SCALIAN ACADEMY

Cultivons nos talents
Scalian Academy

FORMATION DEEP LEARNING

Fondamentaux des Réseaux de convolution

FONDAMENTAUX DES RÉSEAUX DE CONVOLUTION

MOTIVATIONS

- **Entrées de tailles importantes -> relier tous les neurones d'une couche à tous les neurones de la couche précédente coute cher**

Ex: image 300*300 = 90 000 pixels
1 couche dense de 1028 : 10m+ poids

- **La proximité des pixels a une influence**

Deux pixels éloignés ont peu de chance d'avoir un rapport, alors que deux pixels proches peuvent être corrélés

- **Certains traitement sont invariants par rapport à leur position.**

Un objet peut être détecté à n'importe quelle position d'une image.

LES RÉSEAUX DE NEURONES CONVOLUTIFS

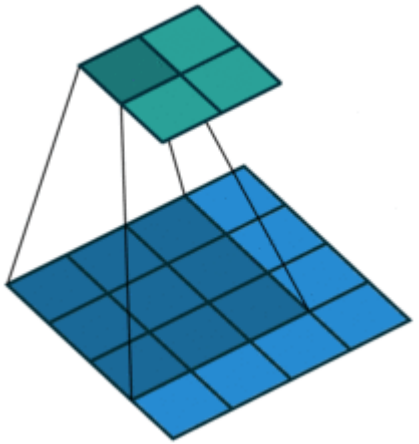
SCALIAN

La solution ? Les CNNs

Les CNNs :

- **Chaque neurone ne voit qu'une partie locale de la couche précédente -> réduction du nombre de poids.**

- **Les poids des neurones de la couches sont partagées :**
  - **calculs indépendants de la position**
  - **économie importante de calcul comme les poids sont partagés**



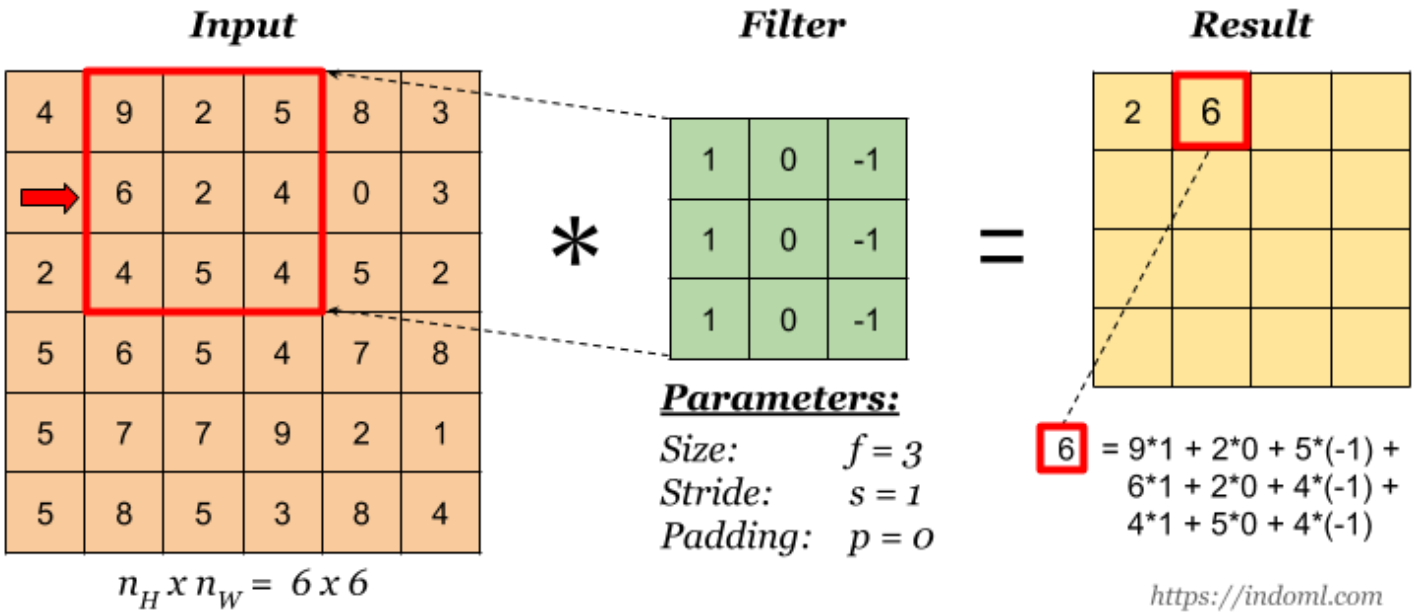
Couche de convolution

Entrée

```
kernel_size=(3, 3),
strides=(1, 1),
```

Source :
https://github.com/vdumoulin/conv_arithmetic
Démo:
TensorSpace

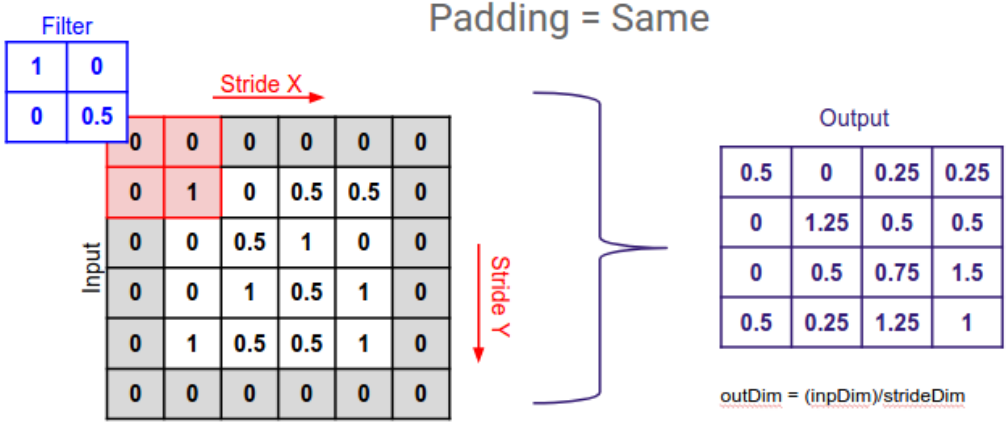
VOCABULAIRE - CONVOLUTION



VOCABULAIRE

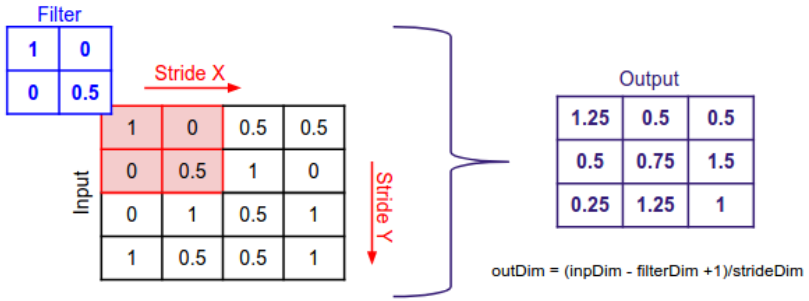
PADDING

Padding = Same

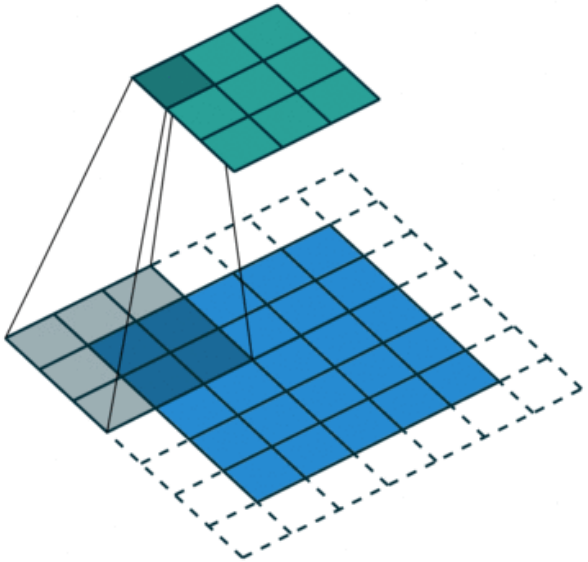


outDim = (inpDim)/strideDim

Padding = Valid

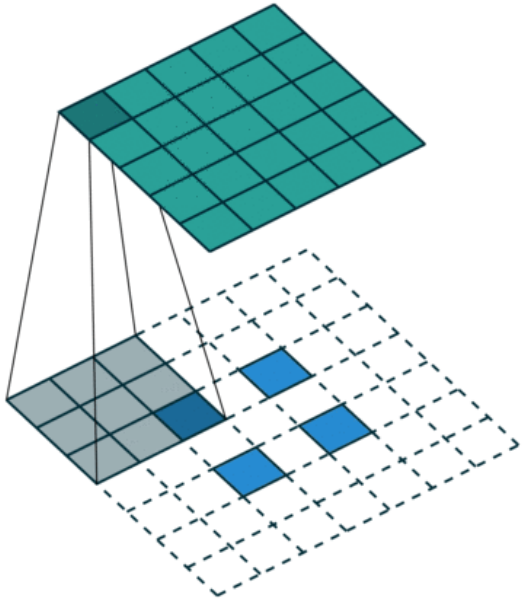


outDim = (inpDim - filterDim +1)/strideDim



Padding

Comment avoir la taille de kernel et le recouvrement que l'on souhaite



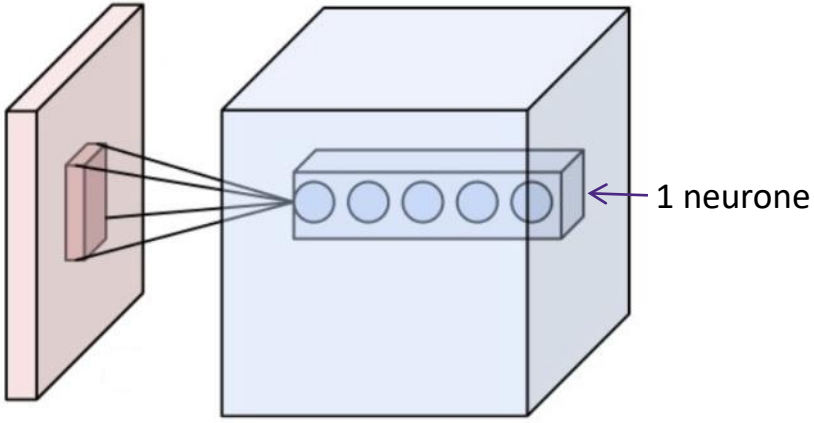
Stride

Comment avoir une taille de sortie plus grande que celle de l'entrée

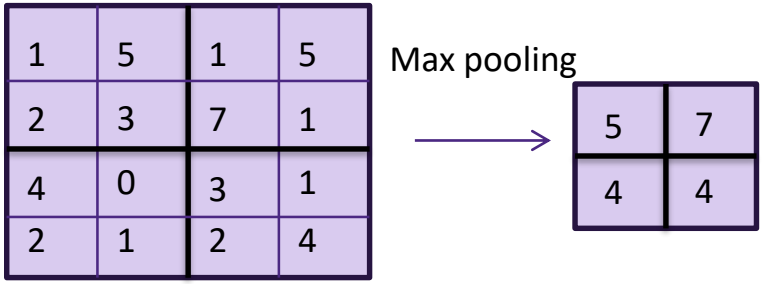
LES RÉSEAUX DE NEURONES CONVOLUTIFS

SCALIAN

- **Calcul de plusieurs informations (« channels ») par région**



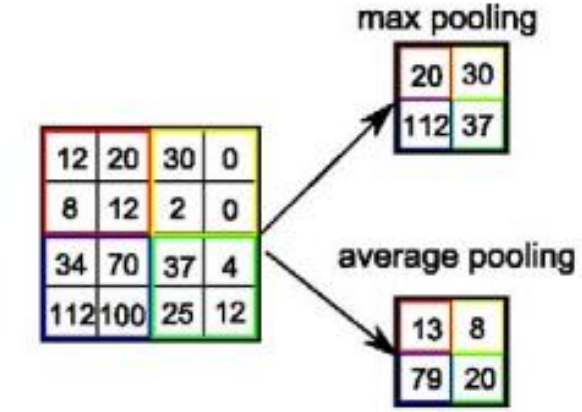
- **Couche de mise en commun (« pooling ») des régions proches**



- **Max pooling, Average pooling, …**

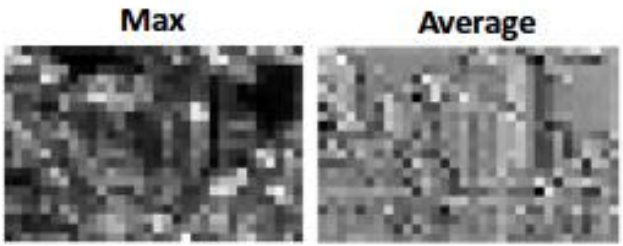
VOCABULAIRE

POOLING



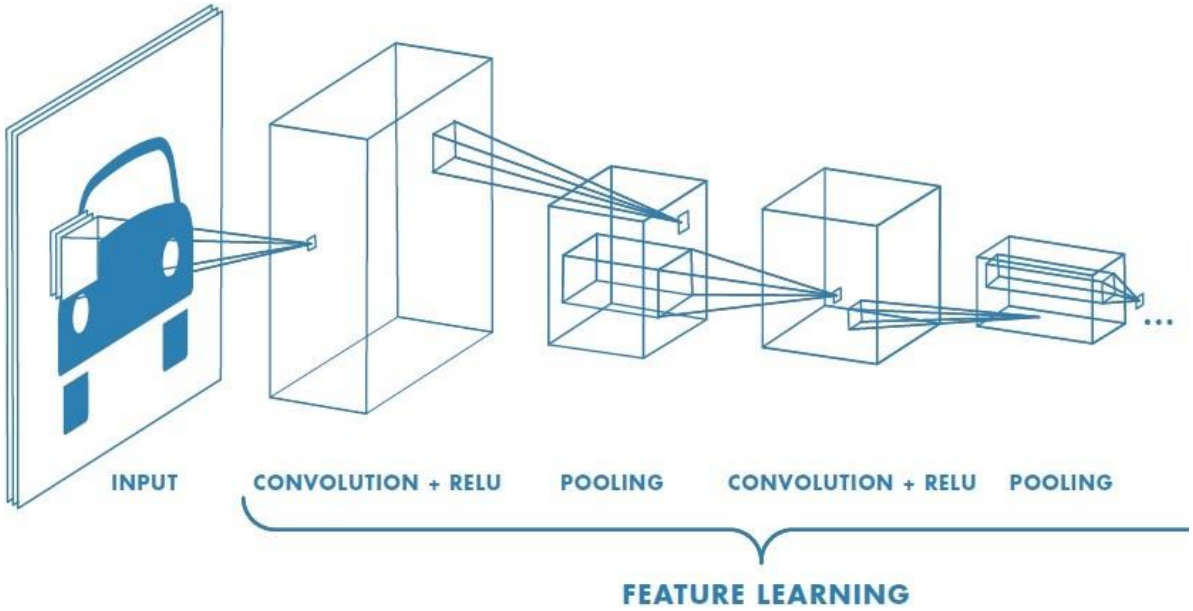
Retient la valeur de la feature la plus importante de la région

Retient la moyenne de la feature sur la région



PRINCIPE DE CONVOLUTION

ON « EMPILE » CES COUCHES POUR EXTRAIRE DES FEATURES COMPLEXES



PRINCIPE DE CONVOLUTION

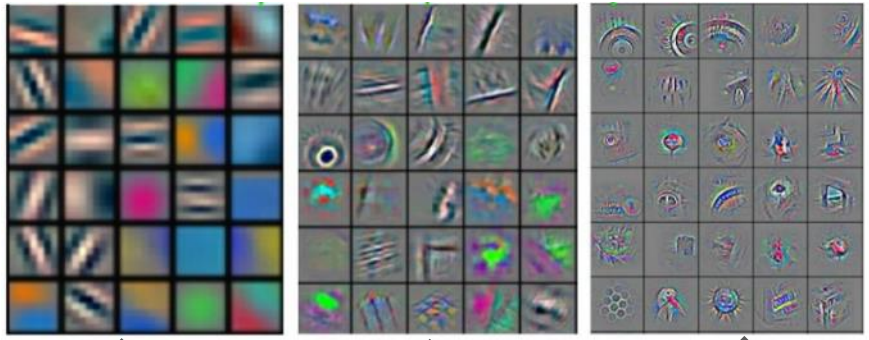
SCALIAN

Extraction d'information

Hiérarchique

Indépendante de la position

Features simples



Features de feature de feature = features complexes

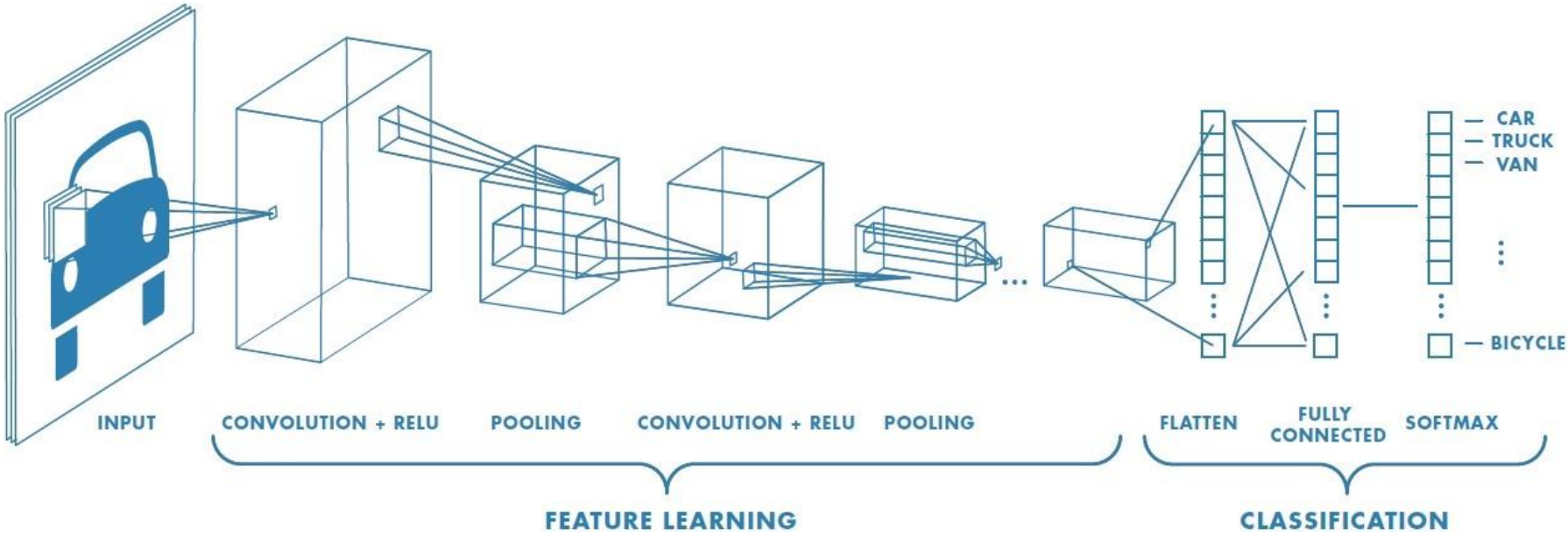
Couche 1

Couche 2

Couche 3

PRINCIPE DE CONVOLUTION

PUIS ON APPREND LA TACHE VIA UN MLP



PRINCIPE DE CONVOLUTION

RÉSUMÉ

Répétés plusieurs fois

Step 1: Convolution:
Application de filtres “Feature Detector” qui génèrent des “Features MAP”. Cette couche permet d’extraire l’information d’une image. La valeur des filtres est obtenu par apprentissage. Les poids sont les valeurs des filtres.

Step 1b: ReLU Layer:
Utilisé pour ajouter de la non-linéarité entre les couches

Step 2: Pooling:
Utilisé pour réduire la dimension des tensors et par consequent réduire l overfitting. On pourra utiliser le maxpooling, averagepooling, etc.

Step 3: Flattening:
Applati la matrice d’entrée en vecteur, facilement interpretable par un réseau de neurons Feed Forward

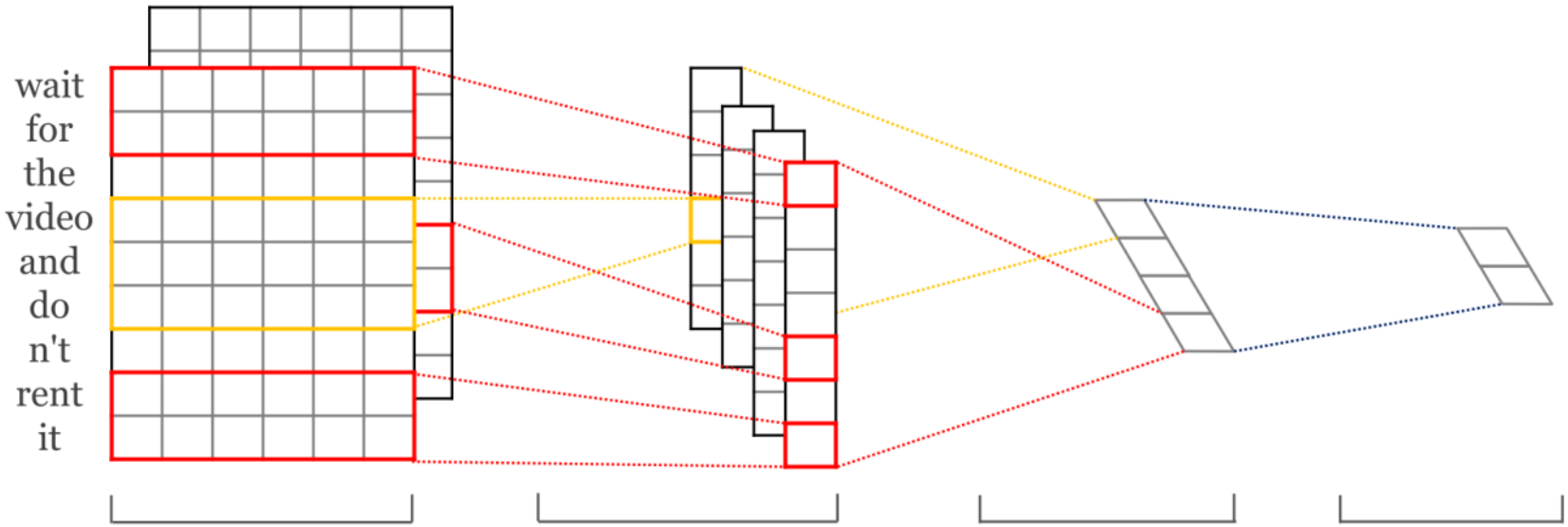
Au moins 2

Step 4: Full Connection
Application d’un reseal Feed Forward pour effectuer la classification

LES RÉSEAUX DE NEURONES CONVOLUTIFS

SCALIAN

Appliqué au texte :



Répresentation de chaque mot sous forme de vecteurs d'entrée

Couche de convolution
1 colonne = 1 channel pour l'ensemble de la phrase

Pooling : max de chaque channel sur l'ensemble de la phrase

Perceptron « fully connected »

Adapté de Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification

L’ÉVOLUTION DES ARCHITECTURES

IMAGENET

Concours annuel

Lancé en 2010

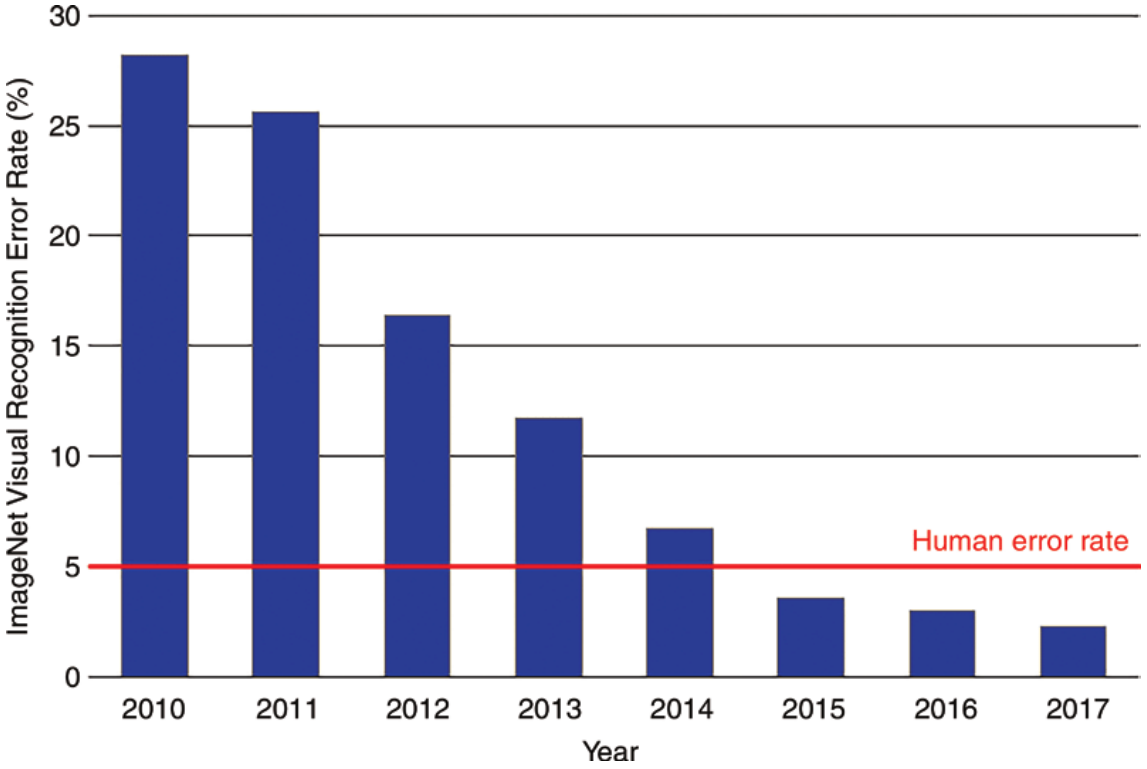
+ 14 millions d'images annotées

+ 24 000 classes

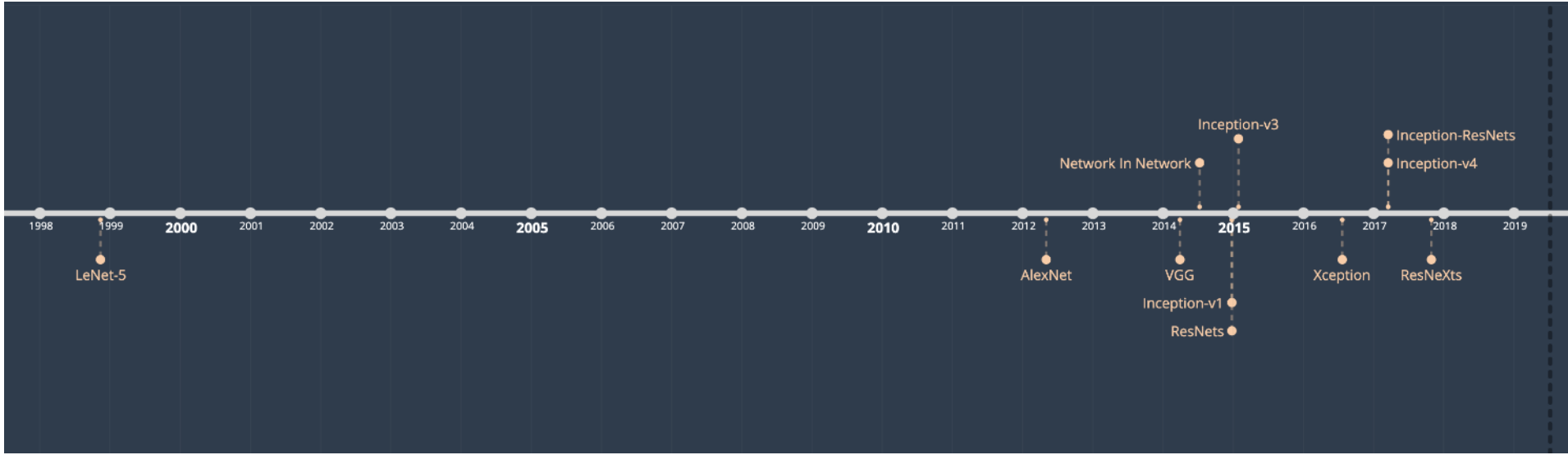
+ 1 millions d'images avec des « bounding boxes »

L'ÉVOLUTION DE LA PERFORMANCE

SCALIAN



L'ÉVOLUTION DE LA PERFORMANCE



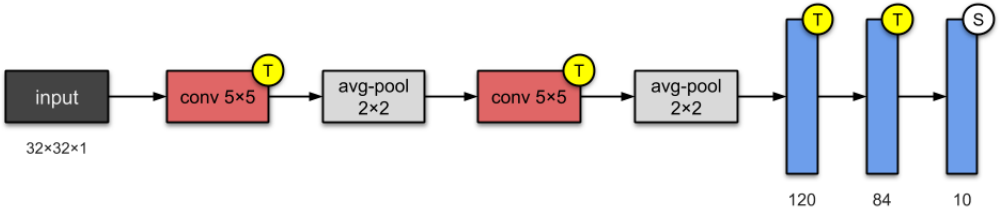
source

L'ÉVOLUTION DE LA PERFORMANCE

SCALIAN

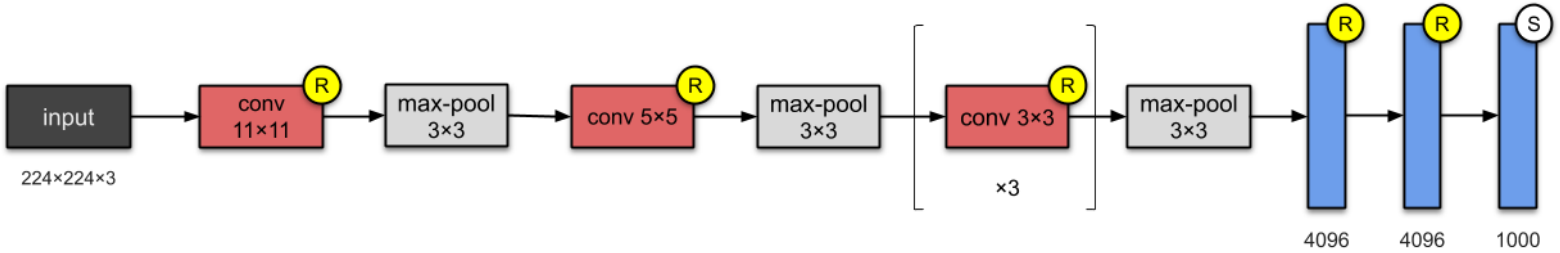
"[m]ost of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures." (Szegedy et al, 2014)

LENET-5 (1998)



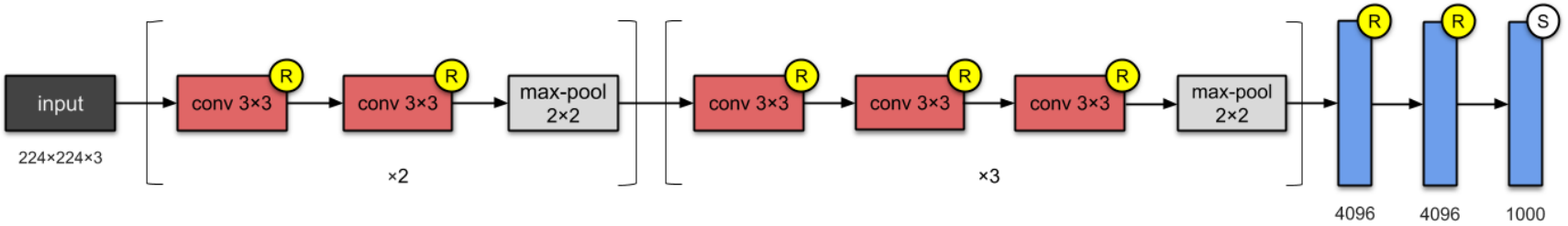
- Une des premières architectures devenus un standard
- 60 000 paramètres

ALEXNET (2012)



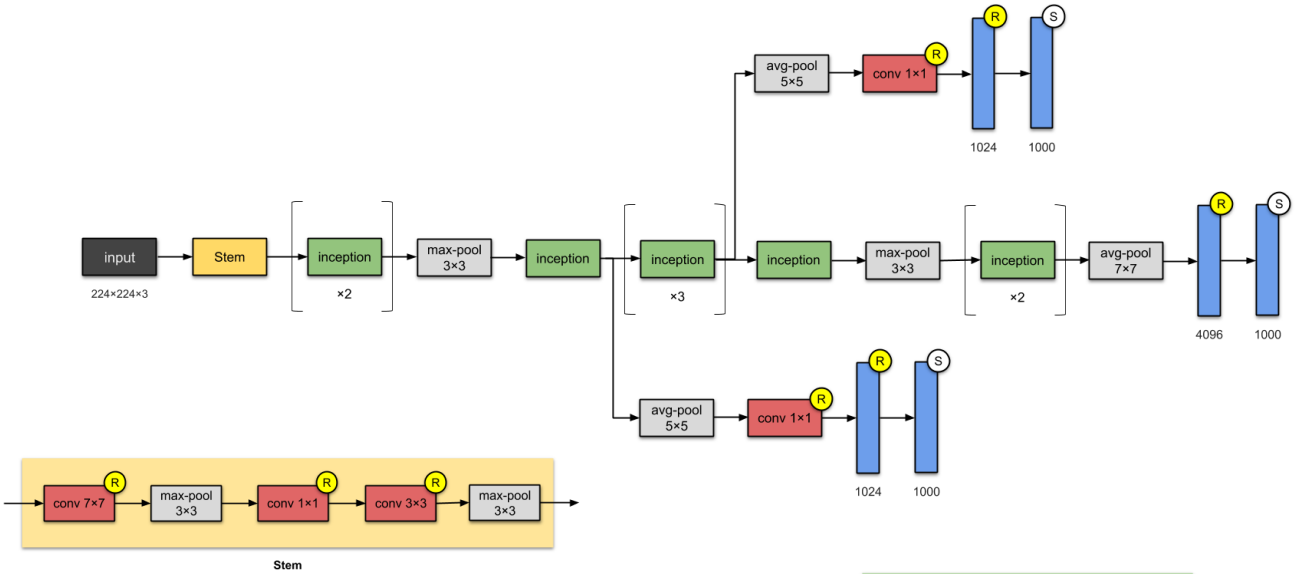
-Une architecture qui surcharge LeNet-5
-60 millions de paramètres
-Intégre des ReLus comme fonctions d'activations

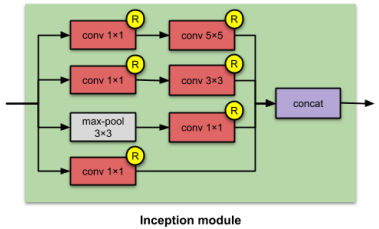
VGG-16 (2014)



-Une architecture qui surcharge AlexNet (deeper)
-138 millions de paramètres
- Stratégie « bruteforce » : plus on a de capacité, mieux on apprendra !

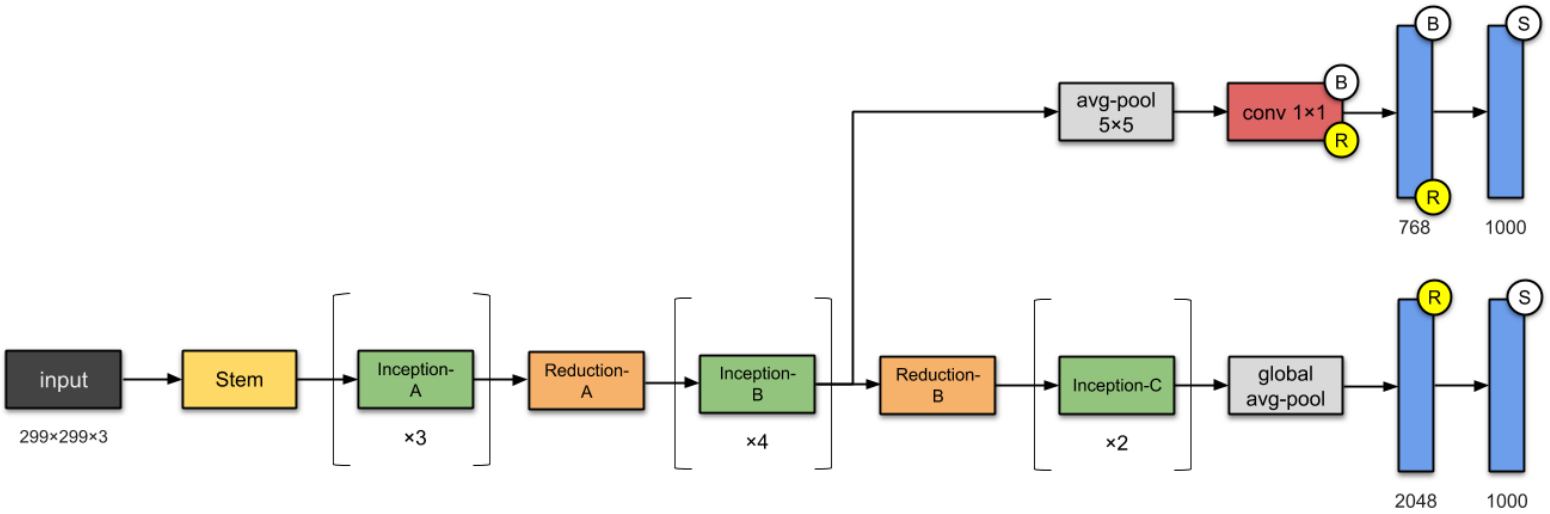
INCEPTION-V1 (2014) - GOOGLENET





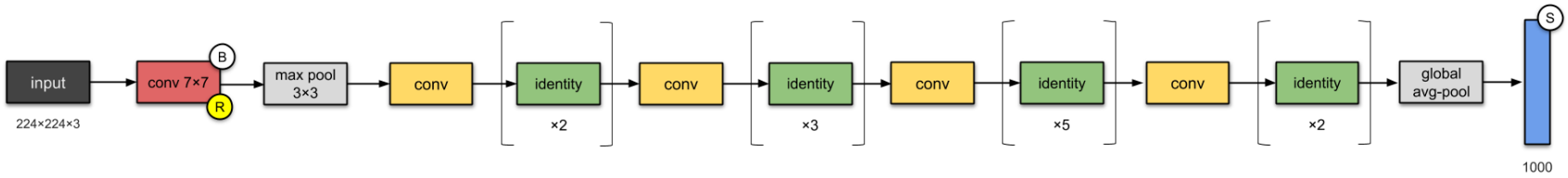
- Stratégie basée sur une architecture plus efficace
- 1×1 convolution : réduction de paramètres + réduit overfitting
- Inception module (différentes tailles de filtres + concaténées)
- Réseaux auxiliaires los de l'entraînement
- 5 millions de paramètres

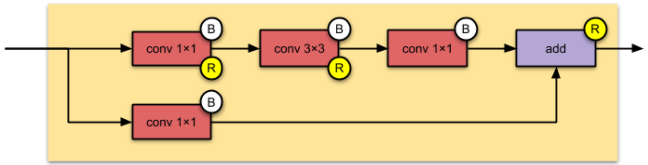
INCEPTION-V3 (2015)



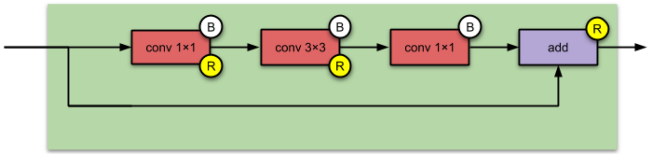
- Factorisation dans l'inception module plus optimisée (par ex : 5*5 => 2 de 3*3)
- 24 millions de paramètres

RESNET-50 (2015)



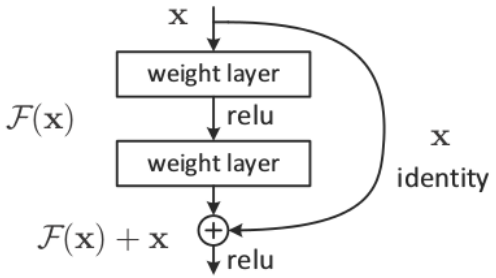


Conv block



Identity block

- utilise la batch normalisation
-intégre les « skip connections » : facilite l'entraînement du réseau
  - lutte contre la disparition des gradients de l'erreur
- Architecture plus profonde (152 couches)
-26millions de paramètres

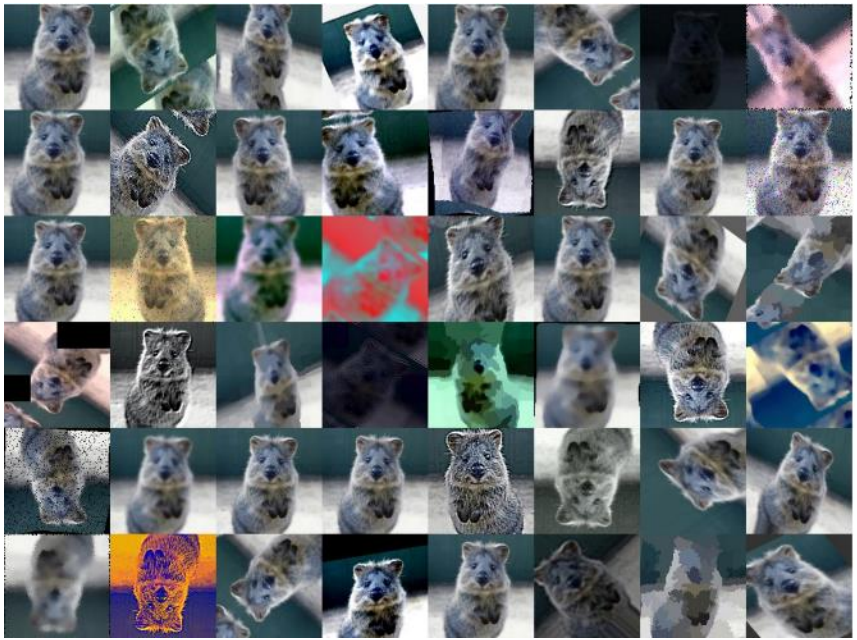


DATA AUGMENTATION

SCALIAN

Problématique : Pas assez de données pour entraîner une réseau de neurones profond

Principe : Augmenter le nombre d'exemples de données en générant de nouvelles données à partir des existantes modifiées



TRANSFERT LEARNING

Les réseaux de convolutions demandent beaucoup de données,

- **souvent trop pour une application réelle**

Image Net

- + 10 millions d'images labelisées
- Ordre de grandeur année
- Pas de limite materielle

Cas d'usage

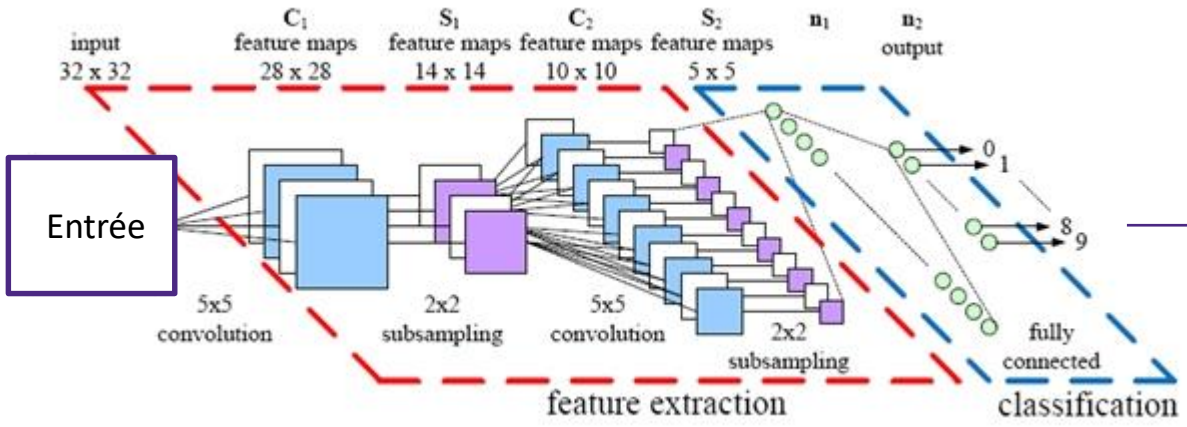
- Millliers d'images
- Echéance courte (<3 mois)
- Pc local

LE TRANSFERT LEARNING

Applicable en théorie à tout type de réseau de neurones.

Idée : bénéficier d'un réseau entraîné sur une tâche pour en effectuer une autre

En pratique particulièrement adaptés aux réseaux à base de CNN, mais pas que !



Réponse du réseau

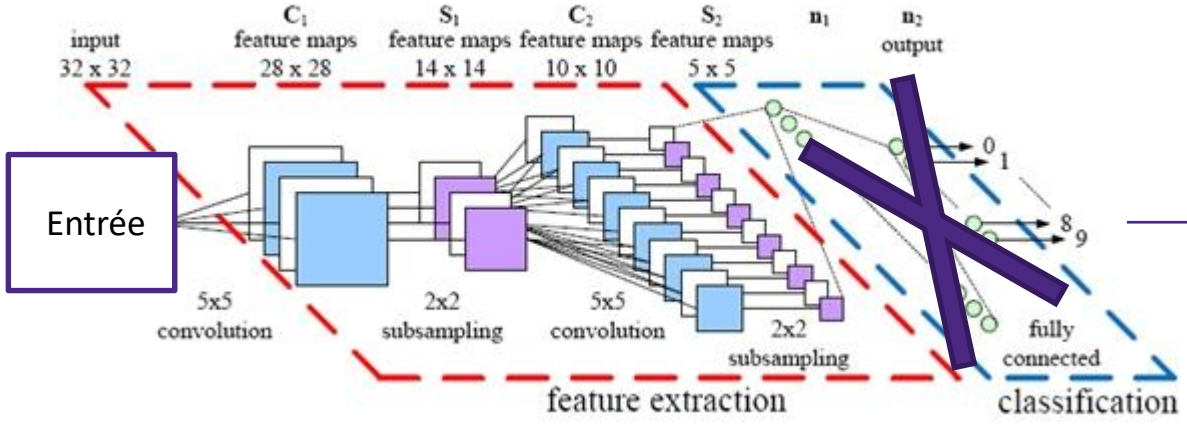
LE TRANSFERT LEARNING

Applicable en théorie à tout type de réseau de neurones.

Idée : bénéficier d'un réseau entraîné sur une tâche pour en effectuer une autre

En pratique particulièrement adaptés aux réseaux à base de CNN, mais pas que !

Oubli et apprentissage d'une nouvelle classification à la place

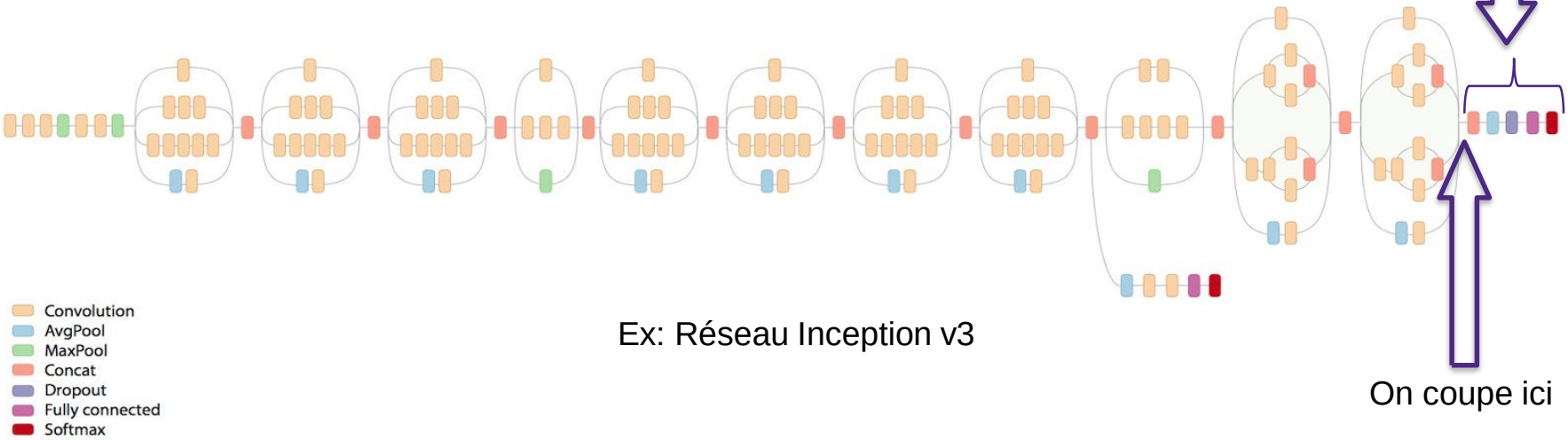


Réponse du réseau à la nouvelle tâche

EXEMPLE SUR LE TRAITEMENT D'IMAGE

SCALIAN

- **Pour des tâches de traitement d'images, les modèles pré-entraînés facilement disponibles online.**
- **Permet de bénéficier d'encoding d'images de qualité.**



Ex: Réseau Inception v3

LE TRANSFERT LEARNING

SCALIAN

Pour le traitement d'image :

- **Gain de temps d'entraînement et d'exemples nécessaire (millions -> centaines ou milliers)**
- **Optimisation par pré-calcul des bottlenecks pour chaque image**
  - **Gain de temps supplémentaire**
  - **Nécessite beaucoup d'espace de stockage (>2Mo/image pour certains modèles)**
- **Utilisé par de nombreux vainqueurs de compétitions kaggle**
  - **Souvent cumulé avec des méthodes ensemblistes sur les features de différents réseaux tirés d'Imagenet.**
- **Les features peuvent être utilisées par des algorithmes non-réseaux de neurones (lstm, etc).**

LE TRANSFERT LEARNING

SCALIAN

Bien choisir ses features:

Plus les couches de convolutions sont profondes plus elles sont spécialisées => on peut donc utiliser les couches plus générales pour faire du transfert learning (TL)

Les modèles les plus performants pour ImageNet ne sont pas forcements les meilleurs: spécificté vs généralisation

Finetuning: Après TL réentrainer tout le réseau pour préciser les features pour notre cas d'usage

Pruning: Elagage ou supression des features/channels non utilisés par le TL

CONCLUSION

SCALIAN

Le deep learning offre de nombreuses applications liées aux données de types images.

Les réseaux de neurones forment l'état de l'art sur beaucoup d'applications liées à la classification d'images, la détection et la génération d'image.

Les contraintes inhérentes aux réseaux de neurones (taille du jeu de données, temps d'entraînement,…) se retrouvent dans leurs applications décrites dans cette présentation.

<u>Pensez au transert learning !!</u>