

1

1. Limite des ANN pour la classification des images
2. Construction et apprentissage d'un CNN
3. Visualisation d'un CNN

Rappel

2

Ce qu'on va voir

Contenu

1. Exemples d'architectures CNN
2. Data Augmentation
3. Transfer Learning
4. Generative Adversial Networks

3

Quelques architectures CNN

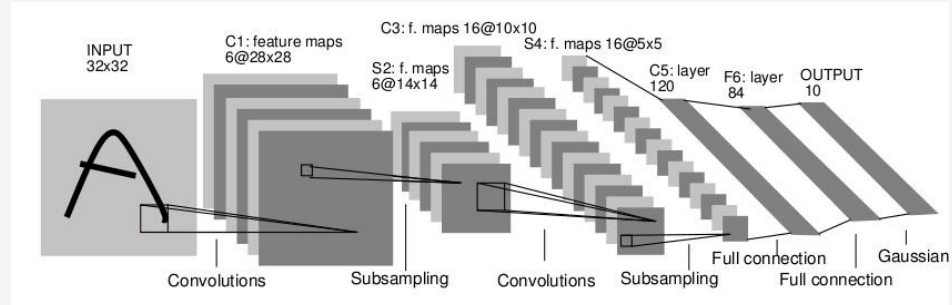
- LeNet (1998)
- AlexNet (2012)
- VGG-16 (2014)
- GoogleNet (2015)
- ResNet (2015)
- DenseNet(2016)

4

LeNet (1998)

- Architecture:

- Conv
- Pool
- Conv
- Pool
- FC
- FC



- Les filtres de conv de 5×5 .
- Pooling 2×2 , *stride* 2.
- Premier "template" d'un CNN avec succession de couches [conv, pool]
- \approx **60000 parametres**

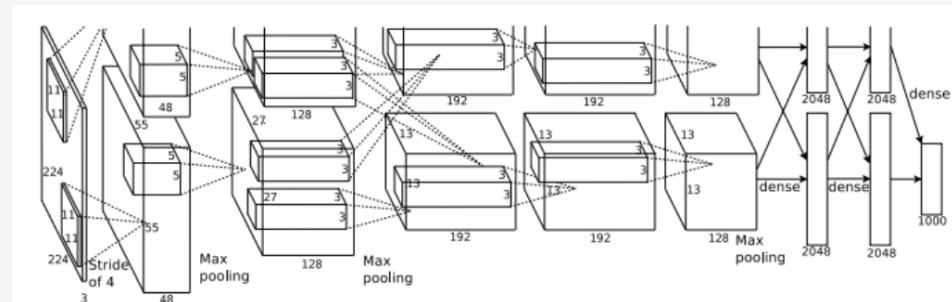
Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, « Gradient-Based Learning Applied to Document Recognition »

5

AlexNet (2012)

- Architecture:

- Conv1
- MaxPool1
- Norm1
- Conv2
- MaxPool2
- Norm2
- Conv3
- Conv4
- Conv5
- MaxPool3
- FC6
- FC7
- FC8



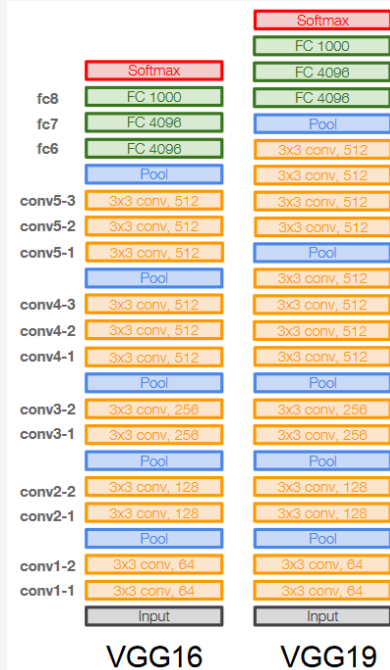
- Premier à implémenter ReLu dans un CNN.
- Plusieurs couches Conv avant un pooling.

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton « ImageNet Classification with Deep Convolutional Neural Networks »

6

VGG (2014)

- Architecture « plus profonde ».
- Plusieurs couches Conv avec filtre de 3×3 .
- **138M paramètres**
- Version plus profonde VGG-19

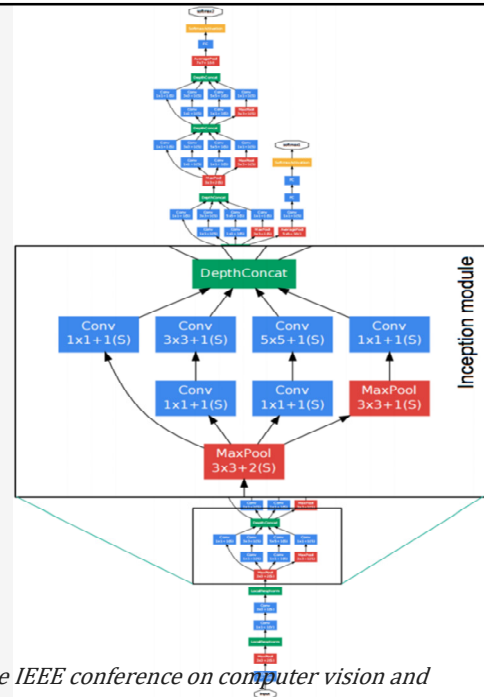


Karen Simonyan, Andrew Zisserman « [Very Deep Convolutional Networks for Large-Scale Image](#) »

7

GoogleNet (2014)

- 22 couches
- Module « inception »
- Seulement **5M paramètres**, 12× moins que AlexNet



Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

8

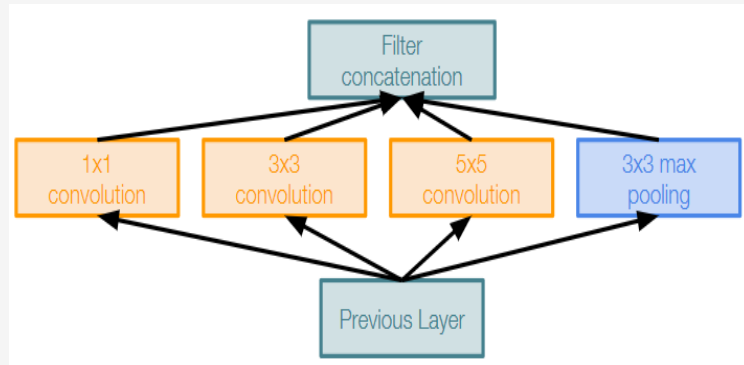
GoogleNet (2014)

- GoogleNet (2014)

➤ Appliquer des filtres parallèles de taille différentes

- Conv (1×1) , (3×3) , (5×5)
- Pool (3×3)

➤ Concaténer les features map ensemble.



Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

9

GoogleNet (2014)

- GoogleNet (2014)

➤ Nombre de paramètres très élevé.

➤ Calcul complexe.

Conv (1×1) $28 \times 28 \times 128 \times 1 \times 1 \times 256$

Conv (3×3) $28 \times 28 \times 192 \times 3 \times 3 \times 256$

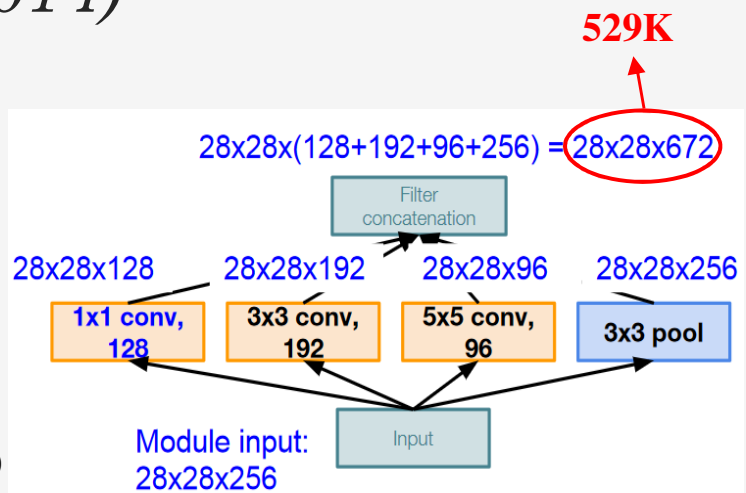
Conv (5×5) $28 \times 28 \times 96 \times 5 \times 5 \times 256$

=> **854M opérations**

➤ **Solution:**

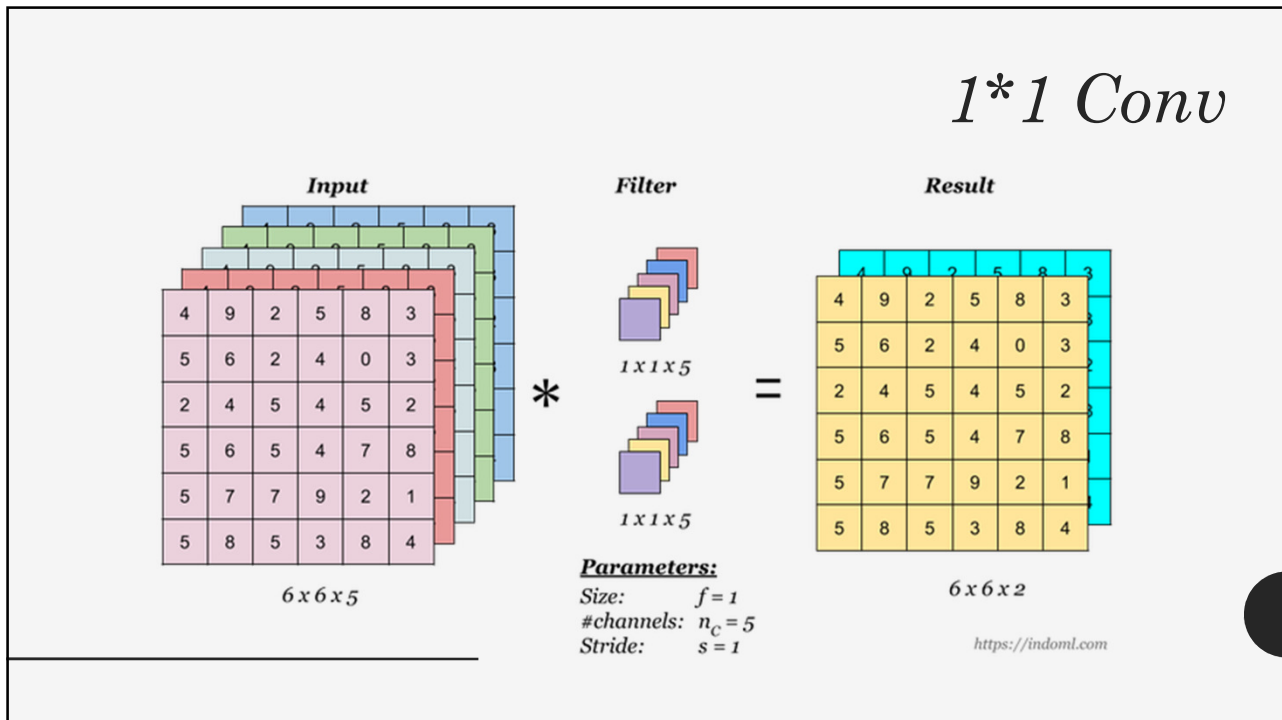
➤ Ajouter une couche Conv (1×1)

Pour réduire la profondeur des feature map



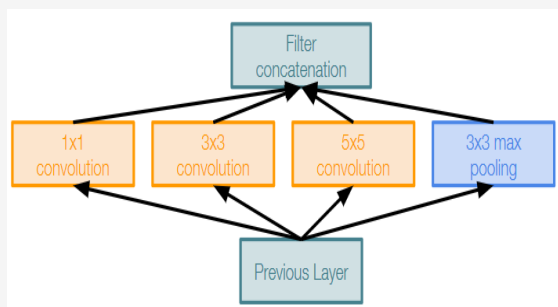
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

10

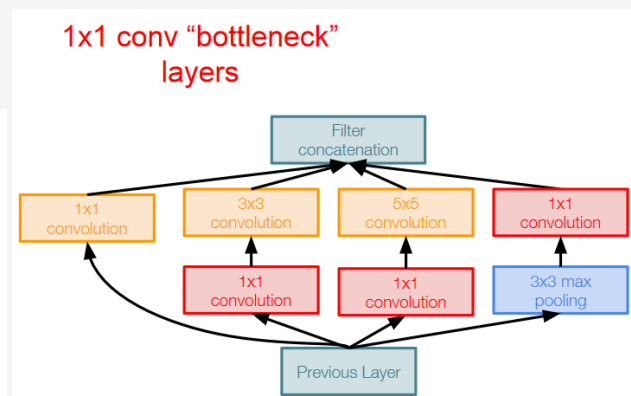


11

GoogleNet (2014)



Module inception



Module inception avec réduction de dimension

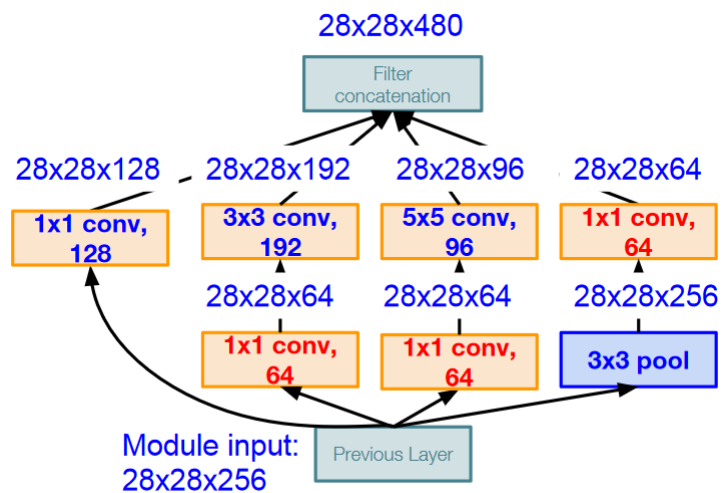
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

12

GoogleNet (2014)

Conv(1×1) 28*28*64*1*1*256
 Conv(1×1) 28*28*64*1*1*256
 Conv(1×1) 28*28*128*1*1*256
 Conv(3×3) 28*28*192*3*3*64
 Conv(5×5) 28*28*96*5*5*64
 Conv(1×1) 28*28*64*1*1*256
 => **358M opérations**

➤ Contre **854M** avec un module Inception basic

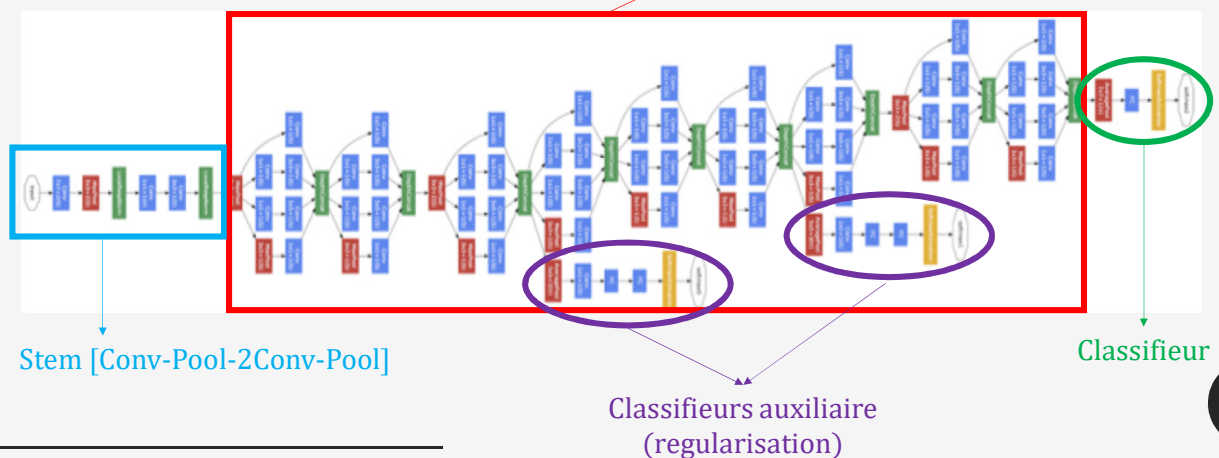


Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

13

GoogleNet (2014)

Empiler des module inception

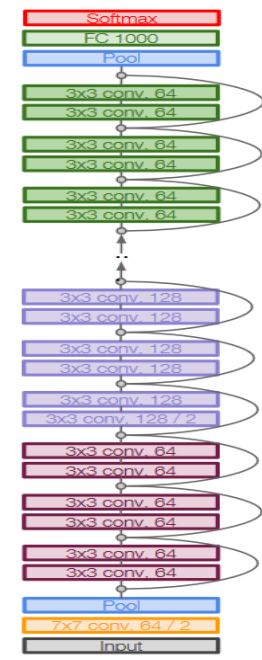


Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

14

ResNet (2015)

- Réseaux « **Très Profond** » (152 couches)
- Introduit le concept de « **skip connection** » dans le block résiduel

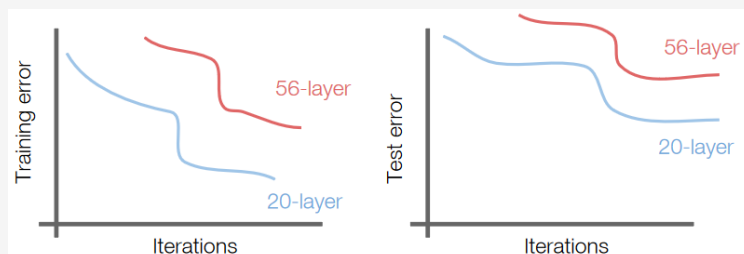


Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun « Deep Residual Learning for Image Recognition »

15

ResNet (2015)

- Problèmes avec les réseaux très profonds:
- Empiler une suite de couche Conv => erreur de training élevée



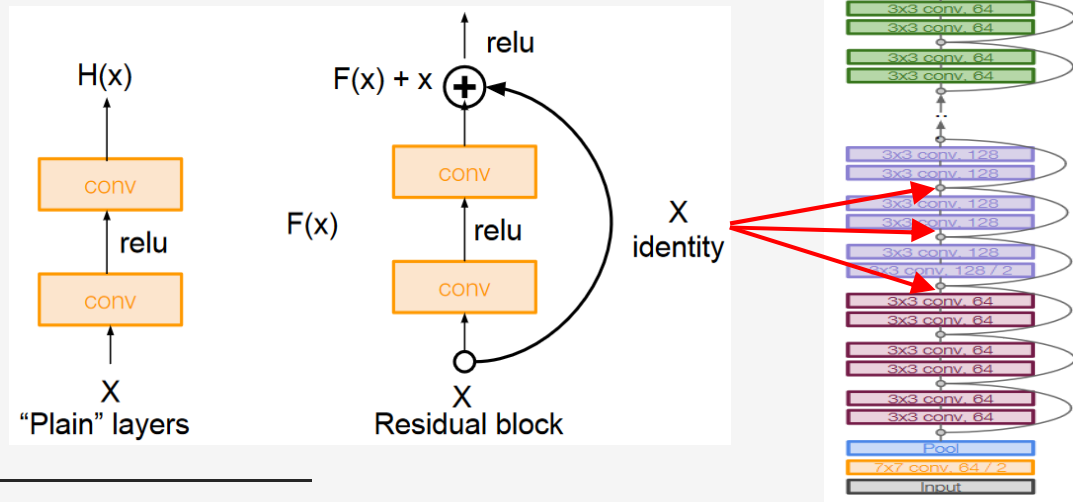
- Problème d'optimisation de réseaux très profonds !
- **Solution:**
- **Ajouter des block résiduels**



Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun « Deep Residual Learning for Image Recognition »

16

ResNet (2015)

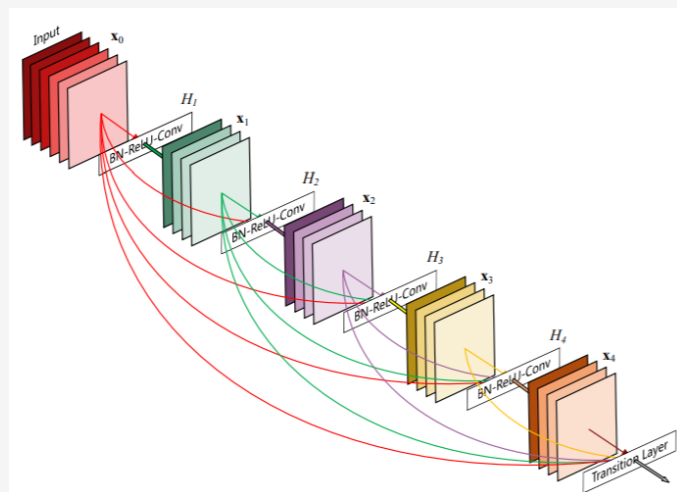


Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun « Deep Residual Learning for Image Recognition »

17

DenseNet (2017)

- Simplifier les connexions entre les couches.
- **Problèmes:**
 - Gradient vanishing pour des réseaux très profonds
 - Nombres très élevé de paramètres à apprendre
 - Pourquoi ne pas réutiliser les paramètres déjà appris sur les couches précédentes?



Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

18

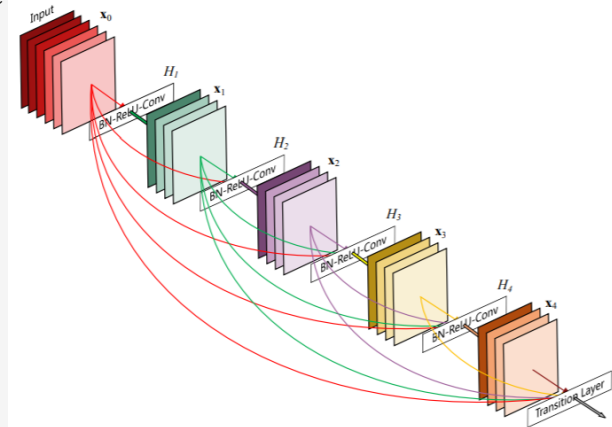
DenseNet (2017)

➤ Différence avec le ResNet:

- Le ResNet “**somme**” la feature map de sortie avec son entrée
- Le DenseNet “**concatène**” à la feature map de sortie son entrée. (et les dimensions ??)
- les DenseBlock assurent que les tailles des feature maps restent compatibles, **mais** le n° filtre change.

➤ Comment?

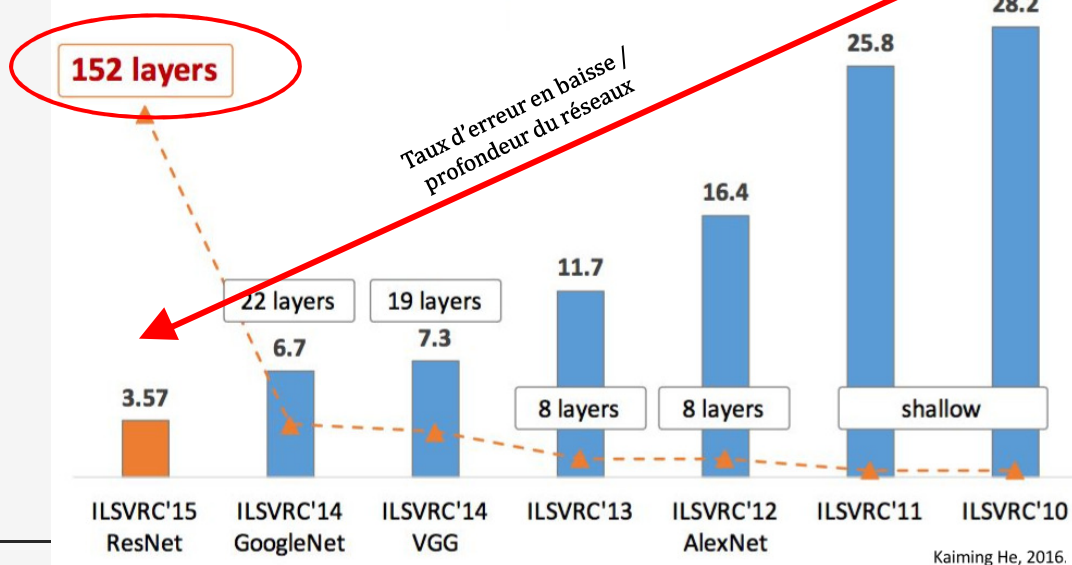
- **Transition layer:** BN + 1×1 Conv + 2×2 pooling
- **Growth rate:** régularise la quantité d'information



Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

19

Plus profond, plus performant



20

Data et CNN

21

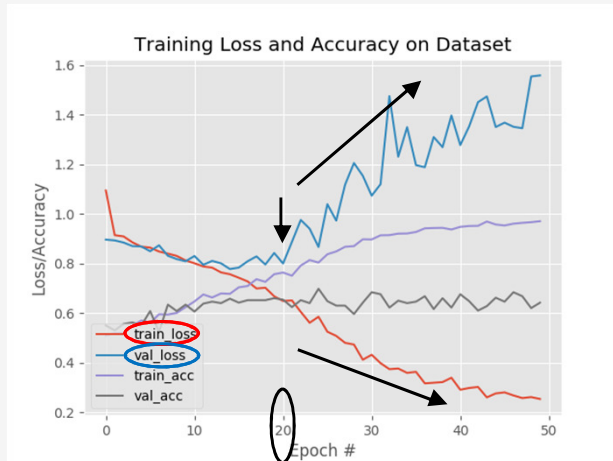
Régularisation: Data augmentation

- **Problème:**
- **généraliser** le modèle CNN sur des données qu'il n'a pas vu lors de l'entraînement.
- **Solution:**
 - Régularisation par Dropout
 - Applicable sur les CNN aussi,
 - Ajouter plus de données lors de la phase d'entraînement,
 - Solution couteuse: collecter et annoter les images est une taches fastidieuse.
 - Ajouter plus d'image ne règle pas toujours le problème de « sur-apprentissage ».
- **Donc:**
 - Ajouter des image **variées** lors de l'entraînement.

22

Data augmentation

- Problème de sur-apprentissage 'over-fitting'

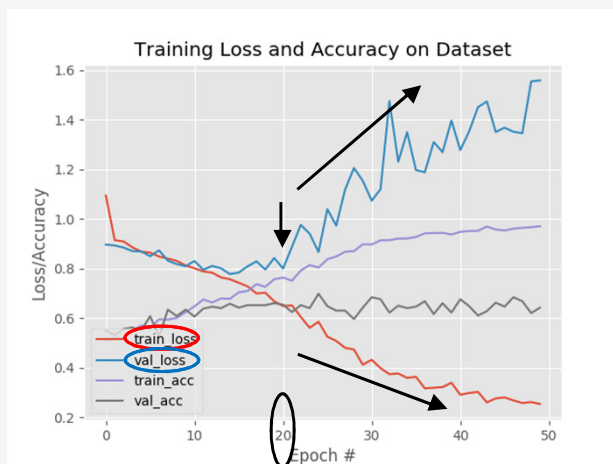


Sans augmentation

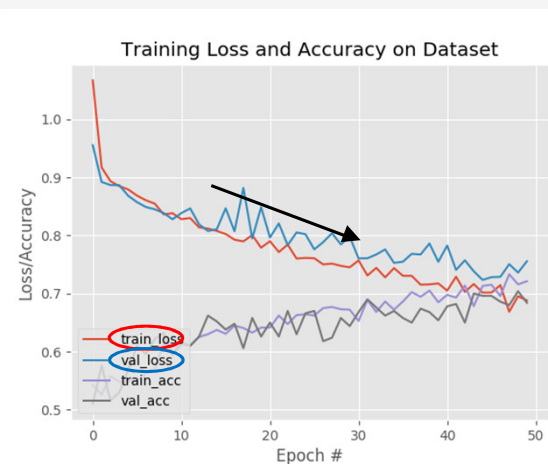
23

Data augmentation

- Problème de sur-apprentissage 'over-fitting'



Sans augmentation

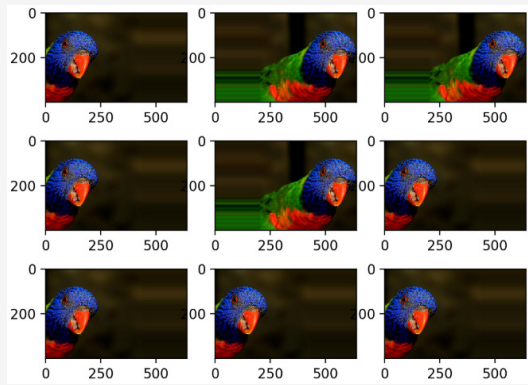


Avec augmentation

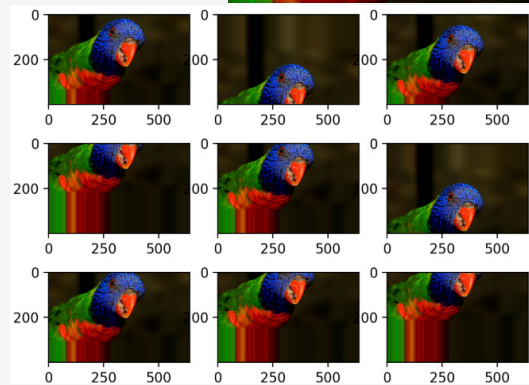
24

- Quelles variations (transformation)

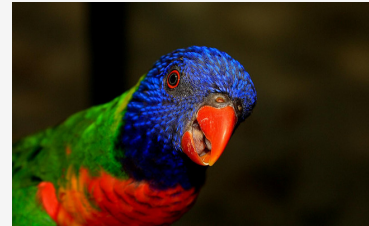
- Translations (*shift*)



horizontal_shift_range



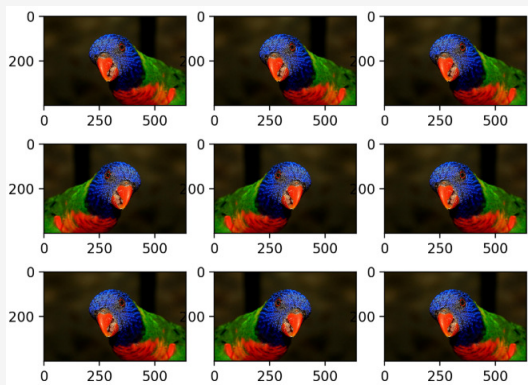
vertical_shift_range



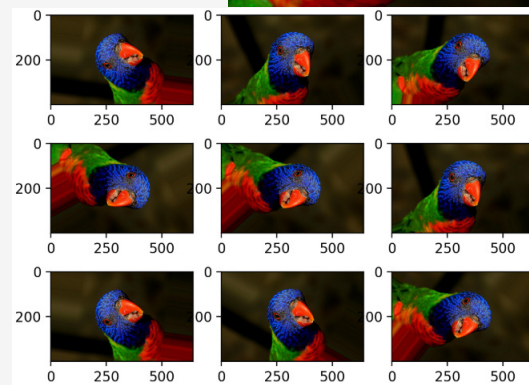
25

- Quelles variations (transformation)

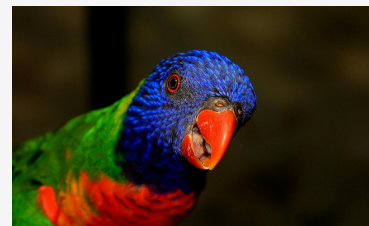
- Renversement horizontal/vertical (*flip*)



horizontal_flip



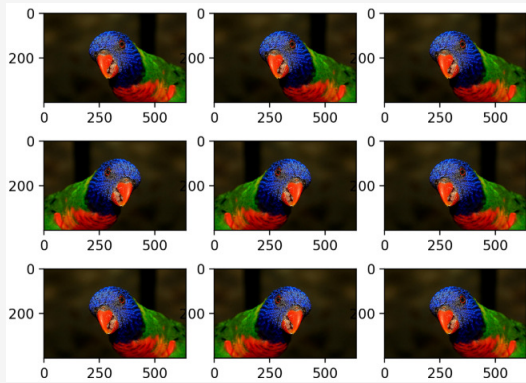
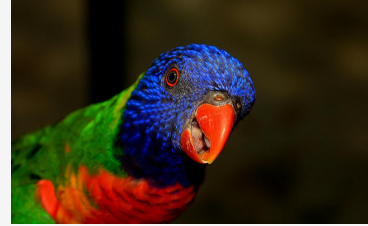
vertical_flip



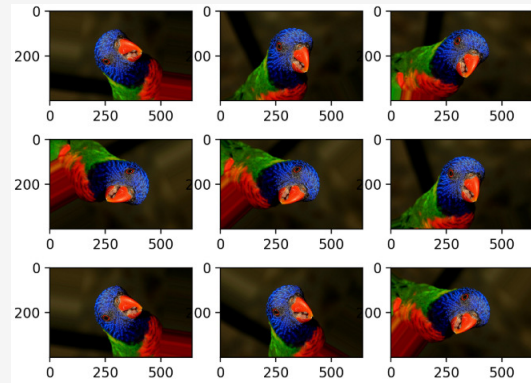
26

- Quelles variations (transformation)

- Renversement horizontal/vertical (*flip*)



horizontal_flip

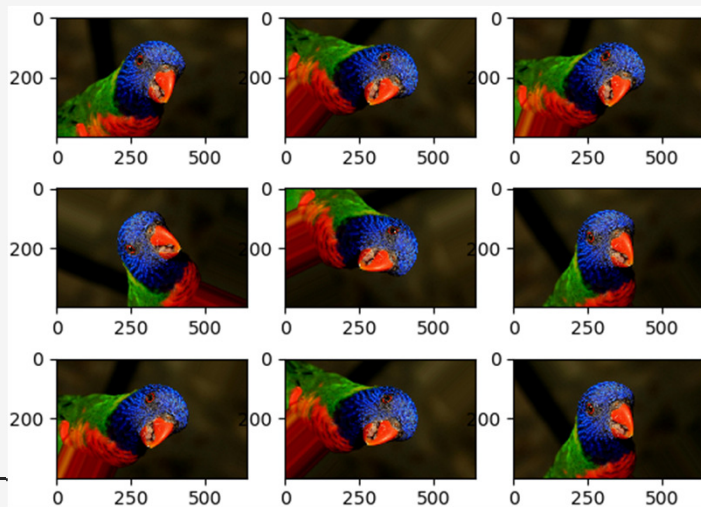
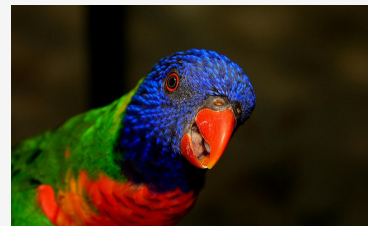


vertical_flip

27

- Quelles variations (transformation)

- Rotations

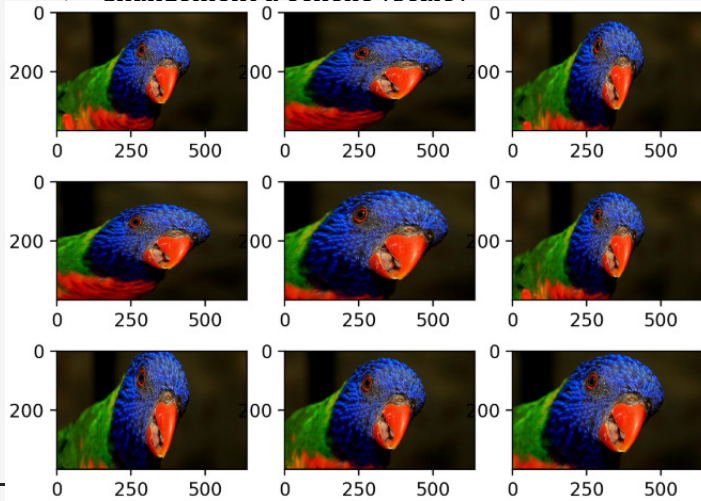


rotation

28

- Quelles variations (transformation)

- Changement d'échelle (*scale*)

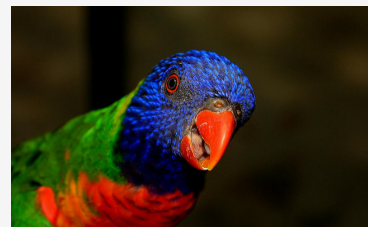
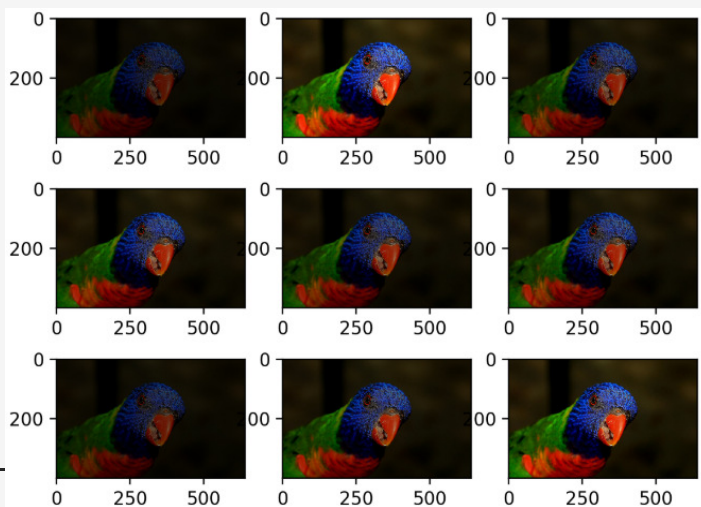


Changement d'échelle

29

- Quelles variations (transformation)

- Changement de contraste

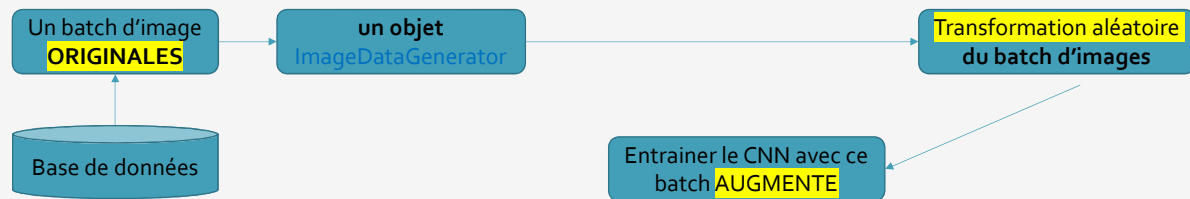


Changement de contraste

30

Data augmentation

- **Comment procéder** (la classe « `ImageDataGenerator` » de Keras)
 - Un batch d'images est présenté en entrée à un objet `ImageDataGenerator`.
 - L'objet `ImageDataGenerator` applique une série de transformation aléatoires sur ce batch
 - Le batch transformé est passé en entrée à un CNN pour l'entraînement.



- ❑ `ImageDataGenerator` ne retourne pas image originales + images transformées, mais **seulement les image transformées.**
- ❑ *Augmentation In-place/on-the-fly* car réalisée **lors** de l'entraînement du CNN et pas à priori.

31

Transfer Learning

32

Transfer Learning: motivation

- Concevoir un nouveau modèle CNN demande beaucoup de données.
- Il est difficile d'avoir des données annotées.
- Entraîner une réseaux « **from scratch** » nécessite beaucoup de calcul et de temps.

33

ImageNet

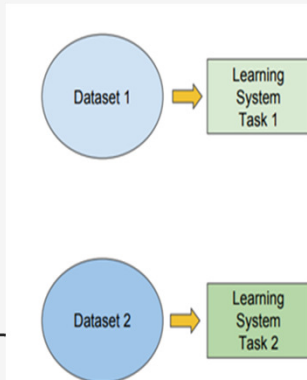


34

Transfer Learning: intuition

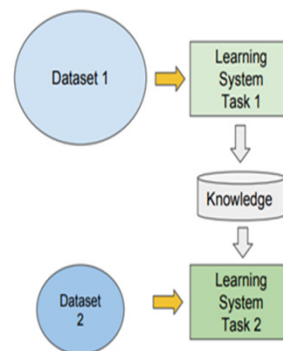
Apprentissage classique

Chaque tâche est entraînée indépendamment d'une autre



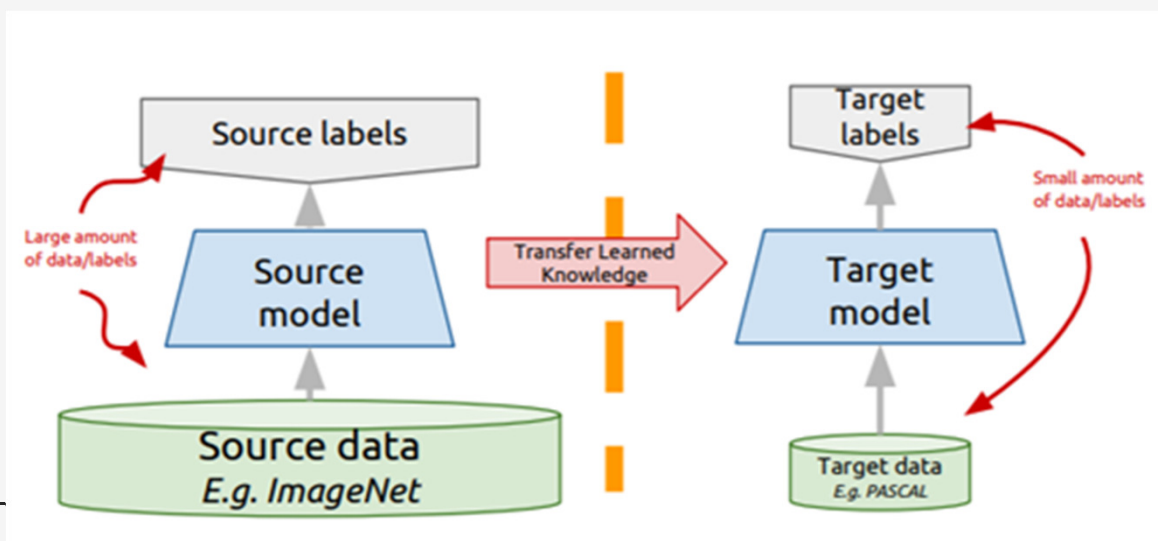
Apprentissage par transfert de connaissance

Entraîner un réseau sur une nouvelle tâche repose sur les connaissances acquises à partir d'une autre tâche.



35

Transfer Learning: idée



36

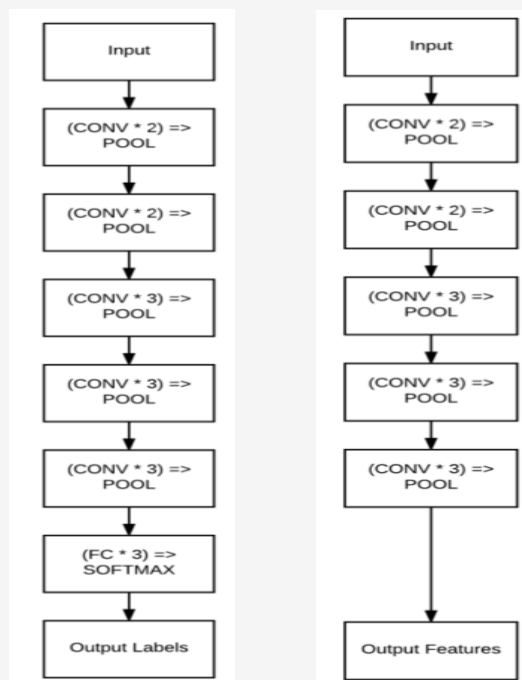
Transfer Learning: en pratique

- Feature Extractor
- Fine-tuning
- Fine-tuning+freeze

37

Feature extractor

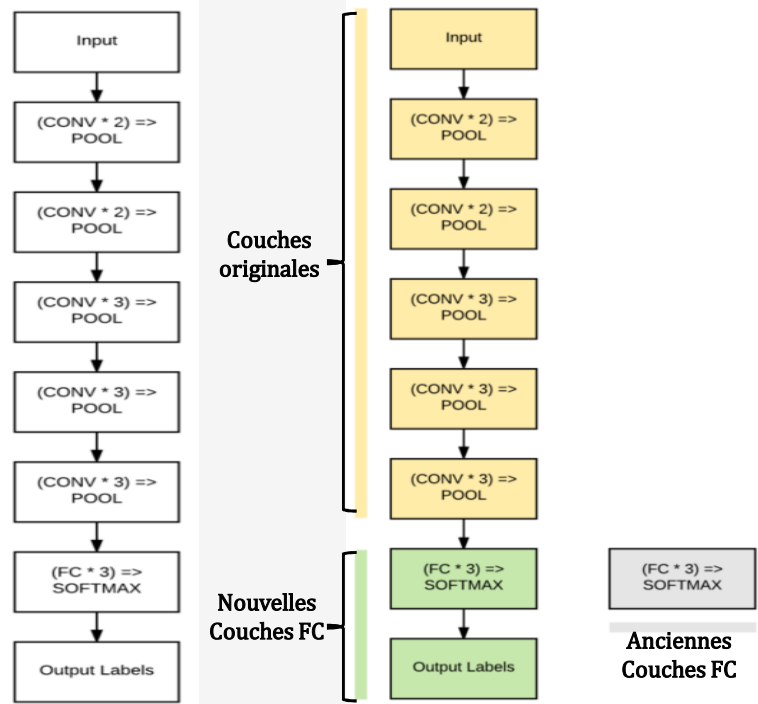
- Réseau **VGG16** « pre-entraîné » (généralement sur Imagenet)
- **Utiliser le CNN comme un extracteur de caractéristiques**
- => enlever les couche FCs,
- => passer une image dans le CNN.
- => récupérer son vecteur descripteur à la dernière couche (pooling ici)



38

Fine-tuning

- Réseau **VGG16** « pre-entraîné » (généralement sur Imagenet)
- Adapter le CNN pour une nouvelle tâche.
- => enlever les couche FCs,
- => Remplacer par de nouvelles couches FC.
- => Réentraîner le réseaux.



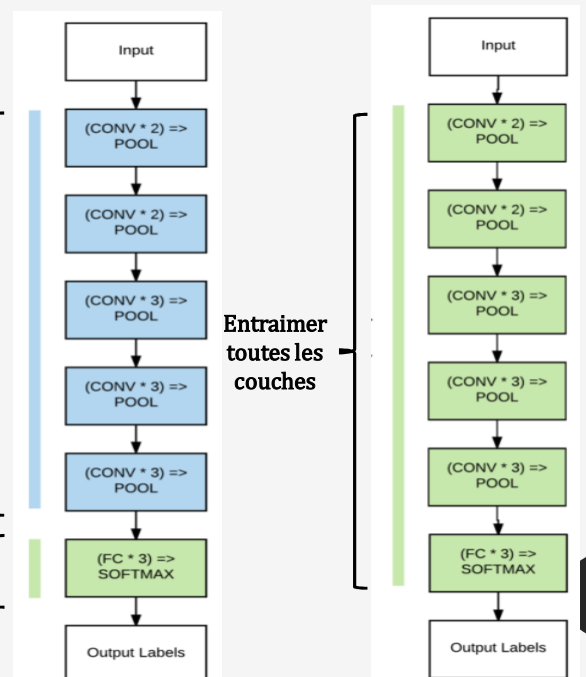
39

Fine-tuning: freeze

- Réseau **VGG16** « pre-entraîné » (généralement sur Imagenet)
- Finetuning que sur les premières couches car contiennent des caractéristiques de bas niveaux communes à toutes les représentations visuelles.
- Finetuning de bout-en-bout: ré-apprendre les poids sur la nouvelle base, pas d'initialisation aléatoire

« geler »
les premières
couches

Entraîner que
les
Couches FC



40

Transfer Learning: Quand faire

41

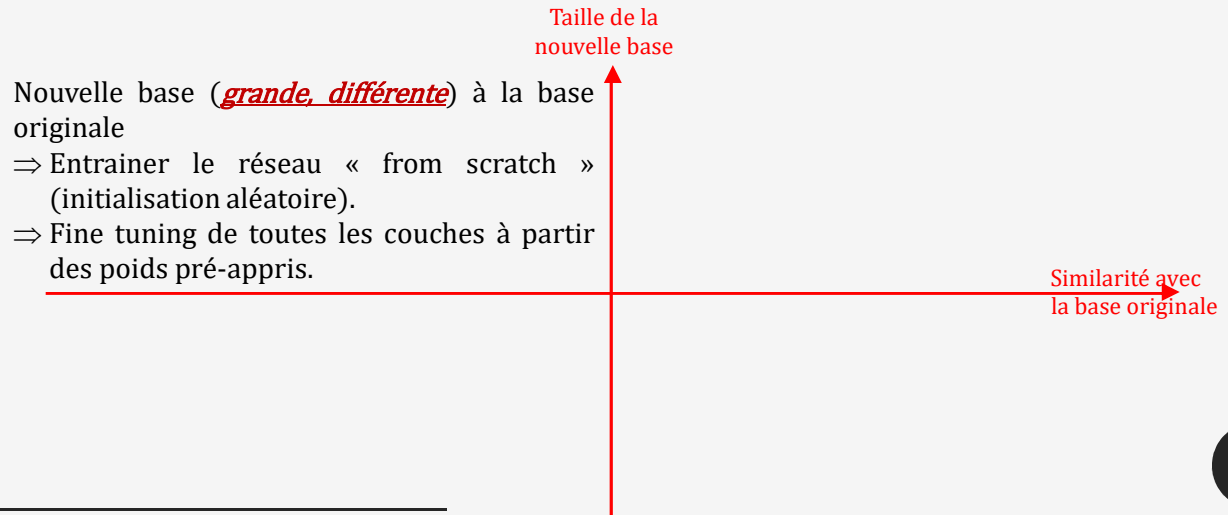
Transfer Learning: quand faire?

Taille de la
nouvelle base

Similarité avec
la base originale

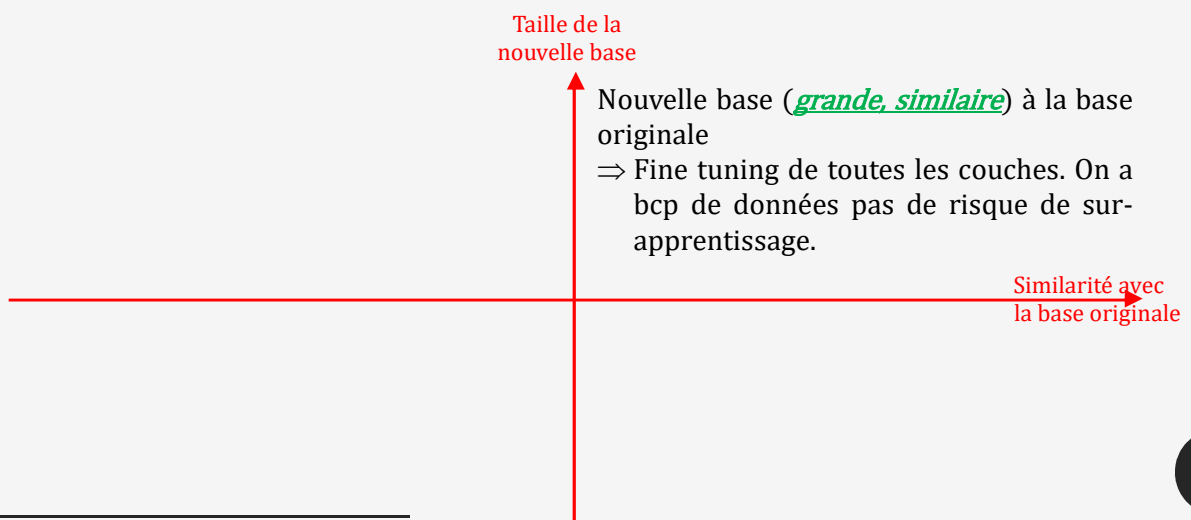
42

Transfer Learning: quand faire?



43

Transfer Learning: quand faire?



44

Transfer Learning: quand faire?

Taille de la
nouvelle base

Similarité avec
la base originale

Nouvelle base (*petite, différente*) de la base originale

⇒ Essayer le data augmentation; avoir plus de données

⇒ Entraîner un classifieur linéaire (SVM) sur des caractéristiques extraite des premières couches du réseaux.

45

Transfer Learning: quand faire?

Taille de la
nouvelle base

Similarité avec
la base originale

Nouvelle base (*petite, similaire*) à la base originale

⇒ Fine tuning cause un sur-apprentissage car pas bcp de données.

⇒ Extracteur de caractéristiques suivi d'un classifieurs linéaire (SVM).

46

Transfer Learning: quand faire?

<p>Nouvelle base (<i>grande, différente</i>) à la base originale</p> <p>⇒ Entraîner le réseau « from scratch » (initialisation aléatoire).</p> <p>⇒ Fine tuning de toutes les couches à partir des poids pré-appris.</p>	<p>Nouvelle base (<i>grande, similaire</i>) à la base originale</p> <p>⇒ Fine tuning de toutes les couches. On a bcp de données pas de risque de sur-apprentissage.</p>
<p>Nouvelle base (<i>petite, différente</i>) de la base originale</p> <p>⇒ Essayer le data augmentation; avoir plus de données</p> <p>⇒ Entraîner un classifieur linéaire (SVM) sur des caractéristiques extraite des premières couches du réseaux.</p>	<p>Nouvelle base (<i>petite, similaire</i>) à la base originale</p> <p>⇒ Fine tuning cause un sur-apprentissage car pas bcp de données.</p> <p>⇒ Extracteur de caractéristiques suivi d'un classifieurs linéaire (SVM).</p>

47

Références

- <https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d>
- <http://scs.ryerson.ca/~aharley/vis/conv/flat.html>
- <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>
- http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf
- <http://cs231n.github.io/assets/conv-demo/index.html>
- <http://cs231n.github.io/convolutional-networks/>
- <https://uijwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
- <http://neuralnetworksanddeeplearning.com/chap1.html>
- <http://cedric.cnam.fr/vertigo/Cours/ml2/preamble.html>
- <http://cs231n.github.io/convolutional-networks/#case>
- <https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/>
- <https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>
- <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253>
- <https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10>

48

Références

- <http://cs231n.github.io/convolutional-networks/#case>
- <https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/>
- <https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>
- <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253>
- <https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10>
- <https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>
- <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>
- <https://keras.io/preprocessing/image/>
- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- <http://cs231n.github.io/transfer-learning/>
- <https://towardsdatascience.com/what-is-deep-transfer-learning-and-why-is-it-becoming-so-popular-91acdcc2717a>
- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- <http://cedric.cnam.fr/vertigo/Cours/ml2/docs/coursDeep5.pdf>