

Architectures de réseaux de traitement d'images

- Comprendre l'architecture des réseaux les plus efficaces de la littérature

- Réseau qui a révélé le Deep Learning [?]
- Vainqueur d'ILSVRC (ImageNet Large Scale Visual Recognition Challenge) en 2012
- Taux d'erreur : 25.8% en 2011 → 16.4% en 2012
- Entraîné sur deux GPUs
- Connections entre couches compliquées en conséquence

Type	Nombre	Taille	Stride	Padding
Input	—	—	—	—
Convolution	96	11×11	4	0
Max Pooling	—	3×3	2	
Normalisation	—	—	—	—
Convolution	256	5×5	1	2
Max Pooling	—	3×3	2	
Normalisation	—	—	—	—
Convolution	384	3×3	1	1
Convolution	384	3×3	1	1
Convolution	256	3×3	1	1
Max Pooling	—	3×3	2	
Linear	4096	—	—	—
Linear	4096	—	—	—
Linear	1000	—	—	—

Exercice

Type	Nombre	Taille	Stride	Padding
Input	—	—	—	—
Convolution	96	11×11	4	0
Max Pooling	—	3×3	2	
Normalisation	—	—	—	—
Convolution	256	5×5	1	2
Max Pooling	—	3×3	2	
Normalisation	—	—	—	—
Convolution	384	3×3	1	1
Convolution	384	3×3	1	1
Convolution	256	3×3	1	1
Max Pooling	—	3×3	2	
Linear	4096	—	—	—
Linear	4096	—	—	—
Linear	1000	—	—	—

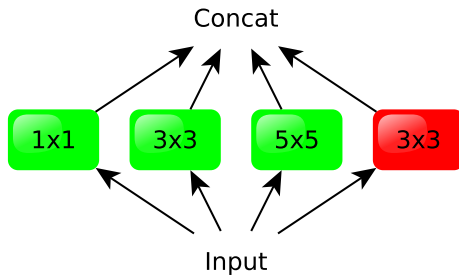
Les couches linéaires. Les couches de convolutions nécessitent peu de paramètres mais plus d'opérations par paramètre.

- Vainqueur d'ILSVRC 2014 (localisation) [?]
- Plus profond
- Filtres plus petits

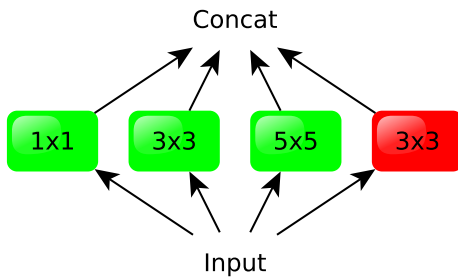
→ Calculs plus simples, plus hiérarchisés

Type	Nombre	Taille	Stride	Padding
Input	—	—	—	—
Convolution * 2	64	3×3	1	1
Max Pooling	—	2×2	2	0
Convolution * 2	128	3×3	1	1
Max Pooling	—	2×2	2	0
Convolution * 2	256	3×3	1	1
Max Pooling	—	2×2	2	0
Convolution * 3	512	3×3	1	1
Max Pooling	—	2×2	2	0
Convolution * 3	512	3×3	1	1
Max Pooling	—	2×2	2	0
Linear	4096	—	—	—
Linear	4096	—	—	—
Linear	1000	—	—	—

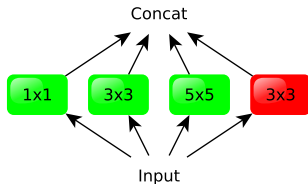
- Vainqueur d'ILSVRC 2014 (classification) [?]
- Plus profond
- Introduction d'un bloc astucieux



Exercice



Quels paddings pour les différents blocs ?



Quels paddings pour les différents blocs ?

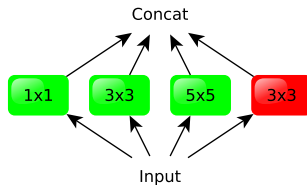
Il faut maintenir des dimensions stables pour pouvoir concaténer :

$$1 \times 1 \rightarrow 0$$

$$3 \times 3 \rightarrow 1$$

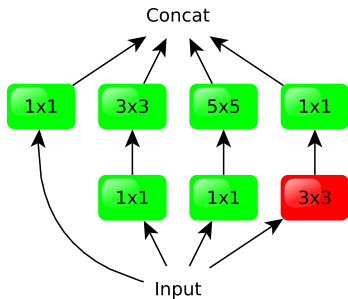
$$5 \times 5 \rightarrow 2$$

Problème du bloc Inception



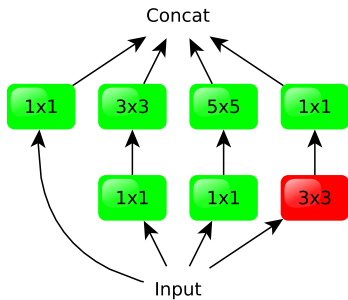
Les profondeurs deviennent prohibitives avec le nombre de couches

Bloc Inception « Bottleneck »



Rajouter des convolutions 1×1 pour contrôler la profondeur.

Exercice



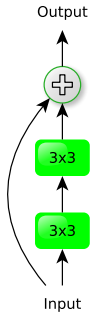
Donner un exemple de contrôle de la profondeur par convolutions 1×1

- Principalement 9 blocs Inception empilés
- Un « petit » réseau classique pour l'input
- 3 « petits » réseaux de prédiction aux blocs 3, 6 et 9

- Principalement 9 blocs Inception empilés
- Un « petit » réseau classique pour l'input
- 3 « petits » réseaux de prédiction aux blocs 3, 6 et 9

Pourquoi pas seulement en couche finale ?

- Microsoft Research [?]
- Vainqueur d'ILSVRC 2015
- **Beaucoup, beaucoup** plus profond (jusqu'à **1000** couches)
- Utilisation de connexions résiduelles
- Très peu de pooling



Intuition : le problème d'optimisation classique est trop dur à résoudre quand il y a beaucoup de couches.



Existe aussi avec un « bottleneck » pour améliorer les performances des réseaux les plus profonds.

Comparaison précise des temps de calculs, performance dans l'étude [?].

