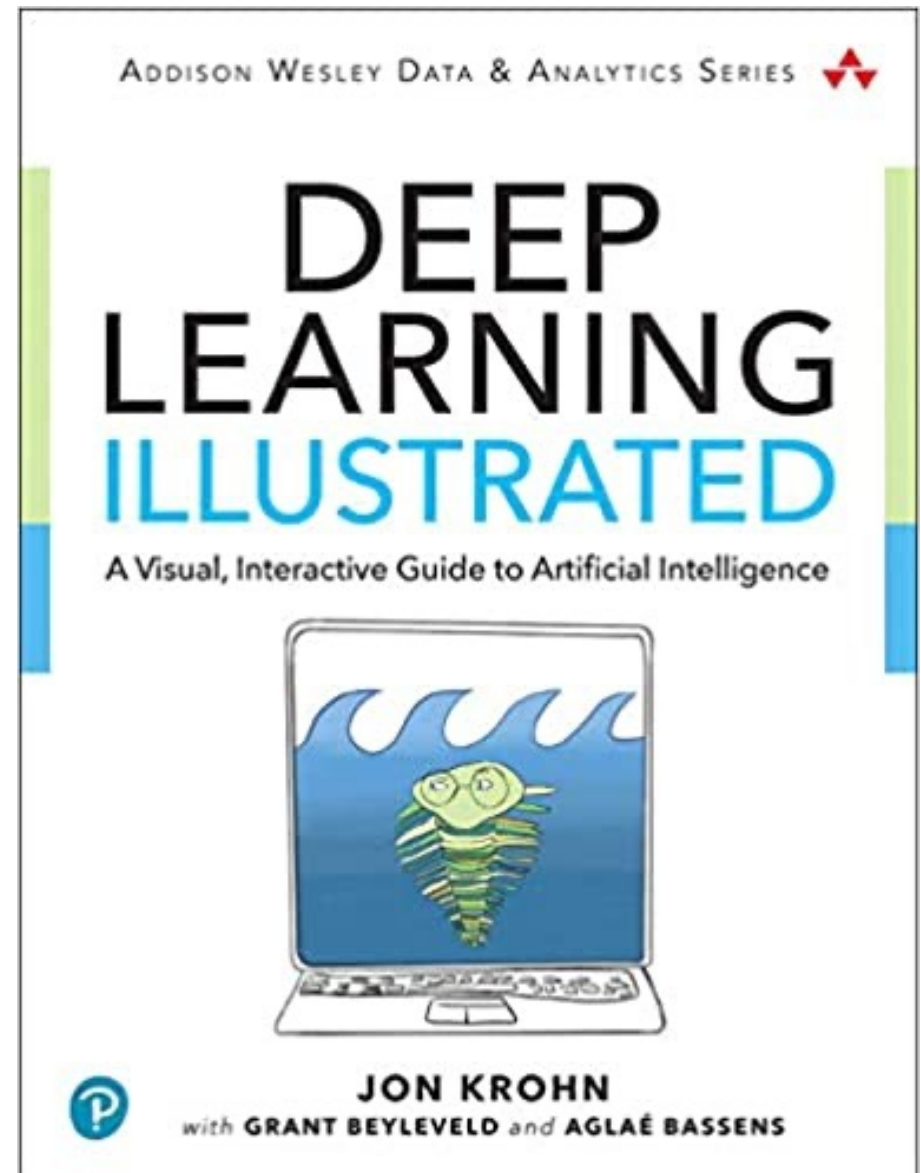
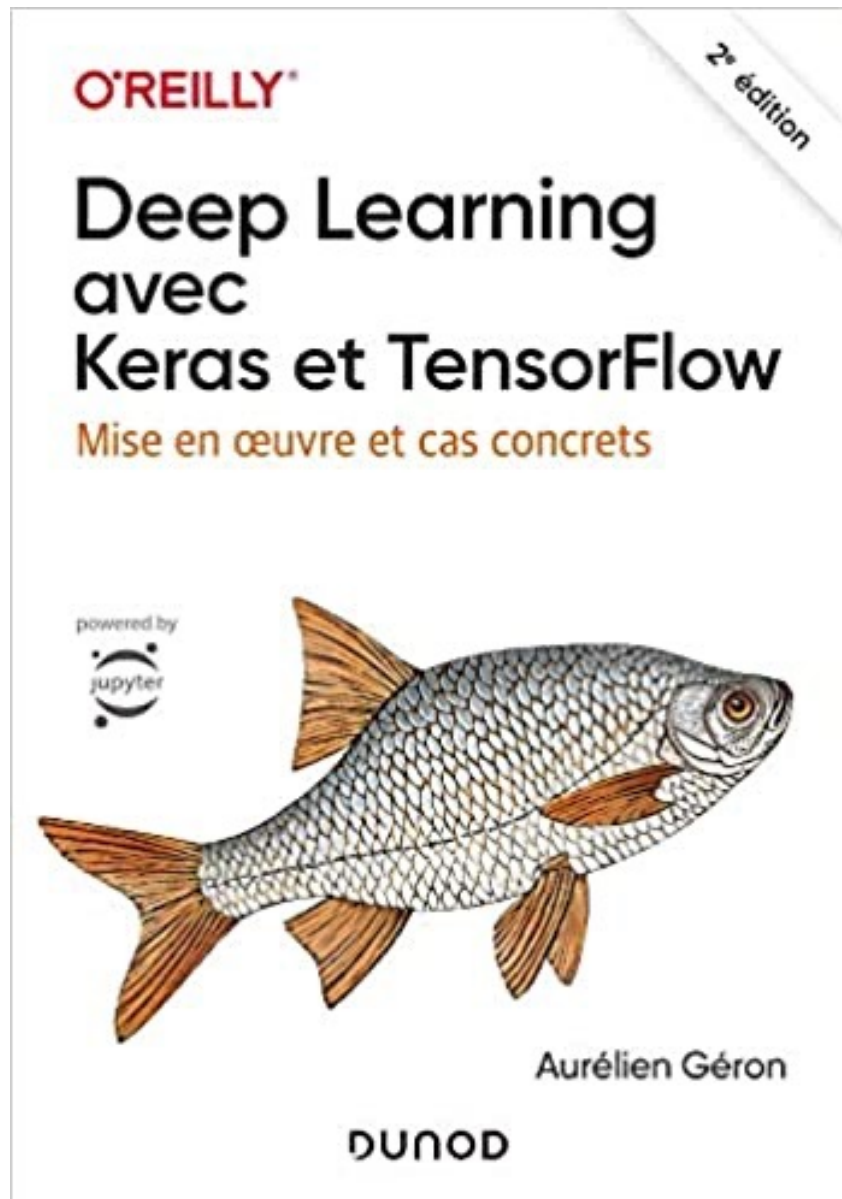


INF7370

Réseaux de neurones convolutifs

Mohamed Bouguessa

Sources





Réseaux de neurones profonds

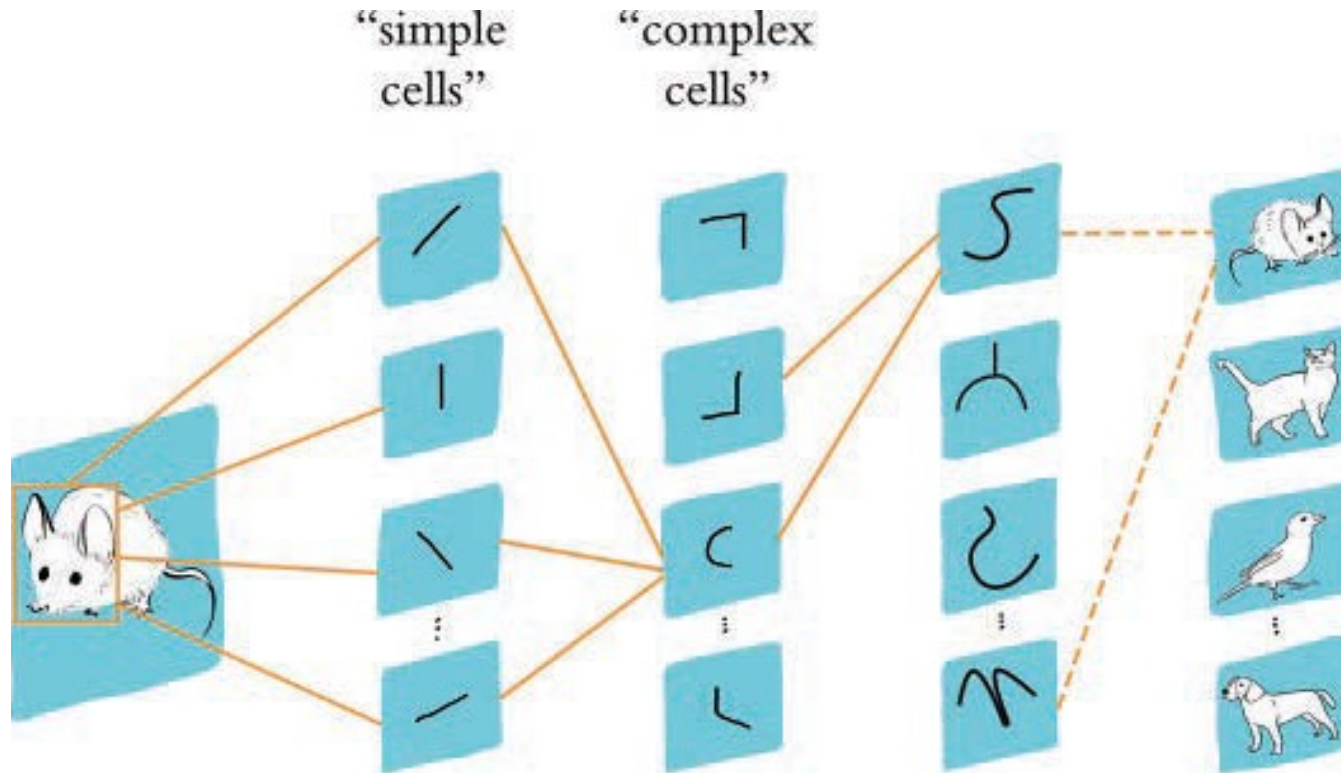
- En termes simples : les modèles de l'apprentissage profond sont bâtis sur le même modèle que les réseaux de neurones multicouches avec de nombreuses couches intermédiaires.
- En empilant plusieurs couches cachées successives, les réseaux profonds permettent d'apprendre des caractéristiques au niveau des couches supérieures qui dépendent elles-mêmes d'autres caractéristiques au niveau des couches inférieures. Ceci permet à chaque neurone des couches inférieures d'être moins spécifique, puisque sa réponse sera croisée avec celle des autres neurones de sa couche.
- Exemple : Couche 1 – Sport; Couche 2 – Sport collectif;
Couche 3 – Soccer; Couche 4 – Impact de Montréal.



Réseaux de neurones profonds

- En général, pour des problèmes complexes, les réseaux profonds ont une efficacité beaucoup plus élevée que les réseaux peu profonds.
- Par exemple, les couches cachées inférieures modélisent des structures de bas niveau (par exemple, des traits aux formes et aux orientations variées). Les couches caches intermédiaires combinent ces structures de bas niveau pour modéliser des structures de niveau intermédiaire (par exemple, des carrés et des cercles). Les couches cachées supérieures et la couche de sortie associent ces structures intermédiaires pour modéliser des structures de haut niveau (par exemple, des visages).

Réseaux de neurones profonds

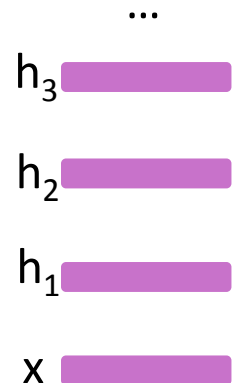


Les couches cachées inférieures modélisent des structures de bas niveau (par exemple, des traits aux formes et aux orientations variées). Les couches caches intermédiaires combinent ces structures de bas niveau pour modéliser des structures de niveau intermédiaire (par exemple, des carrés et des cercles). Les couches cachées supérieures et la couche de sortie associent ces structures intermédiaires pour modéliser des structures de haut niveau (par exemple, des visages).



Réseaux de neurones convolutifs

- Les réseaux de neurones convolutifs (CNN, Convolutional Neural Network) se basent sur le principe des réseaux de neurones profonds (d'ailleurs ce sont des réseaux profonds):
 - Apprendre des niveaux de représentation de plus en plus abstraits par mise en relation des niveaux inférieurs.



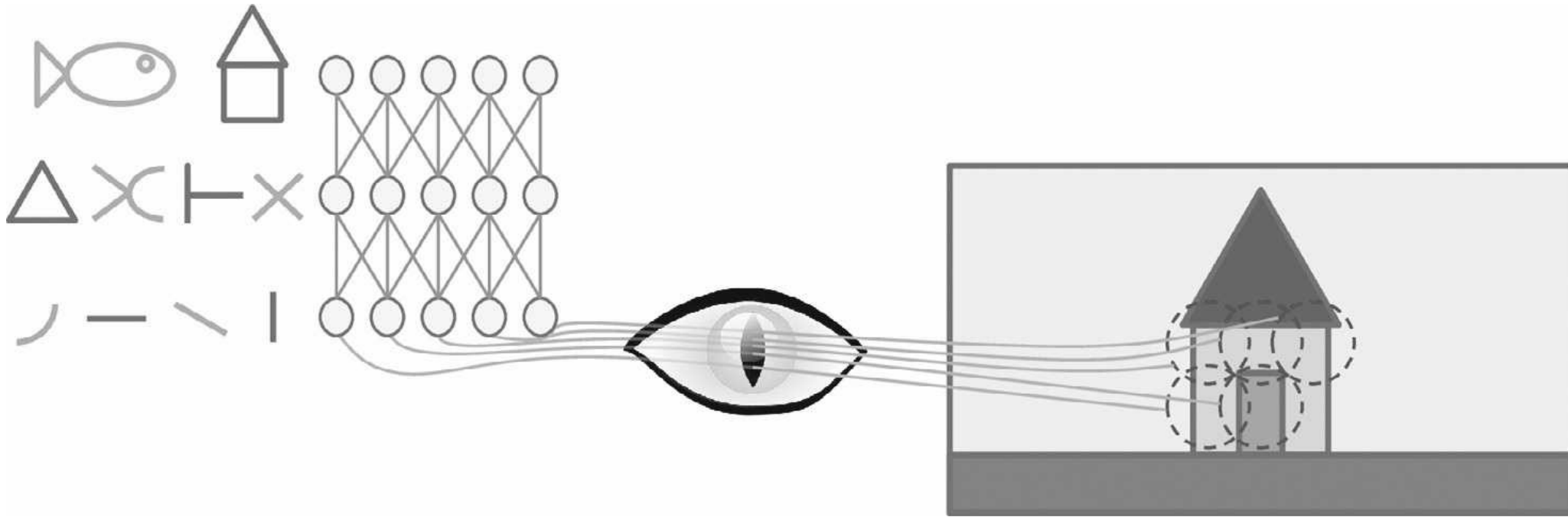


Réseaux de neurones convolutifs

- Les CNN sont apparus à la suite de l'étude du cortex visuel du cerveau et ils ont été initialement utilisés dans la reconnaissance d'images.
- Les CNN ont été capables de performances surhumaines sur des tâches visuelles complexes. Ils sont au cœur des services de recherche d'images, des voitures autonomes, des systèmes de classification automatique des vidéos, etc.*

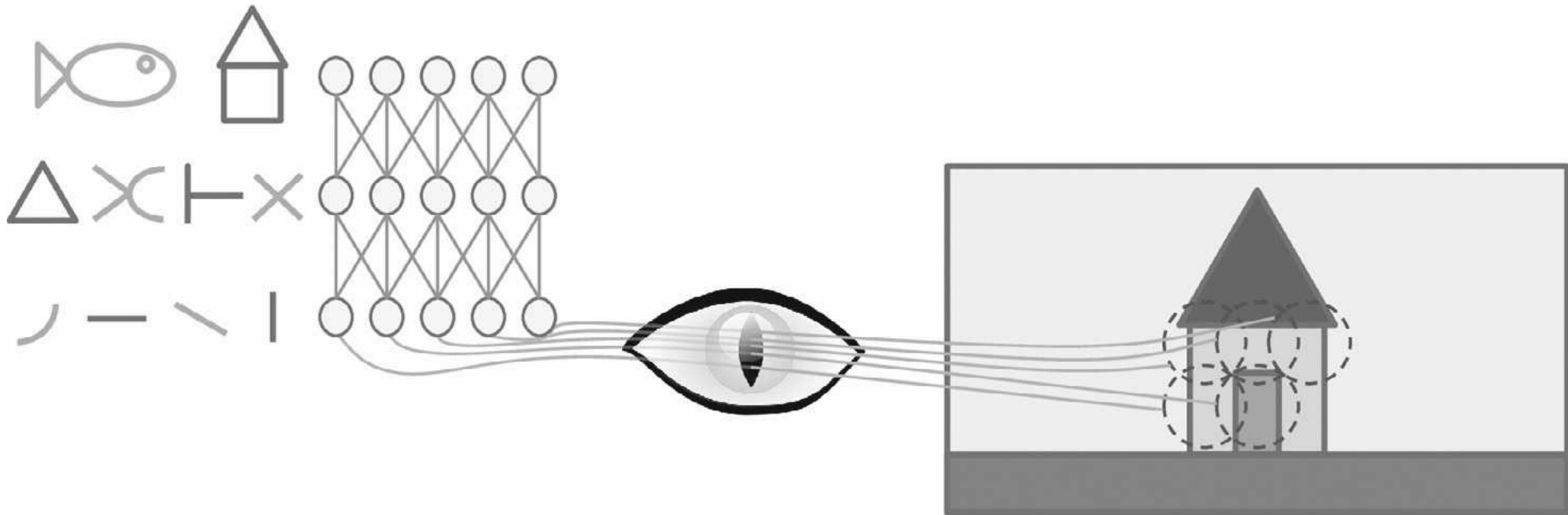
*Source : Aurélien Géron. Deep Learning avec Keras et TensorFlow: Mise en œuvre et cas concrets. Dunod, 2020.

L'architecture du cortex visuel*



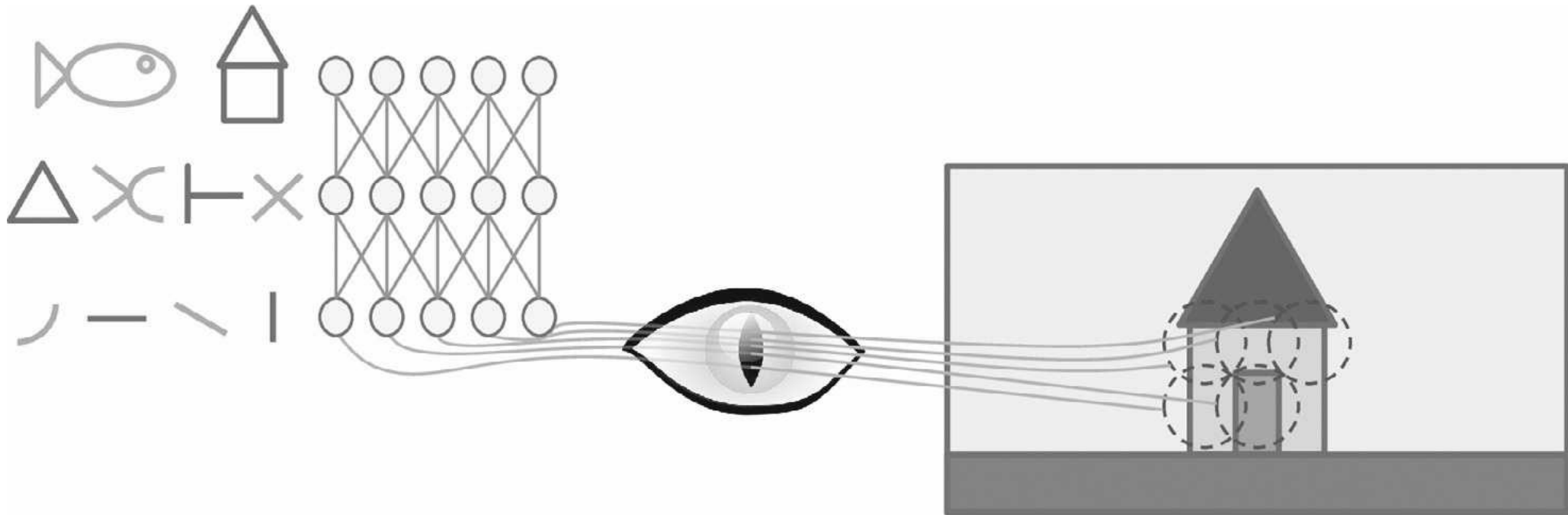
En 1958 et en 1959, David H. Hubel et Torsten Wiesel ont mené une série d'expériences sur des chats (et, quelques années plus tard, sur des singes), apportant des informations essentielles sur la structure du cortex visuel (en 1981, ils ont reçu le prix Nobel de physiologie ou médecine pour leurs travaux). Ils ont notamment montré que de nombreux neurones du cortex visuel ont un petit champ récepteur local et qu'ils réagissent donc uniquement à un stimulus visuel qui se trouve dans une région limitée du champ visuel (voir la figure, sur laquelle les champs récepteurs locaux de cinq neurones sont représentés par les cercles en pointillé). Les champs récepteurs des différents neurones peuvent se chevaucher et ils couvrent ensemble l'intégralité du champ visuel.

L'architecture du cortex visuel*



Les neurones biologiques du cortex visuel répondent à des motifs spécifiques dans de petites régions du champ visuel appelées champs récepteurs; au fur et à mesure que le signal visuel traverse les modules cérébraux consécutifs, les neurones répondent à des motifs plus complexes dans des champs récepteurs plus larges.*

L'architecture du cortex visuel*

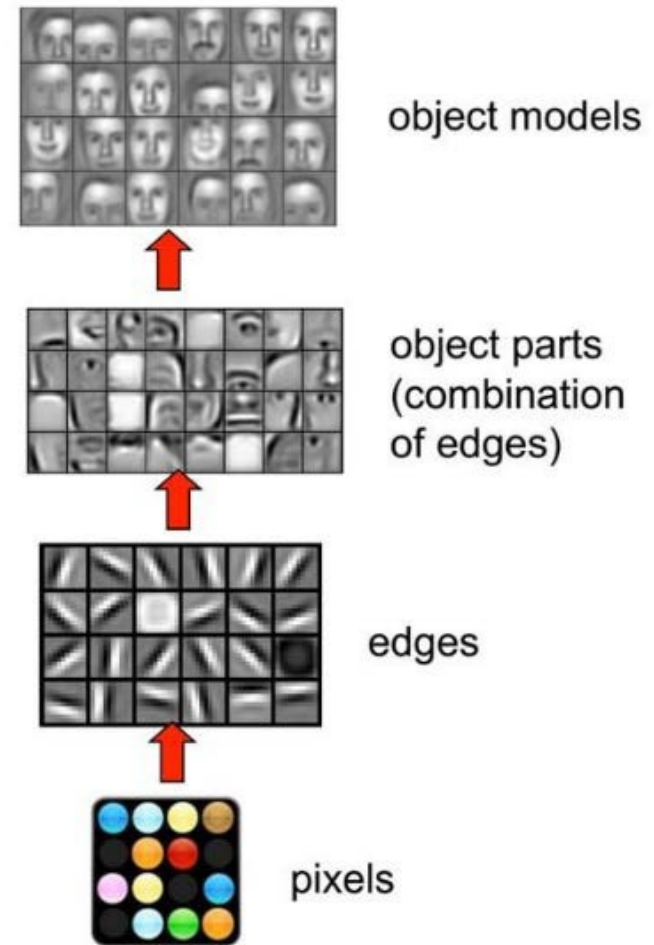


Les champs récepteurs des différents neurones peuvent se chevaucher et ils couvrent ensemble l'intégralité du champ visuel.*

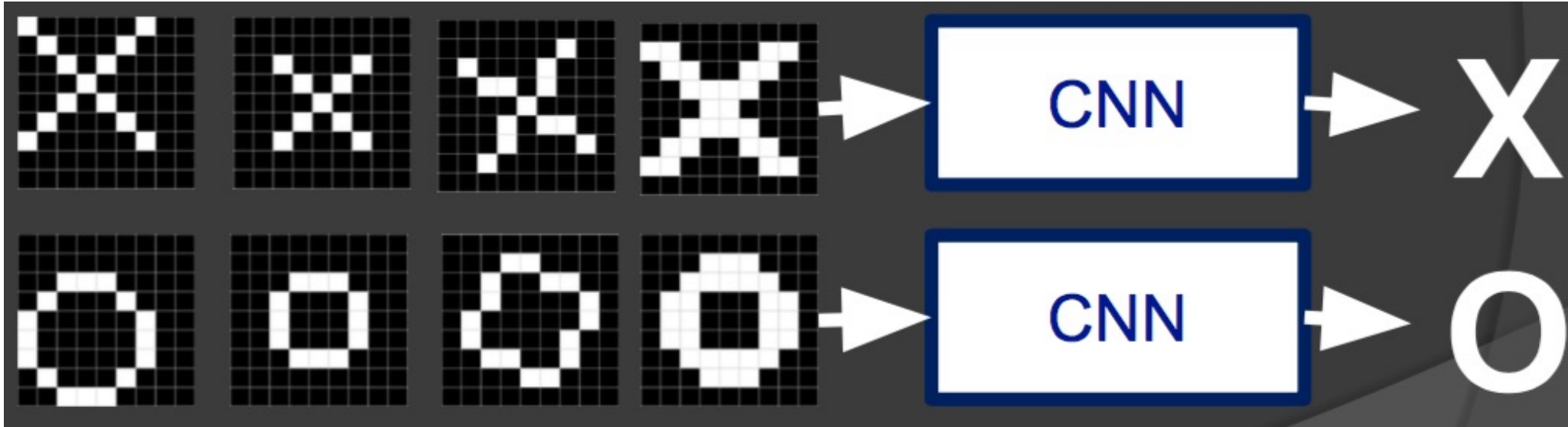
Les études du cortex visuel sont à l'origine du neocognitron, qui a progressivement évolué vers ce que nous appelons aujourd'hui réseau de neurones convolutif.

Réseaux de neurones convolutifs

- Les CNN sont des architectures hiérarchiques qui peuvent apprendre des hiérarchies de caractéristique à plusieurs niveaux.
 - Exemple : le CNN reçoit des images en entrée et peut extraire des représentations automatiquement.
 - Ces représentations sont faites hiérarchiquement, où les pixels forment des bords, les bords forment des motifs, les motifs combinés forment les parties, les parties forment les objets et les objets forment les scènes.



Classification d'images



- CNN : comment apprendre à reconnaître une image?

Classification d'images

- La machine ne voit que des données numériques



What We See

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See



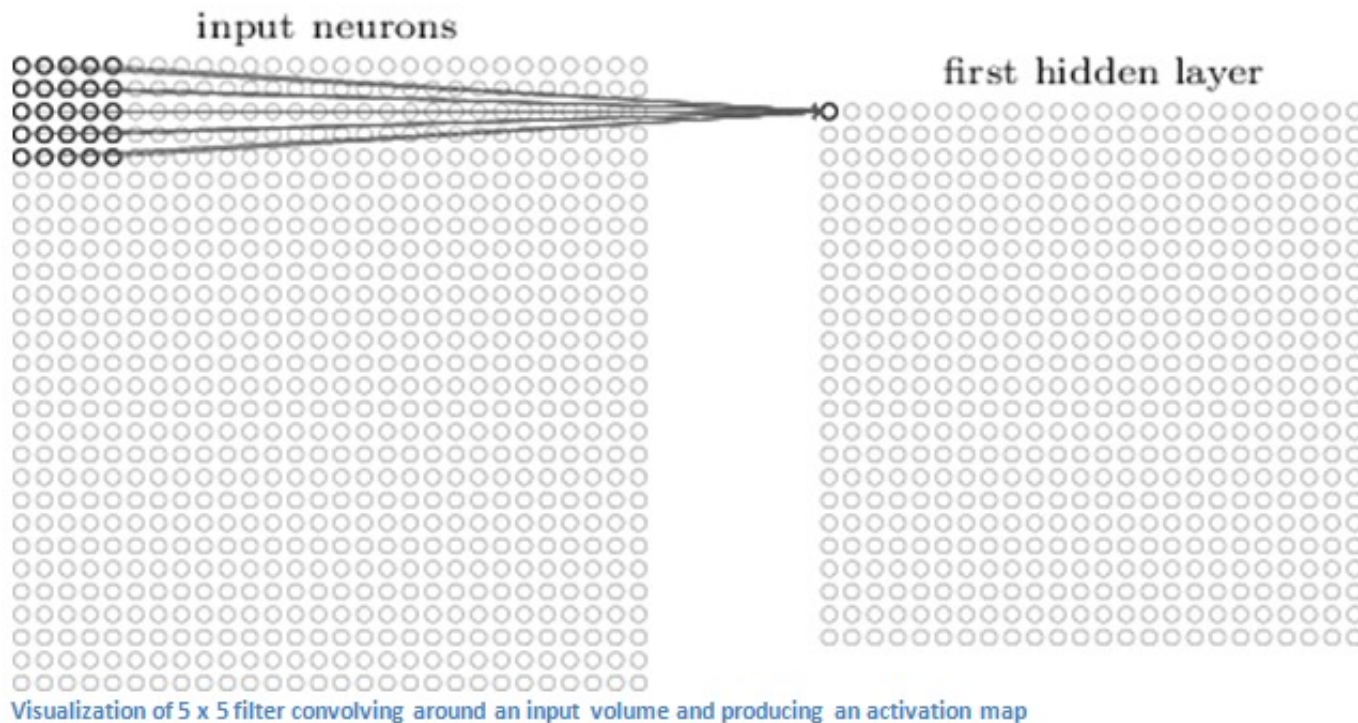
Classification d'images

Traiter une image est un défi pour un réseau de neurones.

- Si on prend un neurone par pixel
 - 3072 neurones pour une image de $32 \times 32 \times 3$
 - 120 000 neurones pour une image de $200 \times 200 \times 3$

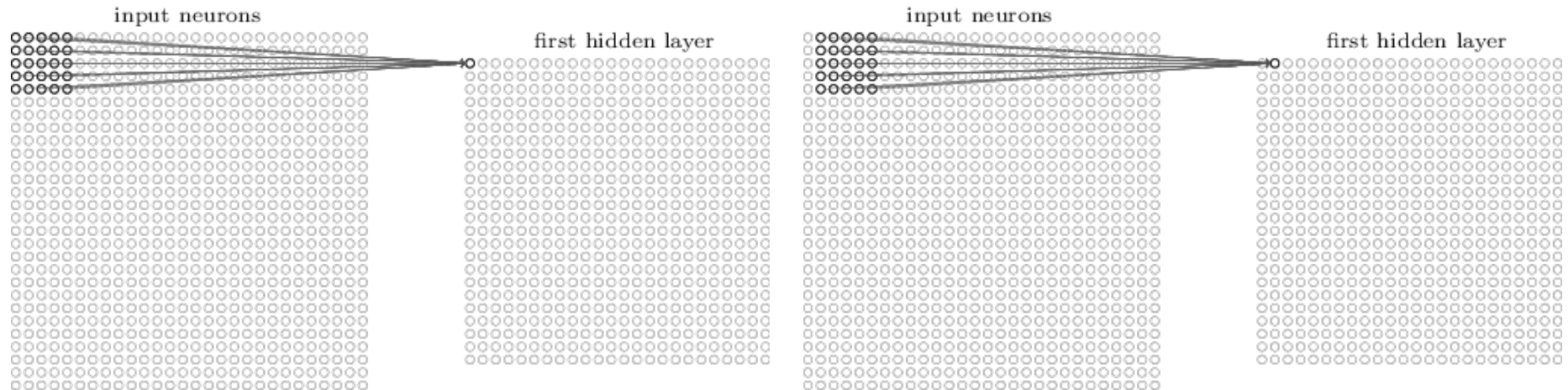
Les filtres

Un CNN utilise des filtres d'apprentissage appliqué sur toute l'entrée (une image), générant des *maps* de caractéristiques. Chaque filtre détecte une caractéristique particulière.



Les filtres

Un CNN utilise des filtres d'apprentissage appliqué sur toute l'entrée (une image), générant des *maps* de caractéristiques. Chaque filtre détecte une caractéristique particulière.

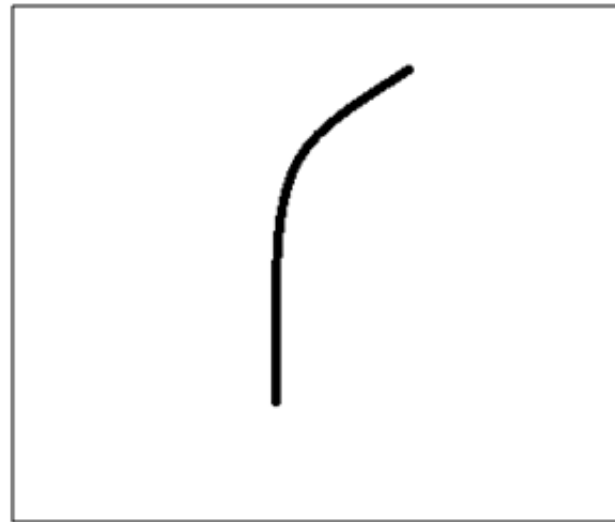


Les filtres

Exemple de filtre

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

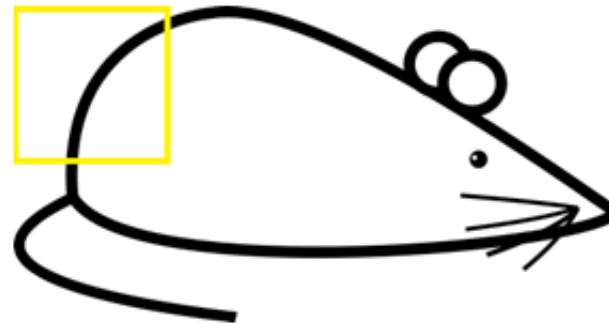
Pixel representation of filter



Visualization of a curve detector filter



Original image



Visualization of the filter on the image

Les filtres

Exemple de filtres



- Un filtre sert à faire ressortir certaines caractéristiques d'une image (couleur, contour, luminosité, netteté, etc.)

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Les filtres

Exemple de filtre



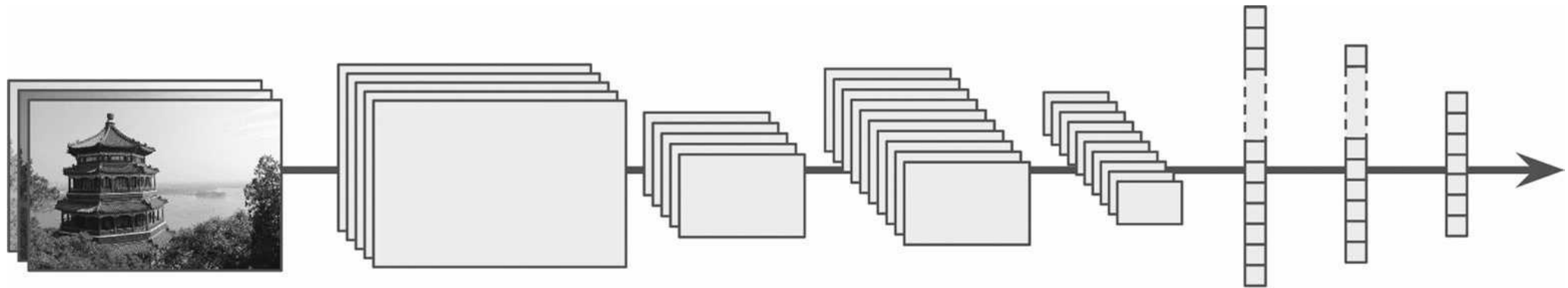
*

0	-1	0
-1	4	-1
0	-1	0



Architecture typique d'un CNN

Le CNN a, dans son architecture typique, des couches de convolution, des couches de sous-échantillonnage et des couches complètement connectées .





Les couches d'un CNN

- Il y a plusieurs types de couches principales pour construire un réseau CNN standard. Les couches INPUT, COUV, POOLING, RELU et FC (Fully Connected).
- Les couches s'empilent un à la suite de l'autre pour former l'architecture du CNN.
- Les couches peuvent aussi se répéter, sauf pour la couche INPUT et FC.
- D'ailleurs les formes les plus communes d'une architecture CNN empilent quelques couches CONV-RELU-POOL et répète ce modèle jusqu'à ce que l'image ait été fusionnée à une petite taille. Ensuite on passe à la couche FC.

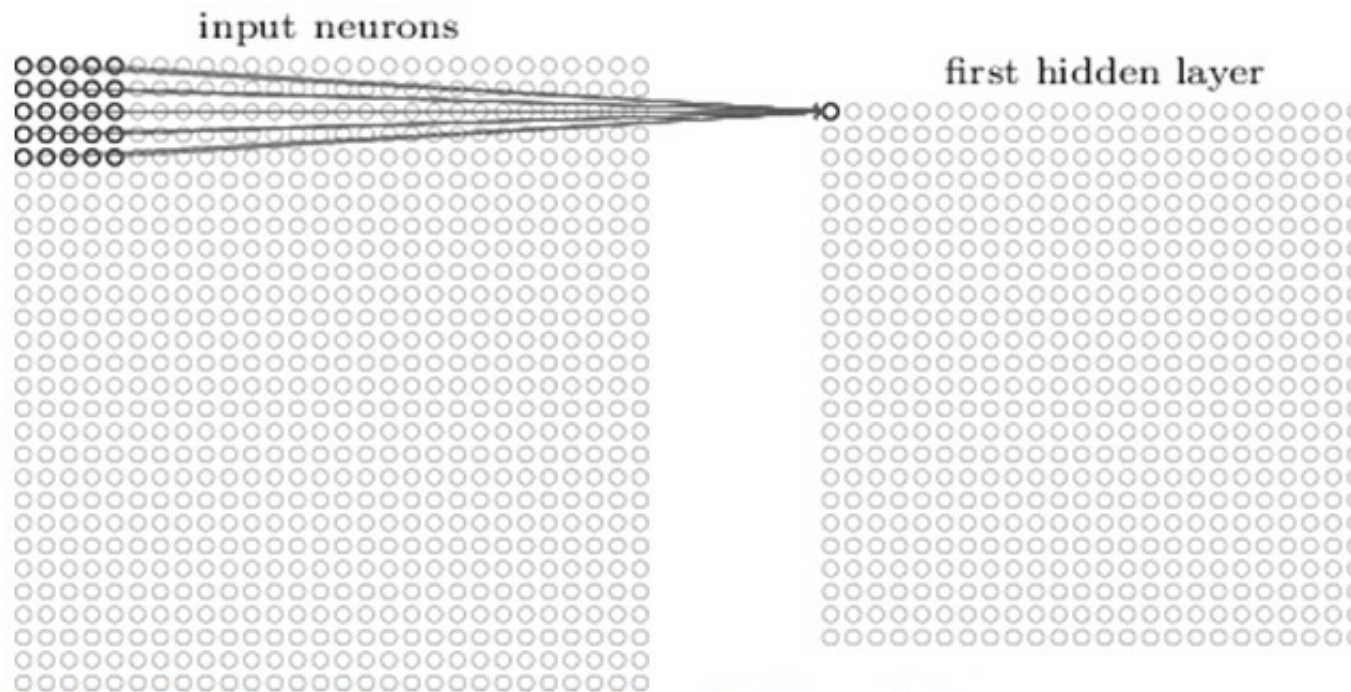


La couche d'entrée

Cette couche prend les pixels bruts de l'image et les transforme en une matrice 3D (Hauteur, Largeur et les trois canaux de couleurs Rouge, Vert, Bleu).

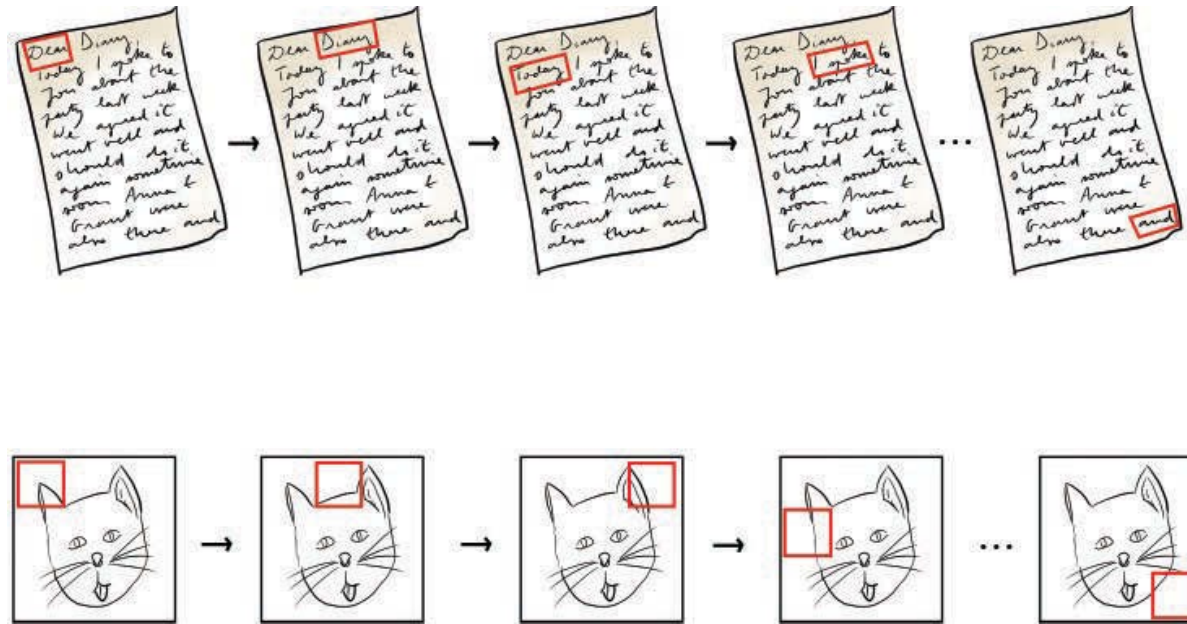
Couches de convolution

- La couche convolution est la base de ce type de réseau neurone. Elle effectue la majeure partie du traitement et des calculs.
- Cette couche fait la recherche de patrons en utilisant des filtres à l'intérieur du volume d'entrée et fabrique une matrice pour chaque filtre. Chaque filtre est de petite taille, mais la recherche effectuée avec les filtres s'étend à travers la totalité du volume d'entrée.



Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

Le filtre d'une couche de convolution



Lors de la lecture d'une page, nous commençons en haut à gauche au coin et lisons à droite. Chaque fois que nous atteignons la fin d'une ligne, nous progressons vers la ligne suivante. De cette façon, nous atteignons finalement le coin inférieur droit, lisant ainsi tous les mots sur la page. De manière analogue, le filtre d'une couche de convolution commence par une petite fenêtre de pixels dans le coin supérieur gauche d'une image donnée. De la rangée du haut vers le bas, le filtre scanne de gauche à droite, jusqu'à ce qu'il atteigne finalement la fin au coin à droite balayant ainsi tous les pixels de l'image.

La convolution – Exemple



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

La convolution – Exemple

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

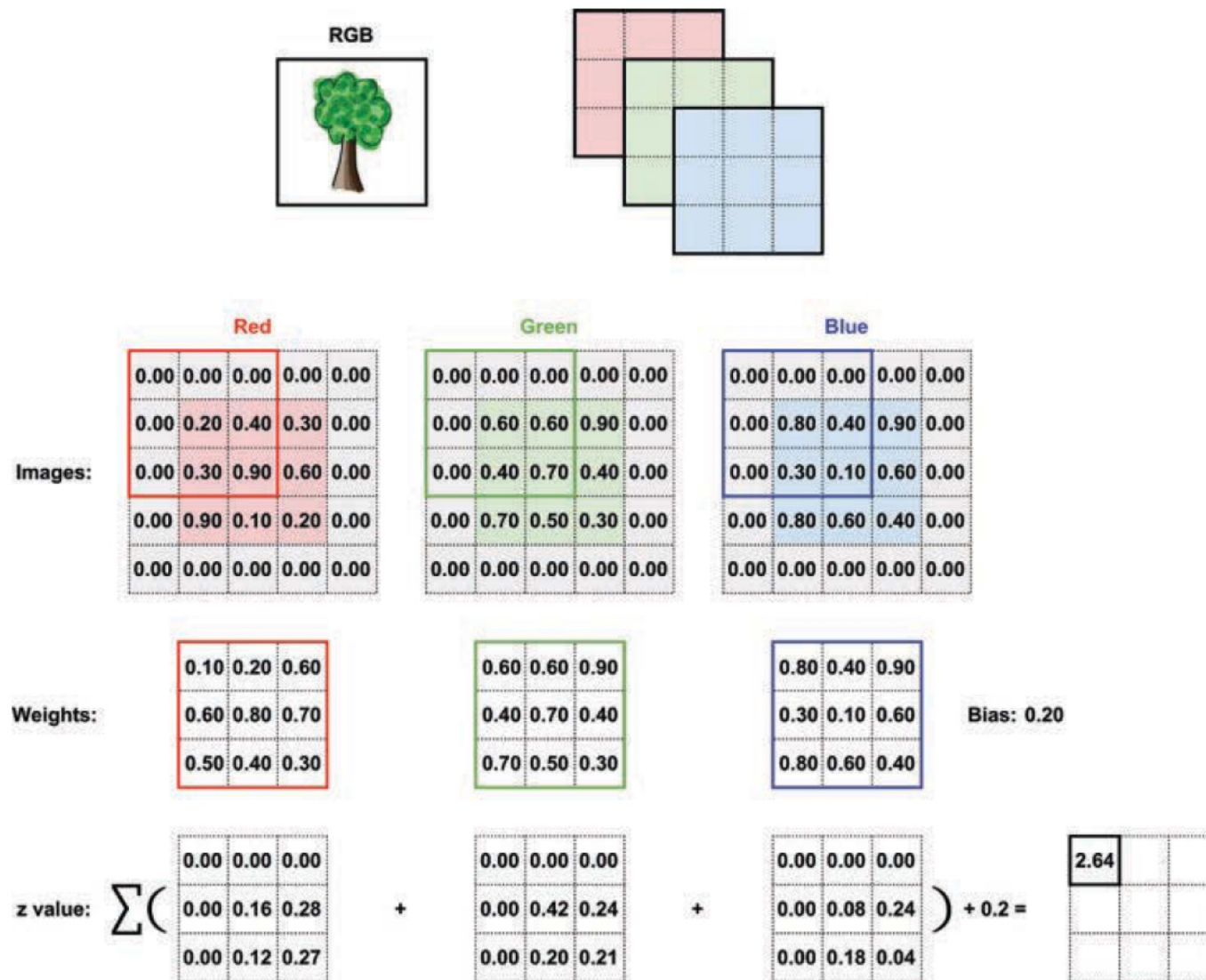


1	-1	-1
-1	1	-1
-1	-1	1

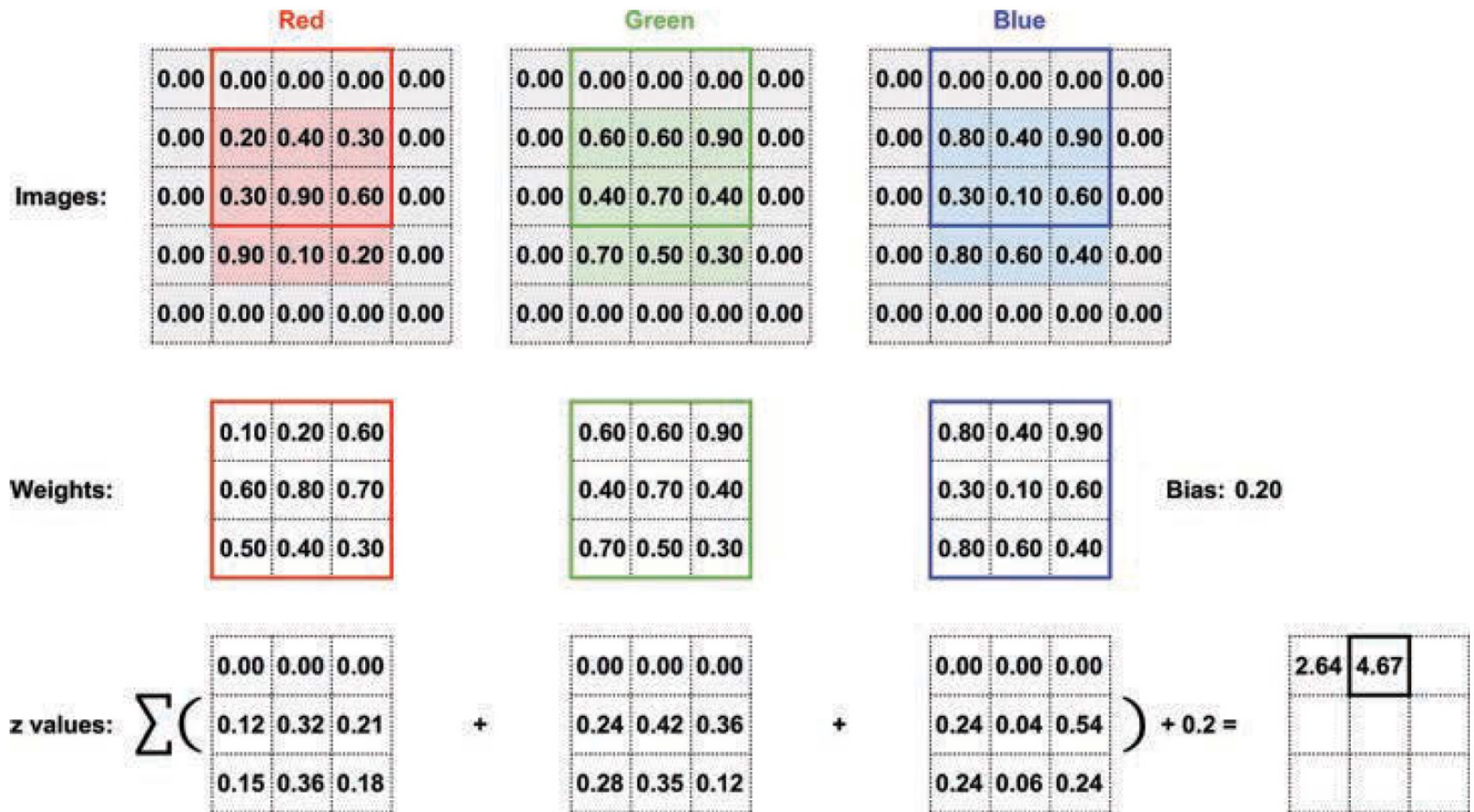
=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

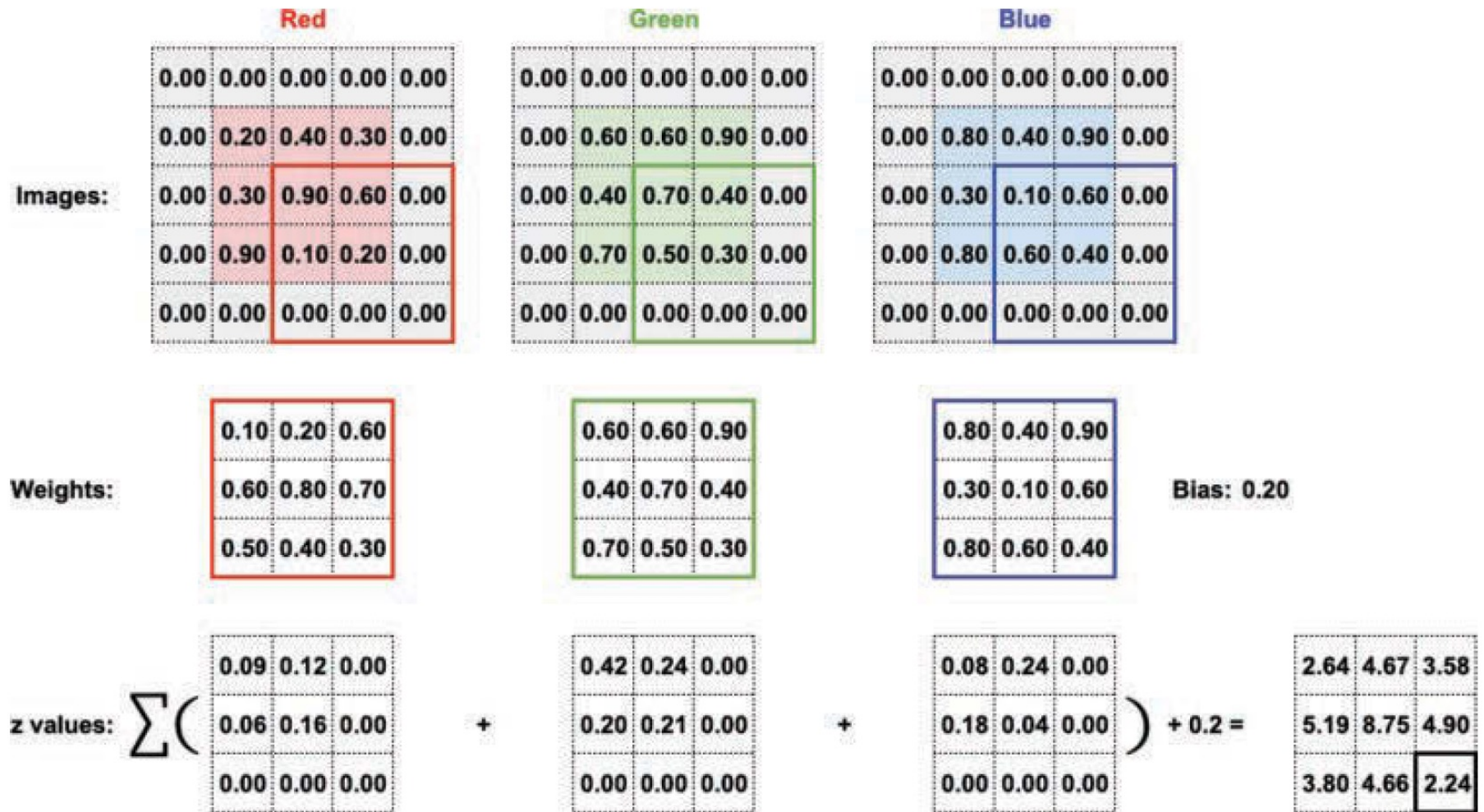
La convolution – Exemple : une image en couleur avec Zero padding



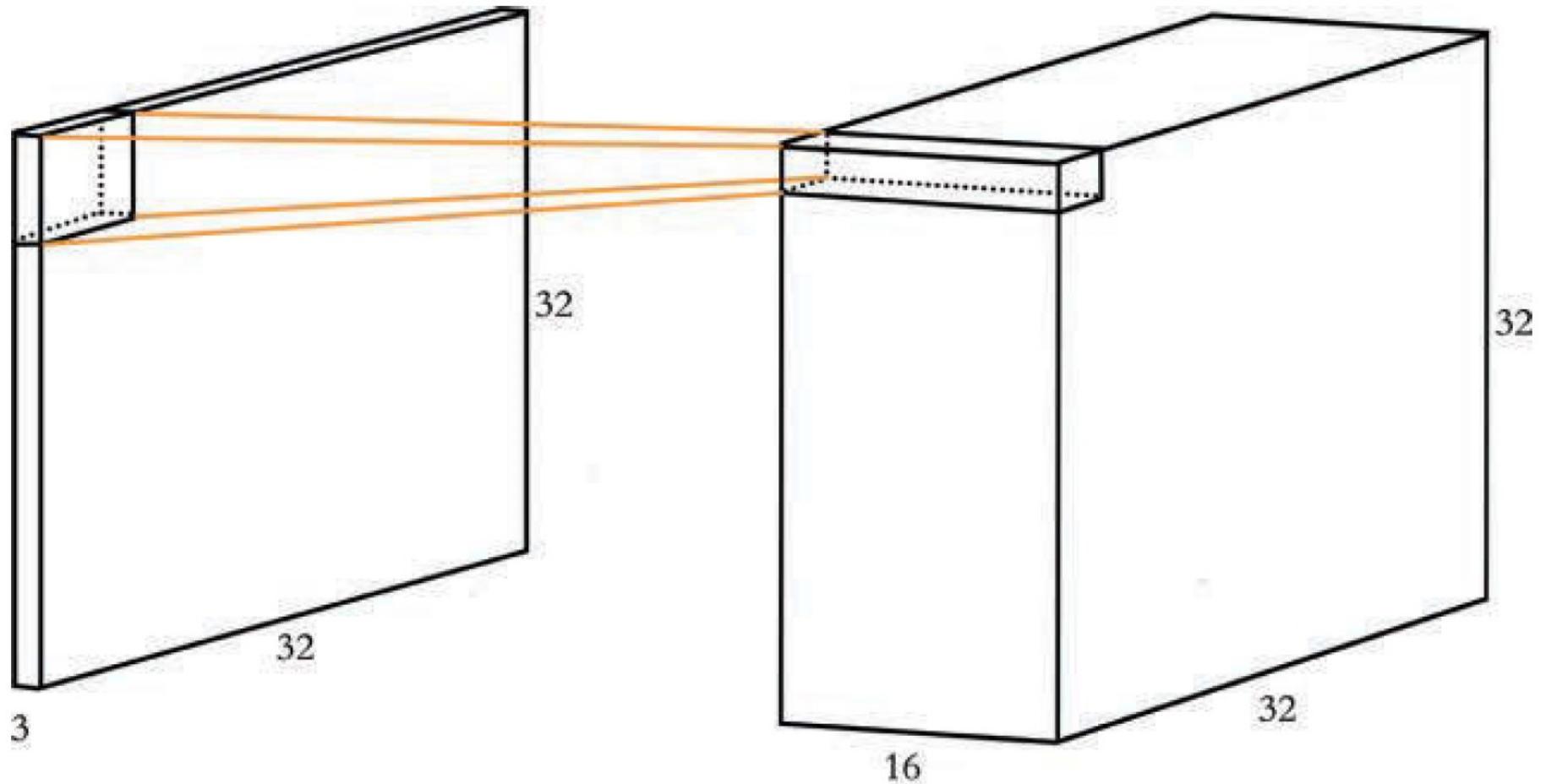
La convolution – Exemple : une image en couleur avec Zero padding



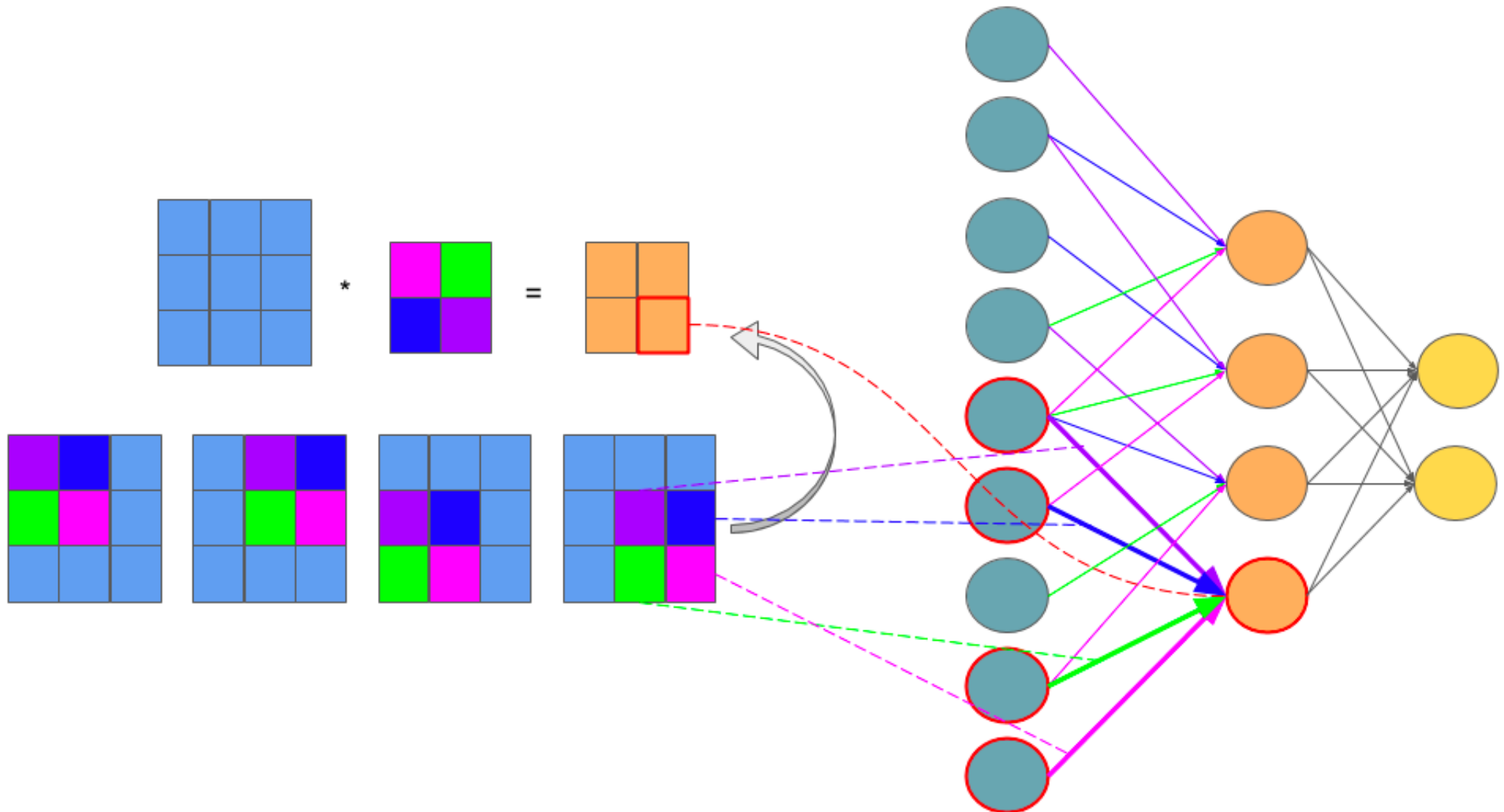
La convolution – Exemple : une image en couleur avec Zero padding



La convolution – Exemple dans une image en couleur avec Zero padding



La convolution – Représentation comme un réseau de neurone





Couches de convolution

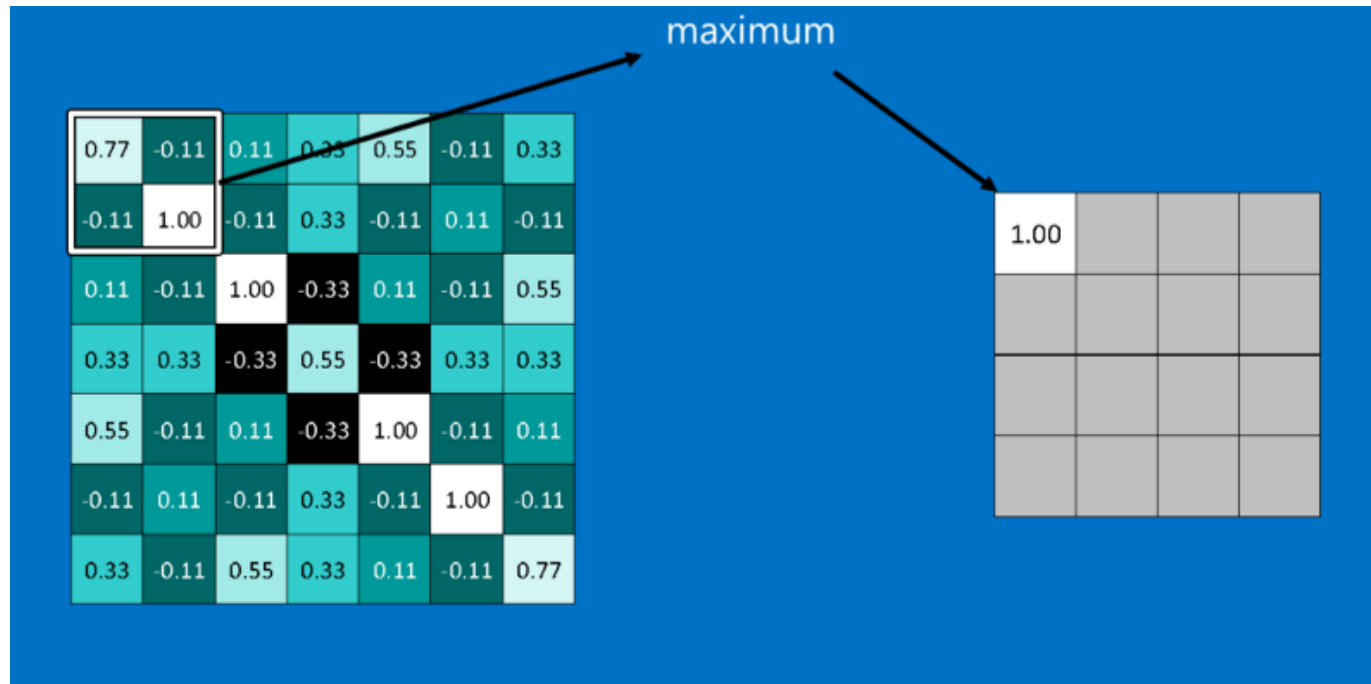
- L'empilement des couches de convolution permet au réseau de se focaliser sur des caractéristiques de bas niveau dans la première couche cachée, puis de les assembler en caractéristiques de plus haut niveau dans la couche cachée suivante, etc.
- Cette structure hiérarchique est récurrente dans les images réelles et c'est l'une des raisons des bons résultats des CNN pour la reconnaissance d'images.



La couche Pooling

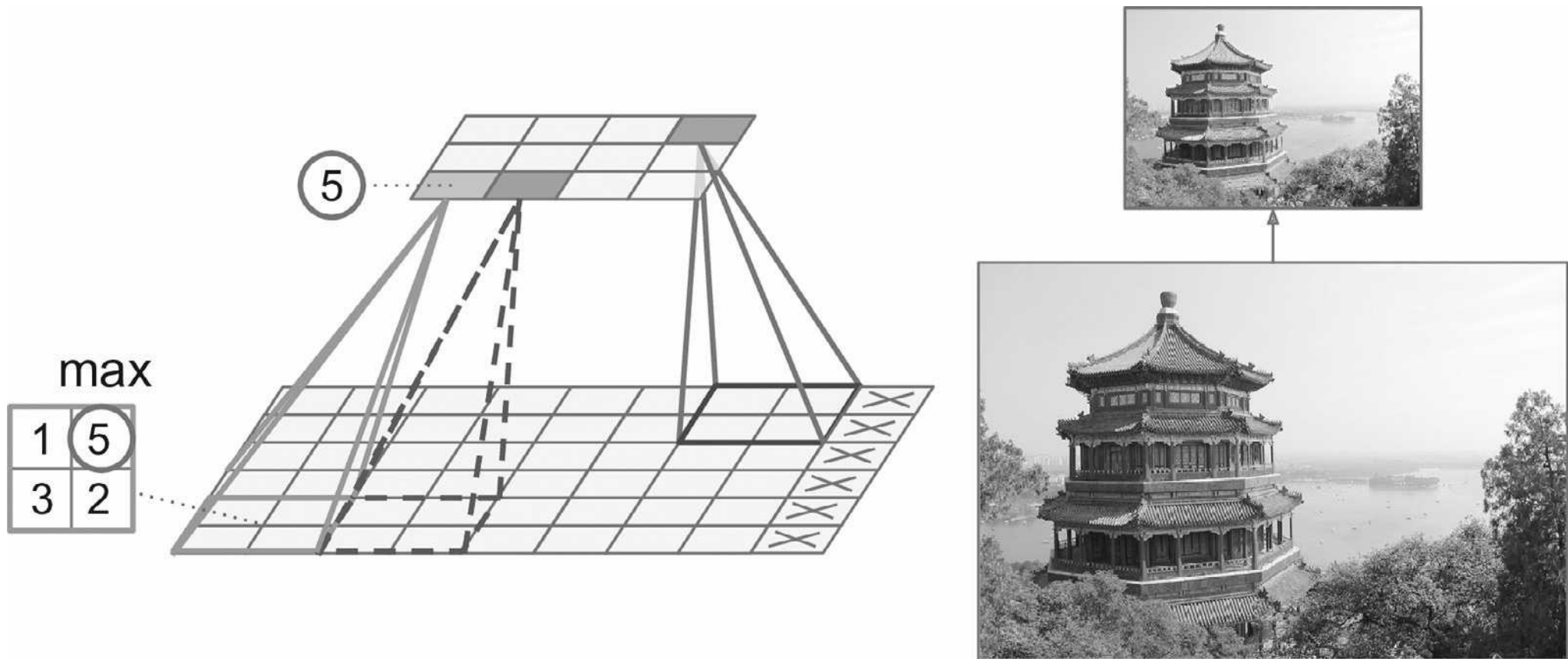
- Le pooling est une méthode permettant de prendre une large image et d'en réduire la taille tout en préservant les informations les plus importantes → échantillonnage des valeurs les plus significatives.
- La fonction de cette couche est de réduire progressivement la taille spatiale de la représentation afin de réduire la quantité de calcul dans le réseau et l'utilisation de la mémoire.
- Il est fréquent d'insérer une couche de POOLING entre les couches CONV successives dans une architecture CNN.

Pooling - Exemple



- Exemple : utiliser un filtre de taille 2x2 appliqué avec un déplacement de deux. Lors de chaque pooling, on compare les nombres du filtre et on garde celui le plus représentatif (max pooling par exemple).
- Remarque : le pooling maximum conserve uniquement les caractéristiques les plus fortes, écartant les moins pertinentes. La couche suivante travaille donc sur un signal plus propre. Par ailleurs, le pooling maximum offre une invariance de translation plus importante que le pooling moyen et demande des calculs légèrement moins intensifs.

Pooling - un autre exemple



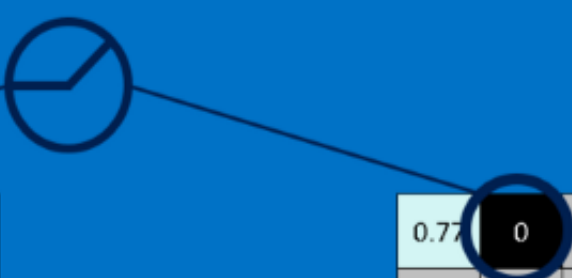
- Exemple : utiliser un filtre de taille 2x2 appliqué avec un déplacement de deux. Lors de chaque pooling, on compare les nombres du filtre et on garde celui le plus représentatif (max pooling par exemple).

Figure tirée de :

Aurélien Géron. Deep Learning avec Keras et TensorFlow: Mise en œuvre et cas concrets. Dunod, 2020. 35

La couche ReLU

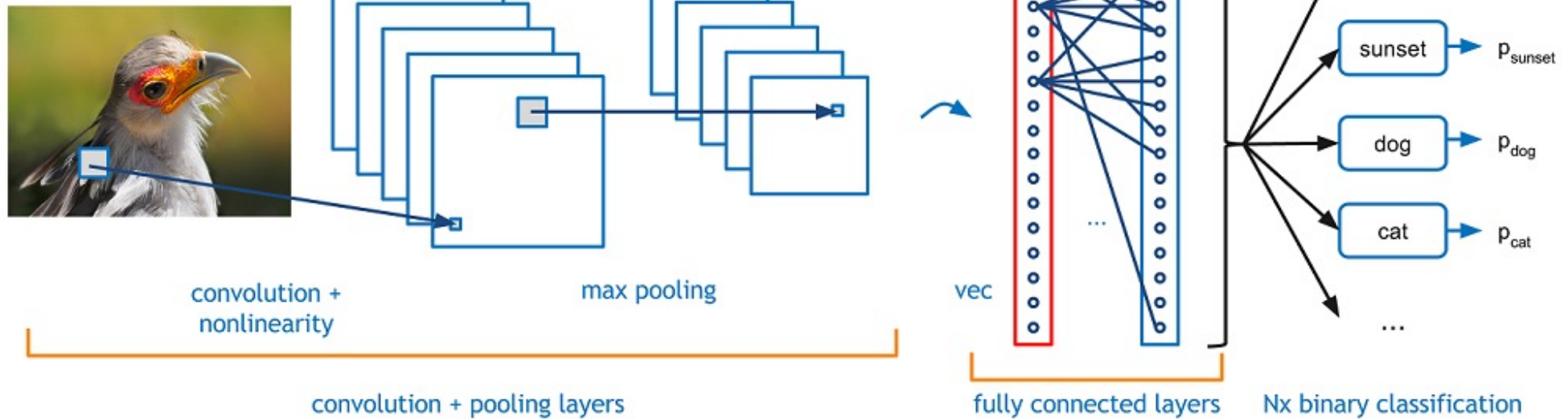
- Unité Rectifié Linéaire (ReLU)
- Consiste à remplacer les valeurs négatives par un zéro.
- Le but est d'introduire la non-linéarité (la convolution est une opération linéaire. On a besoin donc de la non-linéarité, sinon, 2 couches de convolution ne sont pas nécessairement plus meilleures qu'une seule.).



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0					

La couche connectée (FC)

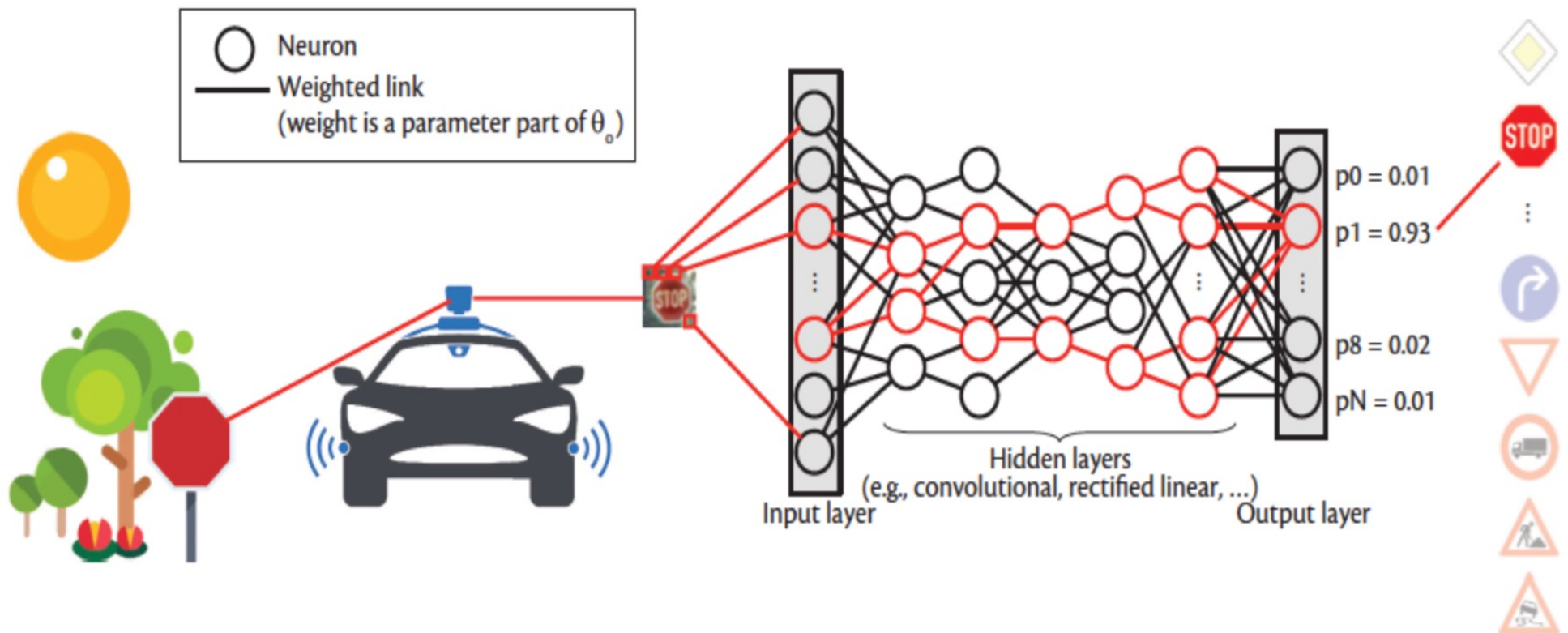


Il s'agit en fait d'un perceptron multicouche où tous les neurones de sortie sont connectés à tous les neurones d'entrée. Les classes sont calculées dans cette couche. Contrairement à d'autres couches du réseau, elle (la couche FC) évolue avec l'apprentissage. Les neurones dans une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente, comme on le voit dans les réseaux neuronaux réguliers.

Exemple d'application

Traffic-Sign Recognition

[Patrick McDaniel et al., IEEE Security & Privacy, 2016]

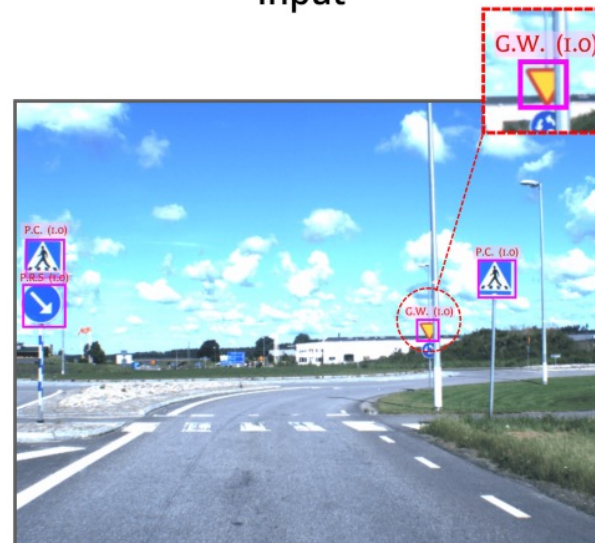


Exemple d'application

Traffic-Sign Recognition



Input



Output

Source : Y. Yang et al. "Efficient Traffic-Sign Recognition with Scale-aware CNN", BMVC 2017.

Exemple d'application

Traffic-Sign Recognition

Cas idéal



Source : S. Hijazi et al. " Using Convolutional Neural Networks for Image Recognition", 2015

Exemple d'application

Traffic-Sign Recognition

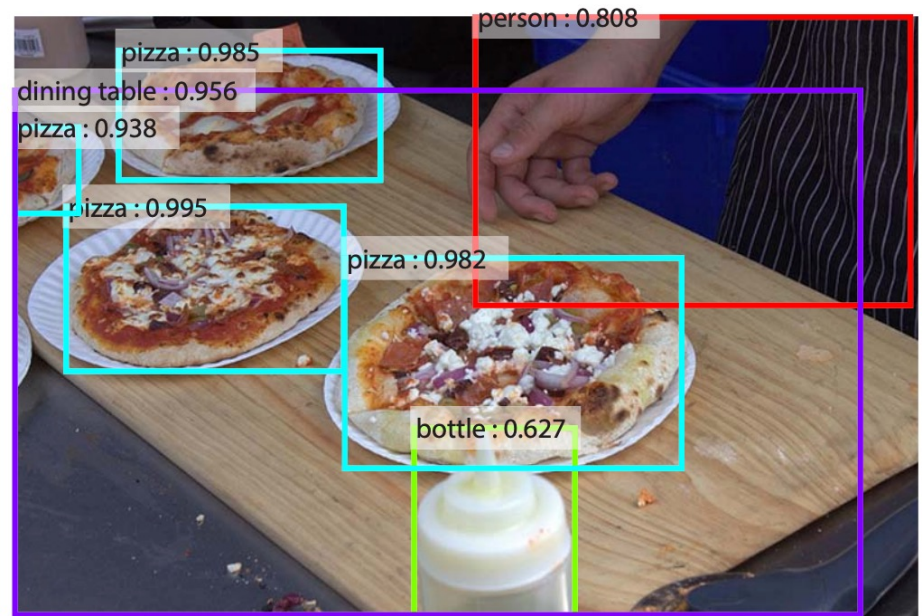
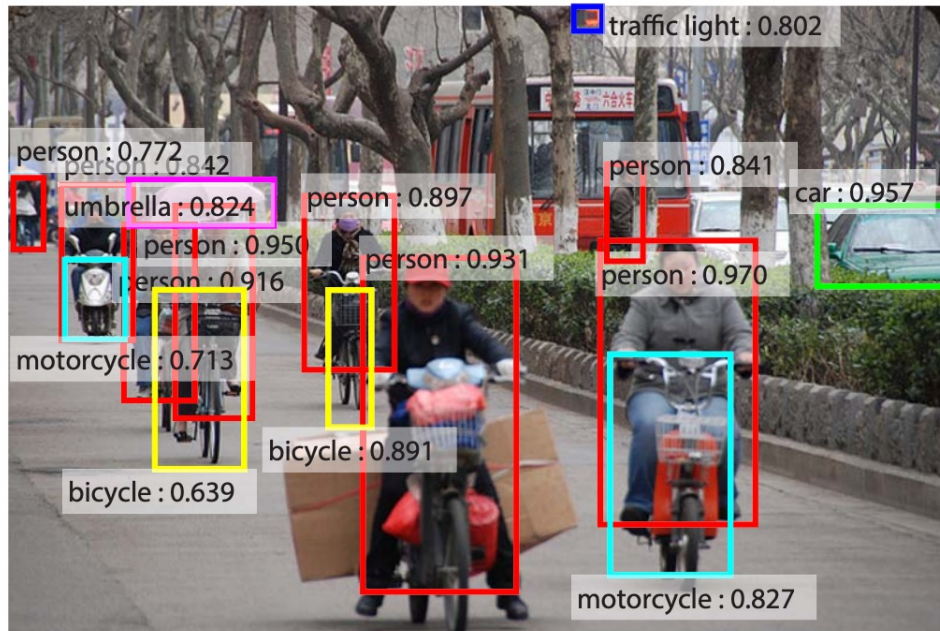
Données réelles



Source : S. Hijazi et al. “ Using Convolutional Neural Networks for Image Recognition”, 2015

Exemple d'application

Segmentation sémantique



Source : Ren et al. : “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 2015



Réseau à convolution

- Voir le tutoriel de Brandon Rohrer

- How it Works:

Convolutional neural Networks

- Disponible dans moodle