

# Fiche de lecture : Réseau neuronal convolutif

Chaolei Cai

*Université Paris Vincennes St-Denis*

*UFR mathématiques, informatique,  
technologies sciences de l'information*

L3 Informatique

April 27, 2020

# Contents

## 1 Avant-propos

Ce document est ma fiche de lecture sur le réseau neuronal convolutif (Convolutional Neural Network or ConvNet), il fait partie de la notation pour le cours de programmation pour l'intelligence artificiel enseigné par M.Jean-Jacques Mariage aux L3 de la licence informatique.

Dans le plan général, je vais vous présenter le néocognitron, un ancêtre des CNN.

Dans un second temps, je vais m'intéresser à une publication d'Alex Krizhevsky, Ilya Sutskever et Geoffrey E.Hinton sur la classification d'image via un Deep Convolutional Neural Networks.

Puis je reviendrai à des sujets plus récente "Batch normalization" proposé par Sergey Ioffe et Christian Szegedy, il s'agit de la méthode de regularization la plus moderne sur les CNN. Enfin je terminerai sur un article de Amir Rosenfeld, Richard Zemel et John K. Tsotsos qui pointe les limites rencontrées par les modèles CNN les plus moderne.

## 2 Néocognitron

### 2.1 Présentation de l'auteur

M.Fukushima est un chercheur en informatique connue pour ses travaux dans le domaine du réseau de neurone artificiel et du deep learning. Le plus célèbre étant son modèle de Néocognitron publié en 1980 [**Fukushima:1**].

### 2.2 Présentation de l'article

Le néocognitron est un modèle élaboré depuis le modèle de cognitron publié par le même auteur en 1975. Il s'agit d'un réseau de neurone multi-couche auto-organisatrice. L'apprentissage n'est pas supervisée, c'est à dire que les données ne sont pas étiquetées. Il s'inspire notamment des travaux de Hubel et Wiesel sur le système visuel animal (chat). [**HubelWiesel:2**]. A nos jour, la recherche a bien évidemment avancé depuis 1959, Le model d'Hubel et Wiesel n'est plus tout à fait exacte mais reste intéressant dans les grandes lignes. Un des professeur en physiologie m'avait d'ailleurs présenté

ces travaux quand j'étudiais encore à la licence Science pour la santé à Paris Descartes quelques années auparavant.

Le point important à retenir c'est que Hubel et Wiesel propose l'idée qu'un réseau de neurone reste quand même traditionnel.

neurone est constitué de cellule "Simple" appelée "S-cells" et de cellule "Complex" appelée "C-cells".

Pour être tout à fait précis, le neocognitron ne reconnait pas l'objet complexe mais présente plutôt une capacité à reconnaître des patterns de stimuli précis après l'entraînement.

Notre sujet présente une organisation étalée sur plusieurs couches, dans l'idée, il faut retenir qu'il s'agit d'un réseau organisé neurone reste quand même traditionnel.

en cascade et hiérarchisé.

## **2.3 Présentation de la structure**

Au niveau de l'entrée, nous pouvons trouver des cellules photoreceptrices chargées de transmettre l'information. Si nous devons faire un rapprochement avec le réel, cela correspond aux rétines sensibles à différents stimuli lumineux.

Dans la suite intervient une succession de modules, un module est constitué d'une ou plusieurs couches de neurones. La première constituée de cellules simples, l'auteur appelle cette couche "S-layer", puis arrivent les couches complexes constituées à leur tour de cellules complexes voire hyper-complexes, de la même manière, ils portent le nom de "C-layer". Il est intéressant de noter que seules les synapses partant vers une "S-cell" ont la propriété de plasticité. La plasticité en neurosciences est le terme pour désigner la capacité d'un neurone à former et modifier ses connexions vers d'autres neurones via les synapses.

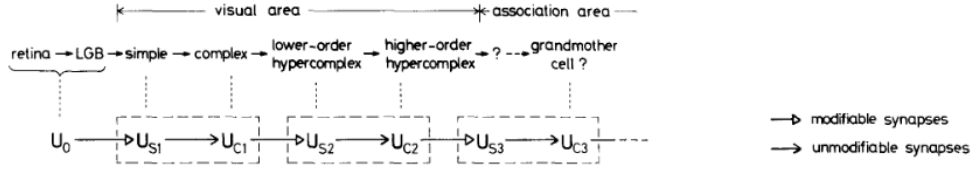


Figure 1: Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

Comme vous pouvez le voir sur ce schéma,  $U_0$  est notre couche d'entrée, par la suite intervient une cascade de module constitué par différentes couches de cellules. Au sein d'une même module, les liens entre les neurones de différentes couches sont fixes, alors qu'à la jointure des modules, ces liens sont modifiables et peuvent être amenés à évoluer au fur et à mesure de l'apprentissage. Tout cela nous permet de faire un parallèle avec le fonctionnement du cortex visuel et associatif.

$$u_{Sl}(k_l, \mathbf{n}) = r_l \cdot \varphi \left[ \frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v \in S_l} a_l(k_{l-1}, \mathbf{v}, k_l) \cdot u_{Cl-1}(k_{l-1}, \mathbf{n} + \mathbf{v})}{1 + \frac{2r_l}{1+r_l} \cdot b_l(k_l) \cdot v_{Cl-1}(\mathbf{n})} - 1 \right],$$

Figure 2: formule de la sortie d'une cellule S

D'après cette formule, la sortie d'une cellule S situé à la couche  $k$  d'un module  $l$  de notre réseau dépend essentiellement de  $al(k(l-1), v, kl)$  et de  $bl(kl)$  qui correspondent à l'efficacité des synapses excitatoire et inhibitoire. Il est possible de rendre la cellule plus ou moins sélective à une patterne d'excitation en variant  $rl$ .

En ce qui concerne le mécanisme d'auto-organisation, à chaque fois qu'une entrée arrive, des "S-cell" représentatives sont sélectionnées, et vont voir renforcé leur synapses d'entrée. Au contraire ceux qui ne manifeste aucun sensibilité ne sont pas renforcé. Au fur et à mesure de l'apprentissage, nous

verrons alors apparaître sur le plan cellulaire des colonnes, une succession de zone qui sont sensible à un même stimulus sur différente couche.

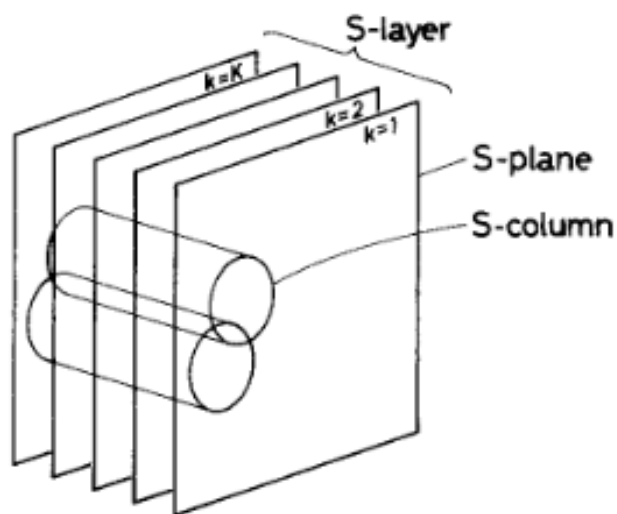


Figure 3: Relation between S-planes and S-columns within an S-layer

Ainsi, nous construisons un réseau qui dans les première niveau de profondeur ne va que reconnaître des pattern simple et élémentaire, et plus nous progressons dans le réseau, les éléments basic se regroupe en motifs, jusqu'à l'obtention d'un "objet".

Comme le schéma ci-dessous le montre, il y a la reconnaissance de trait basique puis dans les niveaux plus profondes, le motif de la lettre 'A' est reconstruite.

Ici, la connexion n'est pas total entre les différentes couche, il y a donc beaucoup moins d'information à traiter et la structure est plus légère. Au final, le réseau n'est pas si chargé, car les "S-columns" peuvent parfaitement s'entre croiser. Il est tout à fait possible que pour la lettre 'R' et 'B' partagent les même cellules comme ces 2 lettres sont proche au niveau de l'écriture.

Enfin, notre réseau offre une tolérance vis à vis des déformations, changement d'orientation et changement de position car quelque soit l'entrée présenté, il est divisé en sous composante puis reconstruite dans les profondeur. Nous pouvons déjà voir ici les prémices du CNN, nous pouvons aussi remarquer

que le réseau est neuronne reste quand même traditionnel.

à sens unique, sur les schémas ci-dessous par exemple, l'information va toujours de la gauche vers la droite, il n'y a qu'un seul mouvement, celui de feedfoward.

Alors que sur les modèles CNN moderne, nous pouvons trouver une étape de backpropagation.

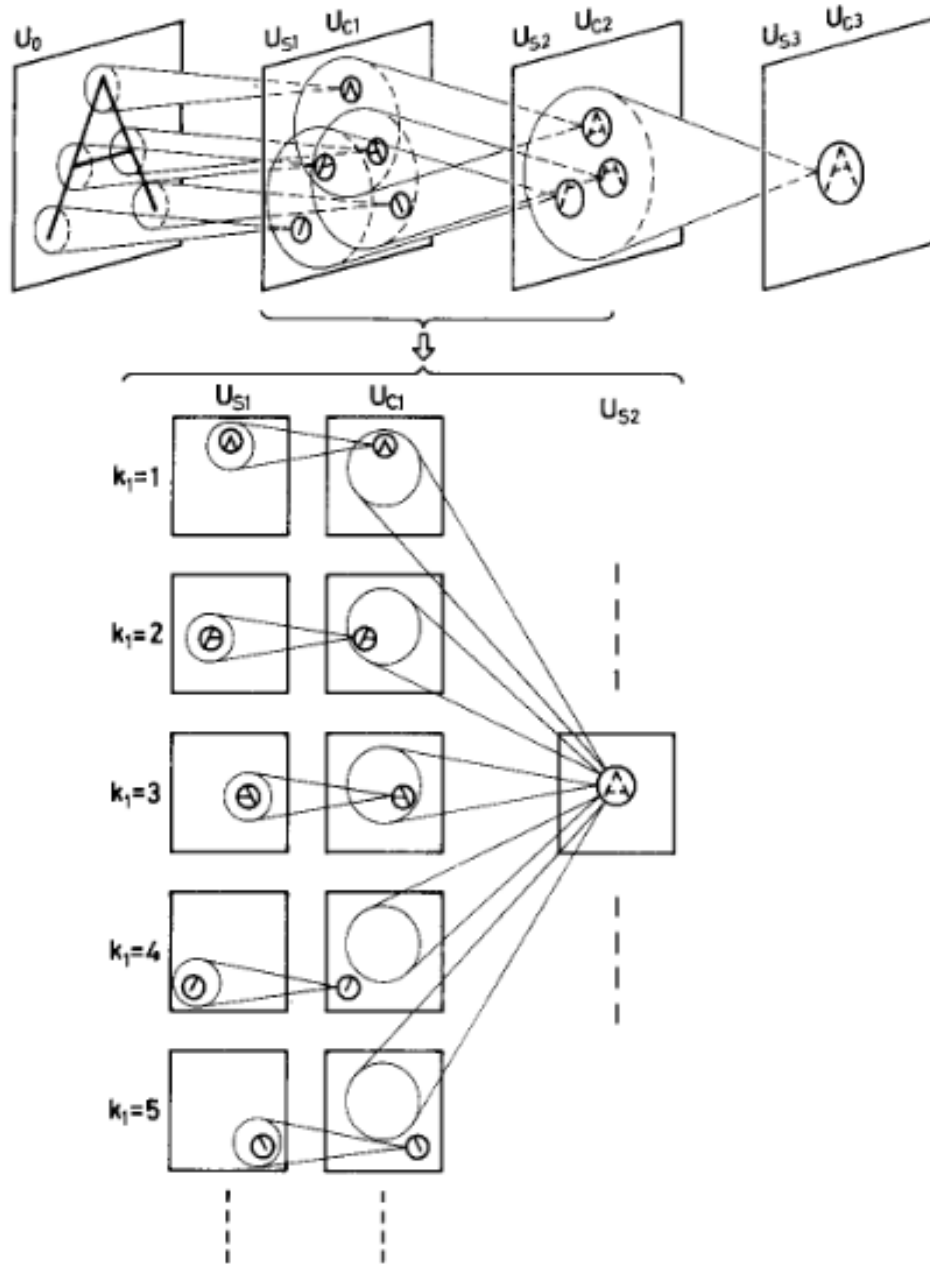


Figure 4: An example of the interconnections between cells and the response of the cells after completion of self-organization

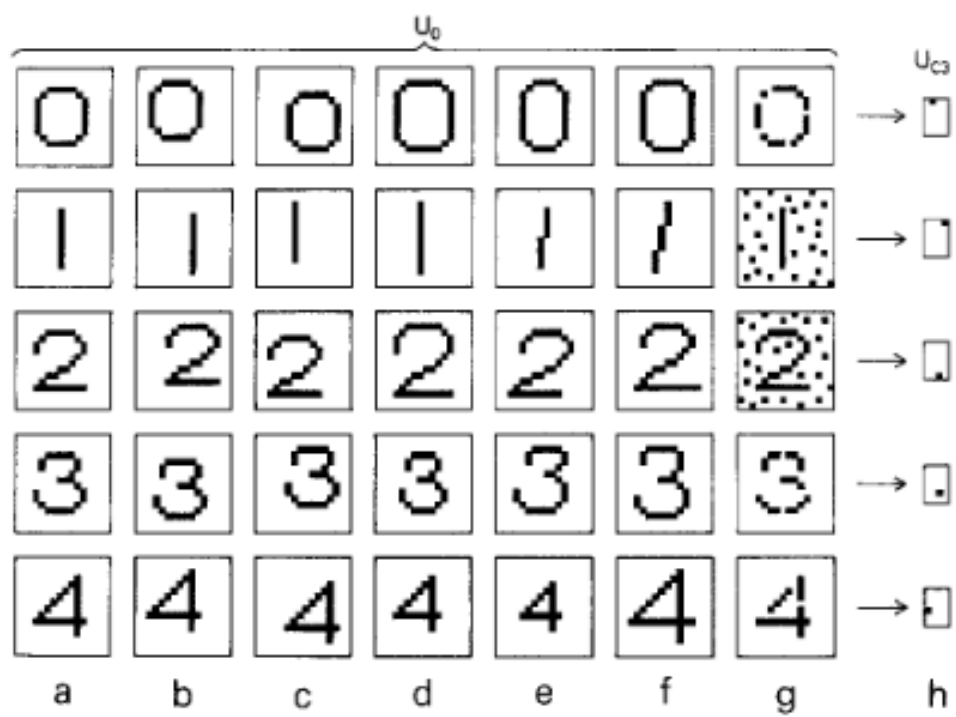


Figure 5: Some examples of distorted stimulus patterns which the neocognitron has correctly recognized, and the response of the final layer of the network



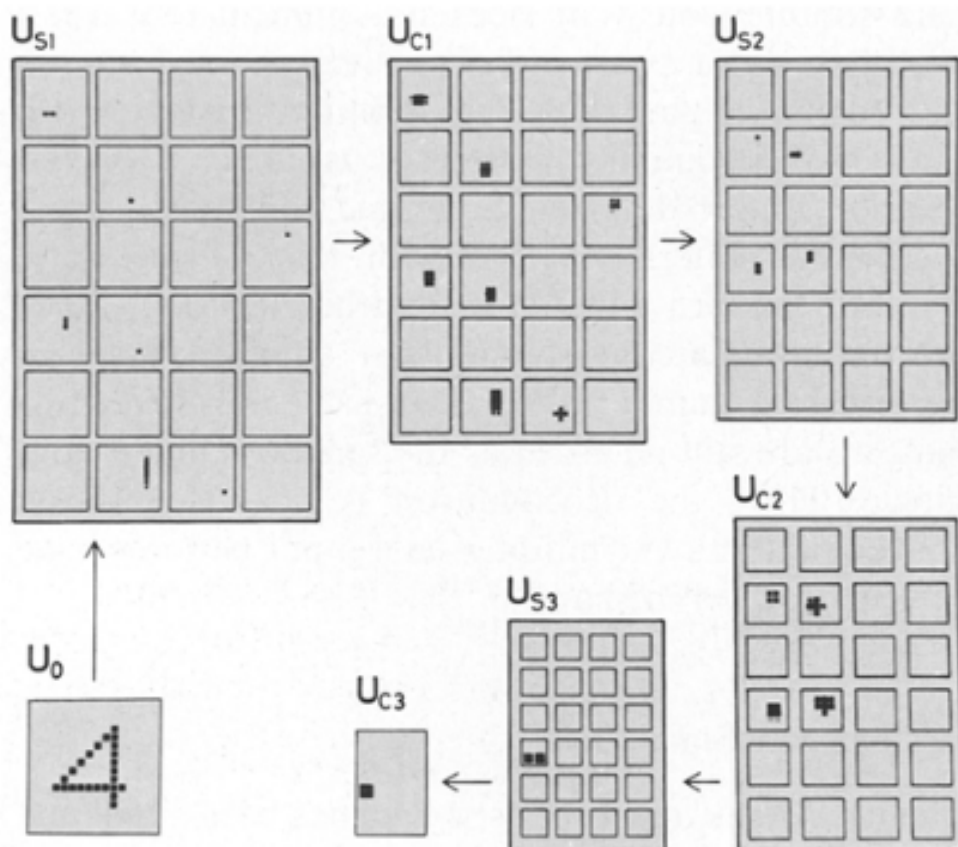


Figure 6: A display of an example of the response of all the individual cells in the neocognitron

### 3 Réseau neuronal convolutif

#### 3.1 Présentation des auteurs

Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton étaient des étudiants de l'université de toronto au moment de la publication de cet article [NIPS2012'4824], J'ai préféré cet article plutôt qu'un des nombreuses publications de M. Yann Lecun car au niveau du contenu cet article est plus simple à comprendre

au niveau de la lecture. Le premier auteur travail maintenant chez Google, les 2 autres continuent à publier régulièrement des article jusqu'à ce jour.

## 3.2 Présentation l'article

Ce n'est pas eux qui inventent le CNN mais il est interessant de noter le bond gigantesque au niveau des performances des CNN sur ImageNet après 2012. Notamment pour la compétition ILSVRC-2012 les auteurs ont réalisé une performances de 15.3% taux d'erreur alors que le second du concours n'arrivait qu'à 26.2%.

L'article est populairement cité dans le domaine, google indique d'ailleurs que le nombre de citation dépasse les 60k. Le dataset utilisé ici est celui d'ImageNet qui est constitué à peu près de 15 millions d'image dans plus de 22 milles catégories. A l'opposé, le MNIST dataset de M.Yann Lecun ne contient que 60k exemples, évidemment il faut prendre en compte que la complexité de l'information n'est pas pareil quand vous comparer des chiffres manuscrites et des images d'objet réel. Le progrès majeur apporté par cet article est qu'il s'agit d'une implémentation GPU de Deep CNN, La structure de leur réseau de neuronne reste quand même traditionnel.

L'avantage majeur du CNN est qu'il ne nécessite pas une connexion total entre les couches, il y a donc moins de paramètre et de poids à ajuster lors de l'apprentissage. Malgré ces qualités, le CNN reste gourmands en terme de mémoire et en performances car le produit de convolution est coûteuse à calculer. D'où l'implémentation sur le plateforme GPU, accompagné d'optimisations accrues pour le produit de convolution 2D.

Le système prends en entrée une image RGB de dimension 256x256, une réadaptation de l'image a été effectué pour avoir des "patch" de dimension valide.

### 3.3 Présentation de la structure

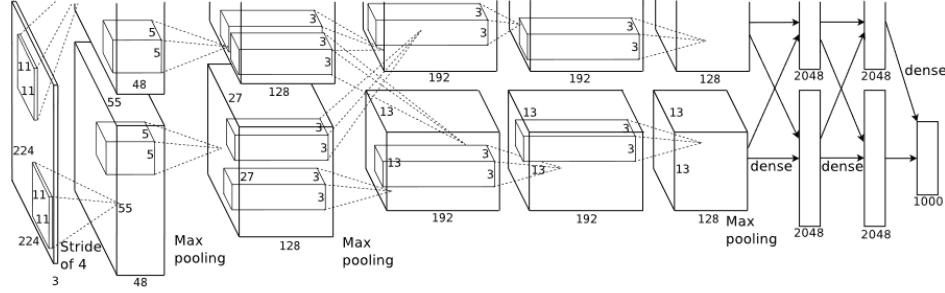


Figure 7: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Comme le montre le schéma ci-dessus, le réseau de neurone est constitué de 5 couches de convolution puis de 3 couches "fully-connected". 2 Cartes graphiques sont nécessaires pour cet implémentation, chaque carte traite ses couches à lui, une communication entre les 2 cartes graphiques n'est possible qu'à une certaine couche précise.

Le cœur du CNN est le produit de convolution, donnant une image d'entrée de dimension  $a * b$ , la première couche de convolution va y passer et produire à son tour plusieurs sous-image qui seront plus petites à cause du filtre. Au fur et à mesure des calculs, plus vous avancez dans la profondeur du réseau, plus les neurones seront petits, à la fin vous obtiendrez une sorte de carte 2d constituée de neurones. Ces derniers auront une réponse spécifique à un stimulus précis.

La manière standard de corriger la sortie d'un neurone est d'utiliser une fonction  $f(x) = \tanh(x)$  ou encore  $f(x) = (1 + e^{-x})^{-1}$ . Alors que nos auteurs ont privilégié la méthode de Rectified Linear Units (ReLU). [Nair Hinton] Mathématiquement, il s'agit de la fonction  $f(x) = \max(0, x)$ . La méthode a effectivement un nom très vendeur mais en réalité

il peut être résumé dans le grande lignes par ce pseudo-code:

```
if input > 0:
    return input
else:
    return 0
```

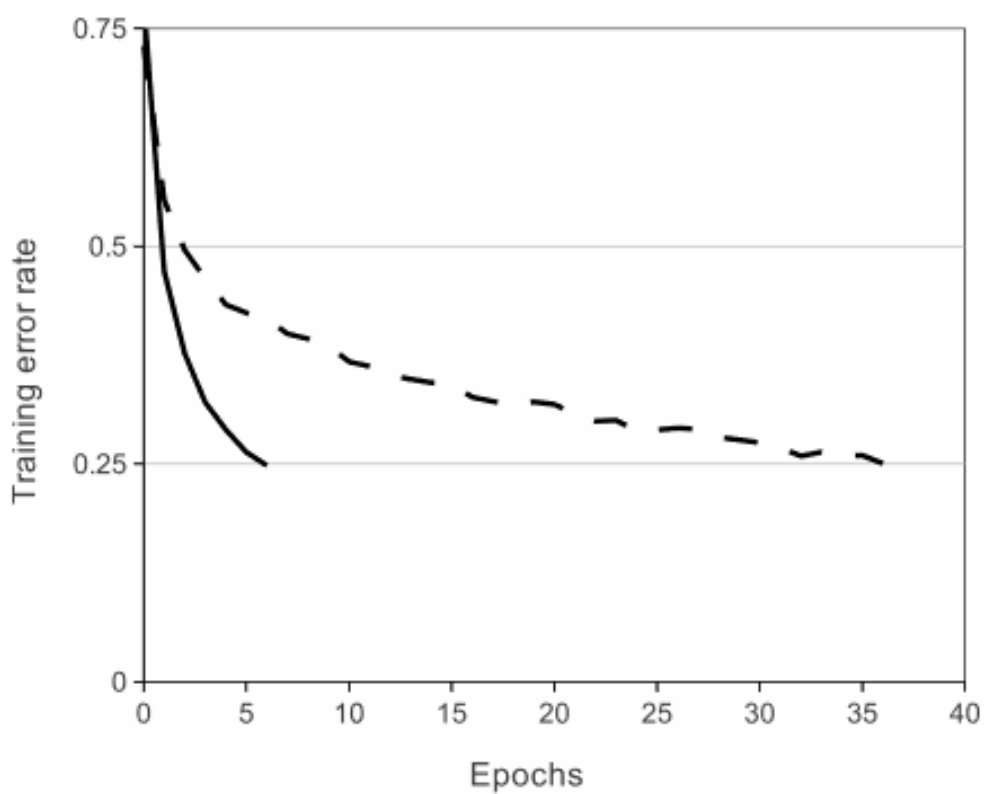


Figure 8: Evolution du taux d'erreur en fonction du cycle sur le dataset CIFAR-10, Trait continue: CNN à 4 couches avec ReLUs, trait discontinue: réseau équivalent avec tanh

Comme vous pouvez le voir sur ce schéma, en utilisant les ReLUs, nous atteignons les 25% taux d'erreur six fois plus vite que la méthode classique.

En dehors de ces 2 couches, les auteurs utilisent aussi une couche de pooling (mise en commun), cela permet de réduire la quantité d'information présente sur l'échantillon et de produire un sous échantillon plus petit, cela permet d'améliorer la puissance de calcul, au détriment de la perte d'une partie d'information.

Il est intéressant de noter que nos auteurs utilisent un filtre de chevauchement "Overlapping Pooling" plutôt qu'un filtre classique de pooling. Dans un processus normal de pooling, l'image est d'abord divisé en grille de dimension  $z \times z$ , à cela nous faisons passer un filtre de taille  $s$  à un pas constant de  $z$ , les filtres se croisent jamais.

Alors que si vous définissez le pas comme  $s/z$ , vous obtenez un parcours de l'image où les filtres se chevauchent en leur bordure. Les auteurs ont observé alors que le risque de surapprentissage est plus basse avec cette méthode.

Dans ce modèle, il y a d'abord l'application de 96 matrices de convolution de dimension  $11 \times 11 \times 3$  avec un pas de 4 pixels.

Puis la seconde couche de convolution utilise 256 matrices de dimension  $5 \times 5 \times 48$ .

La troisième couche nécessite 384 matrices de dimension  $3 \times 3 \times 256$ .

La quatrième couche nécessite 384 matrices de dimension  $3 \times 3 \times 196$ .

Enfin la cinquième couche nécessite 256 matrices de dimension  $3 \times 3 \times 192$ .

Finalement, les 3 dernières couche "Fully-connected" sont constitué de 4096 neurones chacun.

Cette construction de réseau présente près de 60 millions de paramètre, alors que le dataset ILSVRC ne propose que 1000 classes de résultat. Ce qui a poussé les auteurs à chercher une solution face au problème du surapprentissage.

Une première solution est d'augmenter le nombre de donnée présenté via des transformation préservant le label de l'image. Pour cela, les auteurs ont découpé l'image d'entrée de dimension  $256 \times 256$  sous plusieurs sous-image de dimension  $224 \times 224$ , cela permet d'augmenter la taille du vecteur d'apprentissage par 2048. Il est possible d'altérer l'image de départ avec des bruits générés aléatoirement via une gaussienne.

Enfin la deuxième solution proposé est le "dropout", il consiste à mettre 0 à la sortie des neurones de la couche caché avec une probabilité de 0.5. Sans cette technique, le coût sur le temps d'apprentissage aurait doublé car il faudra alors doubler le nombre de cycle pour avoir des résultats qui convergent.

A la fin de tous ces couche de convolution, pooling et de dropout, nous

obtenons une carte neuronale sophistiquée où l'interprétation de ces résultats nécessite l'emploi des couches entièrement connectées (fully-connected), ces neurones particuliers ont une connexion totale vers toutes les sorties de la couche précédente, c'est à ce niveau que les résultats sont traduits pour être humainement compréhensibles, dans notre exemple, le résultat est la détermination de la classe qu'appartient l'objet présente dans l'image.



Figure 9: 96 convolutional kernels of size 11x11 learned by the first convolutional layer on the 224x224 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU2

Comme vous pouvez le voir ci-dessus, le raisonnement machine est assez différent du raisonnement humain, dès la première couche convolution, les matrices résultantes n'ont plus de propriété spatiale ni sémantique.

Voici un exemple de résultat sur

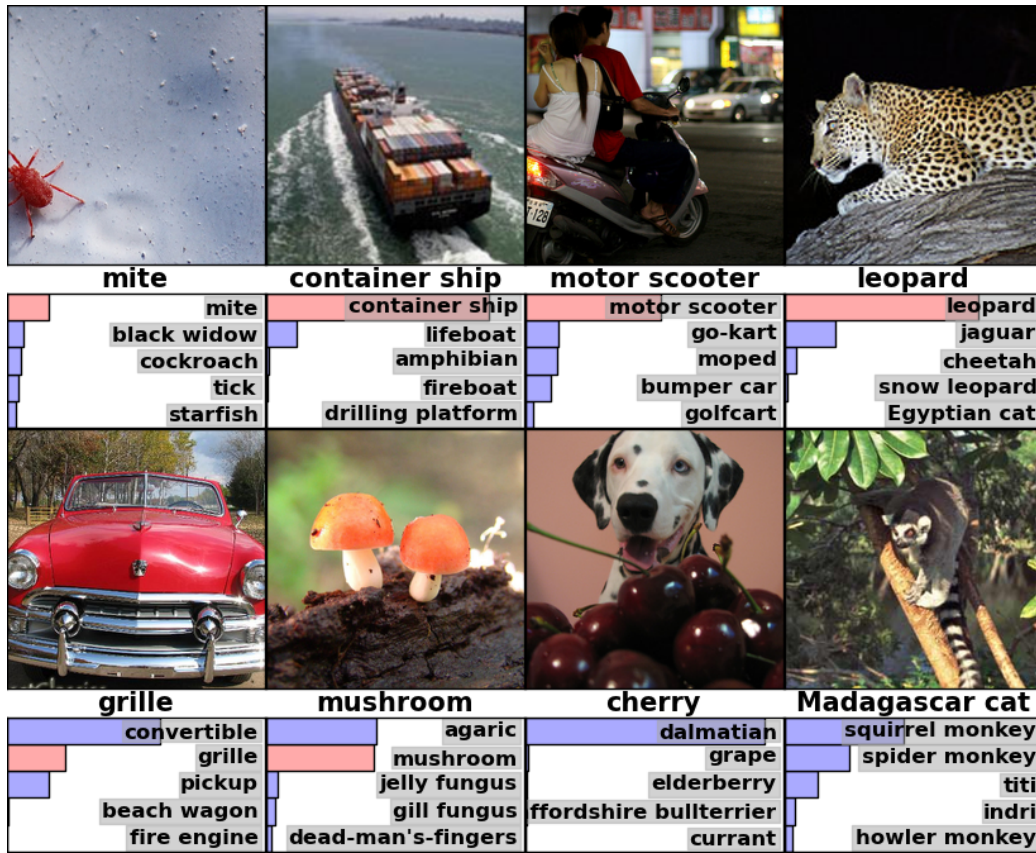


Figure 10: Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5)

## 4 Conclusion