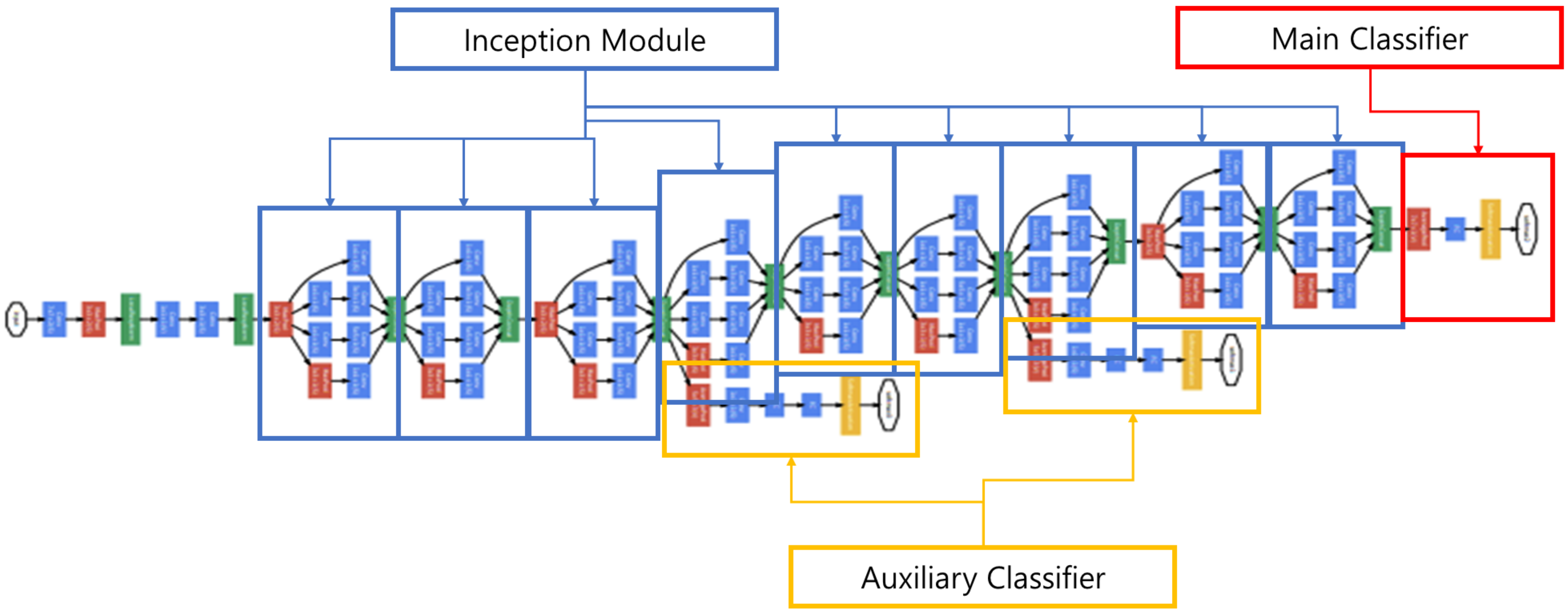
Network in network structure



NIN = mlpconv layer 3 + global average pooling layer 1

▼ GoogLeNet

Overall Architecture



Inception

1. Motivation

- Increasing depth and width ⇒ Bigger size network ⇒ Training higher quality
  - Depth = The number of levels
  - Width = The number of units

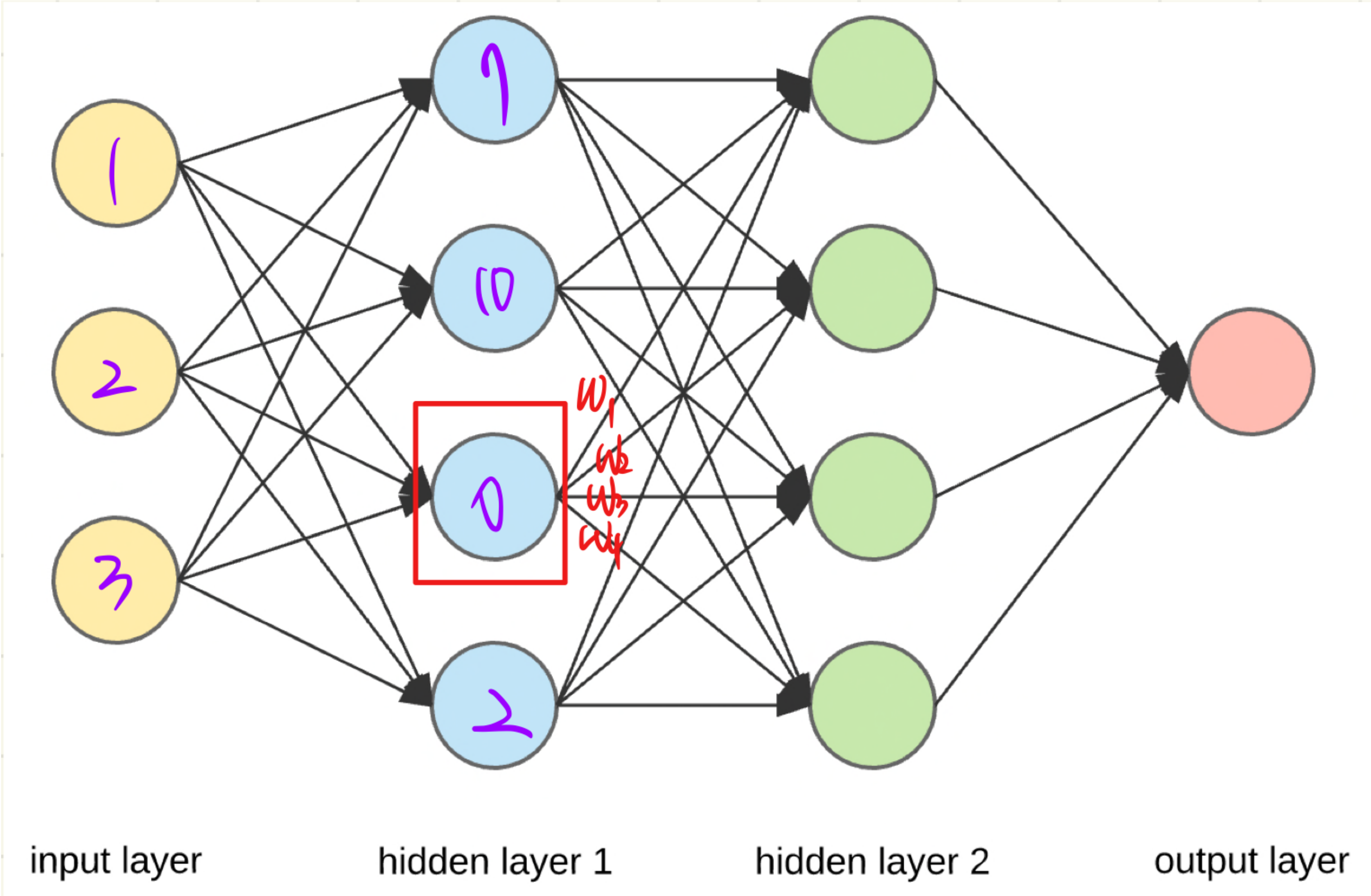
Problem

1. A larger number of parameters ⇒ Overfitting
2. Computational resources dramatically increase.

Solution

- Change fully connected to **sparsely** connected architecture

- Sparsely Connected Architecture
  - Dense Layer(Fully Connected Layer)



Reference : https://iq.opengenus.org/dense-layer-in-tensorflow/

  - Neurons are not necessarily connected each other.
  - Some neurons would be activated, the others would not.

  - Convolution

1.2	4.2	6.2
0.1	10	0.2
6.6	9.7	0.1

Conv weight

1.2	0.1	0.2	4.8	7.2
0.1	10	0.2	0.2	0.1
0.1	9.7	0.1	0.2	0.1
0.2	7.3	0.2	0.3	0.2
0.2	0.2	0.1	0.2	0

Conv weight(Add capacity)

  - When network capacity is added and if most weights end up to be close to zero, **a lot of computation is wasted.**

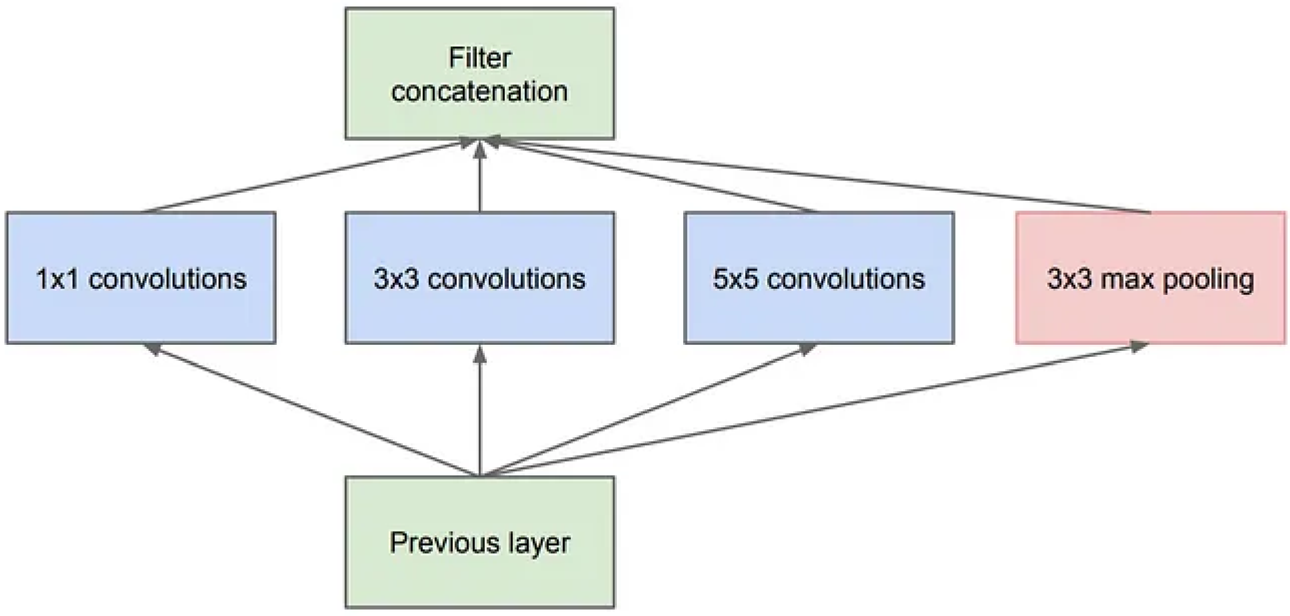
  - Sparse architecture ⇒ Parameters decrease.

- Because of GPU calculation, implementation is hard, inefficient.

⇒ Approximate a sparse structure using inception architecture

2. Naive Inception Module

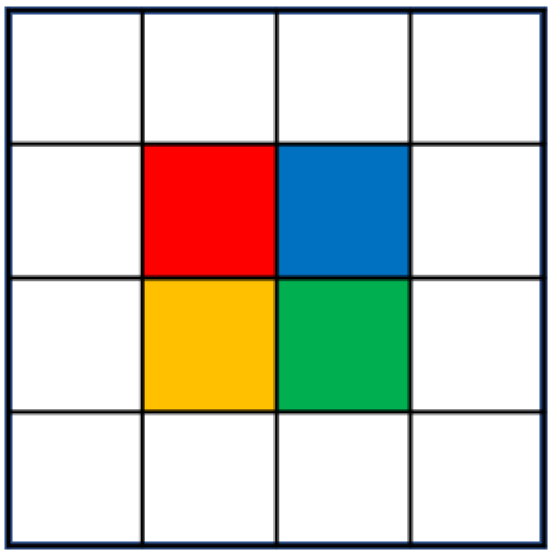
📌 Let's find the optimal local construction and repeat it spatially!



(a) Inception module, naïve version

- Why use different convolution sizes?
  1. High-level features are abstracted, so their spatial concentration is reduced.
     - Using 3x3, 5x5 convolutions, increase spatial concentration.
  2. Obtain various features by using convolutions of multiple scales.

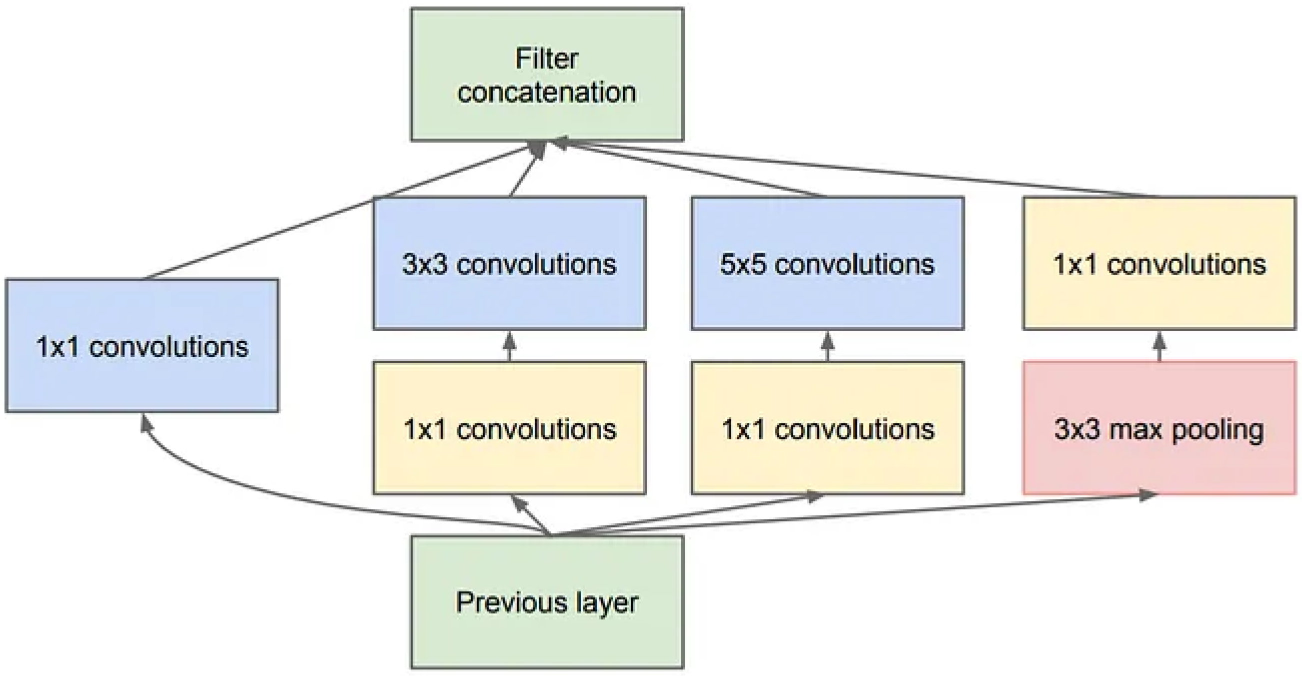
- Why use odd scales? ⇒ To avoid patch-alignment issues



⇒ The cost would increase dramatically when the number of filters is increased

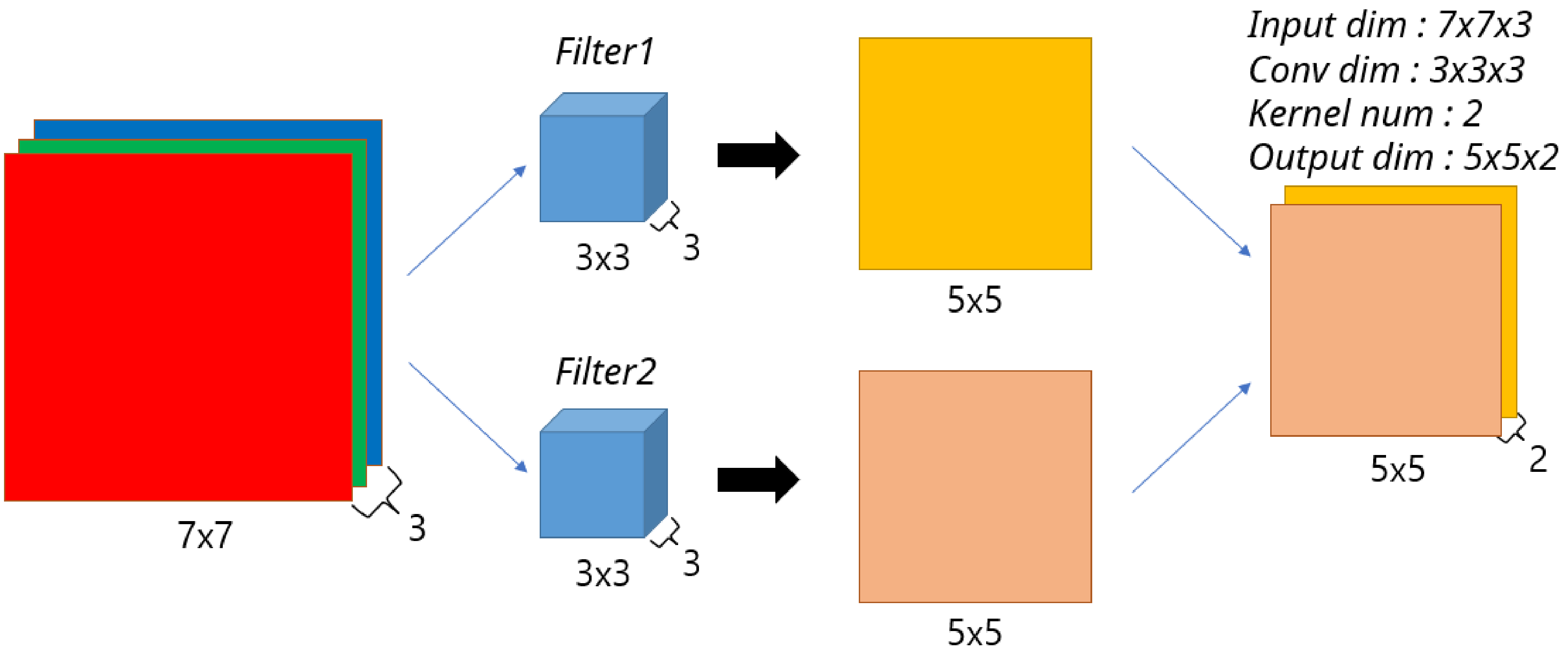
3. Inception Module with Dimension Reduction

📌 Reduce dimension and the number of parameters using 1x1 convolutions!

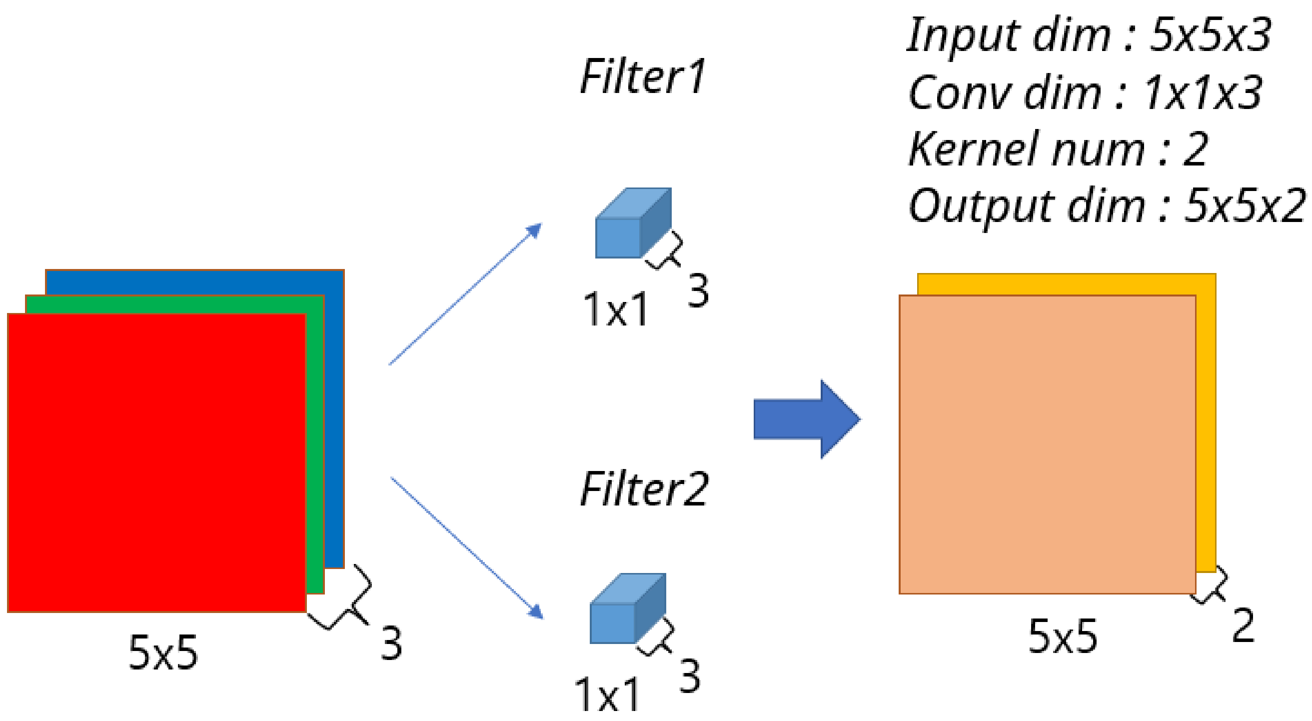


(b) Inception module with dimension reductions

cf) Understanding calculation of convolution channel



- Use 1x1 convolutions - Dimension Reduction



- 1x1 convolutions
    - Compress channels ⇒ Reduce dimensions ⇒ Decrease computational cost.
    - Compressed channels ⇒ Compressed informations.
    - Sparsity

4. Benefit of Inception Module

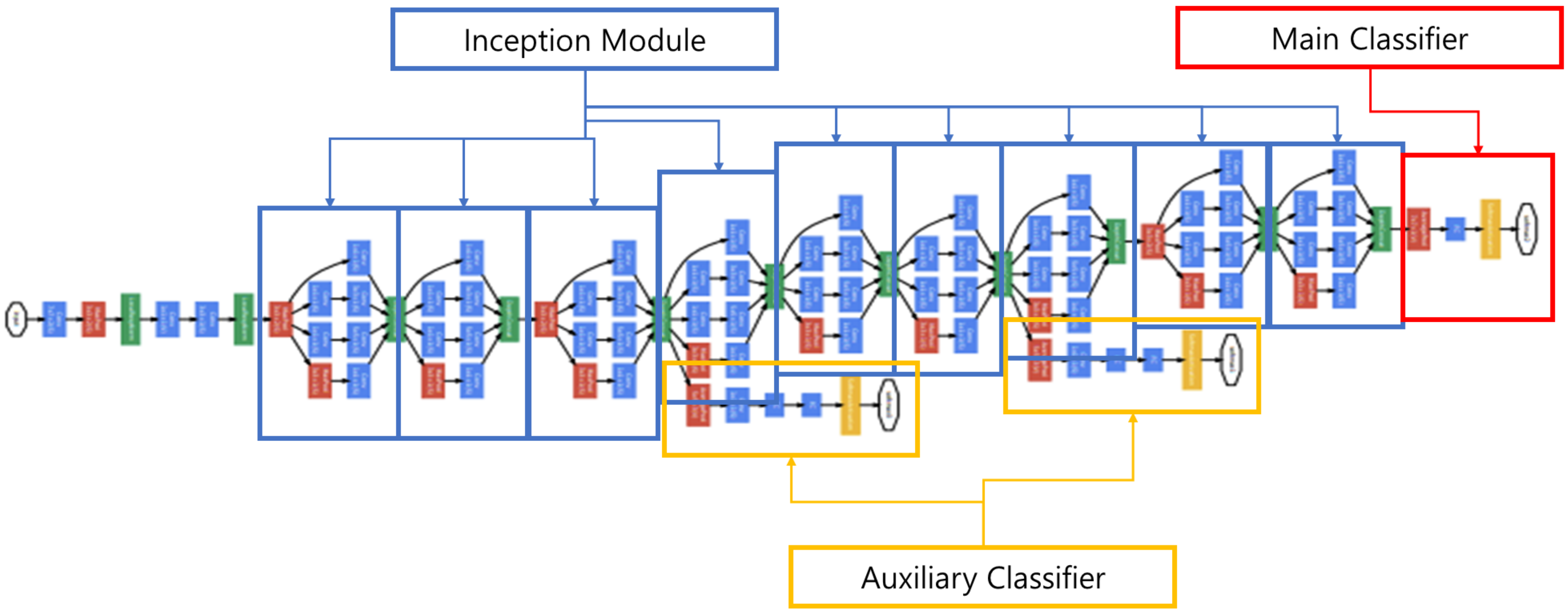
1. It uses computational resources efficiently ⇒ Better Computational cost, Speed
2. It **increases depth and width** without dramatically increasing of cost.
3. It allows to gain **intuition of visual information** and **abstract features** from different scales simultaneously.

Auxiliary Classifier

📌 Solve vanishing gradient problem using auxiliary classifier!

- Deep Network → backpropagation would not be linked to the first layer of network.
- Auxiliary classifier
    1. Make propagation flow well
    2. regularization effect

Overall Architecture



▼ Result & Conclusion

Classification

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Table 3: GoogLeNet classification performance break down

Detection

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Table 4: Detection performance

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

Table 5: Single model performance for detection

Conclusion

1. The expected optimal sparse structure can be approximated using an easily available dense building block through the inception module.
2. Significant performance improvements were achieved despite the increased computational cost.

▼ Reference

Inception-v1 번역, 분석 참고 : https://sike6054.github.io/blog/paper/second-post/

Inception-v1 논문 : https://arxiv.org/abs/1409.4842