Réseaux de neurones convolutionnels!!

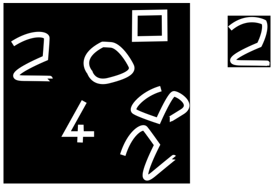
Agissent comme des détecteurs de motifs!

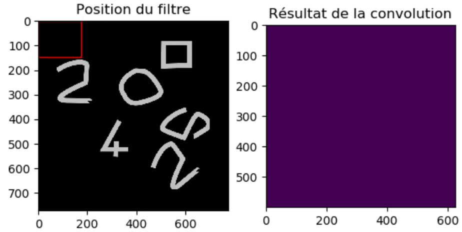
Réseaux de neurones convolutionnels!!

Agissent comme des détecteurs de motifs!

Voyons cela sur une forme simple...

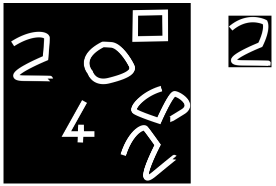
À quoi ressemblera la convolution de cette image avec ce patch?

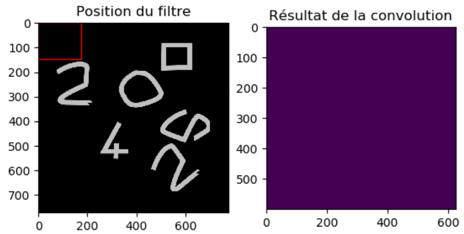





```
np.unravel_index(np.argmax(result),result.shape)
Out[10]: (174, 68) # position du chiffre 2 sur l'image.
```

À quoi ressemblera la convolution de cette image avec ce patch?

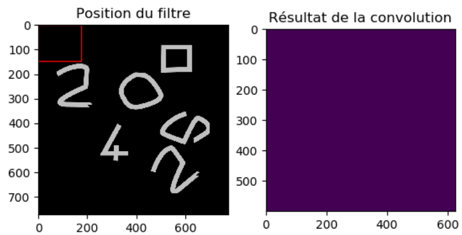





```
np.unravel_index(np.argmax(result),result.shape)
Out[10]: (174, 68) # position du chiffre 2 sur l'image.
```

- La convolution détecte le motif quelque soit la translation

À quoi ressemblera la convolution de cette image avec ce patch?

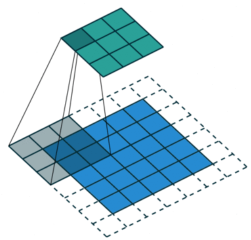



```
np.unravel_index(np.argmax(result),result.shape)
Out[10]: (174, 68) # position du chiffre 2 sur l'image.
```

- La convolution détecte le motif quelque soit la translation
- Évidemment, en pratique, les filtres sont appris, et sont beaucoup plus abstraits.

Transposed convolution

En deep learning, le pas et le pooling réduisent la taille de l'image.



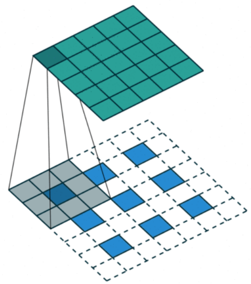
Parfois, nous voulons obtenir l'effet inverse,

- Segmentation d'image
- Visualisation et interpetation

mais comment?

Transposed convolution (parfois abusivement appelé Déconvolution)

Insertion de zéros et convolution classique:



Justification en 1D:

Soient

- $x = [x_1 \dots x_7]$ un signal 1D,
- $w = [w_1, w_2, w_3]$ un filtre,
- $y = [y_1, y_2, y_3]$ la convolution $(x \star w)$ avec un pas de 2.

Cette opération peut s'écrire sous la forme:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_1 & w_2 & w_3 & 0 & \dots & & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & & \dots \\ 0 & \dots & & & w_1 & w_2 & w_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{pmatrix}$$

Ou de manière plus compacte: $y = Wx$

Justification en 1D:

Étant donné que la convolution s'écrit

$$y = Wx$$

Cherchons à déconvoluer en calculant:

$$W^{-1}y$$

Justification en 1D:

Étant donné que la convolution s'écrit

$$y = Wx$$

Cherchons à déconvoluer en calculant:

$$W^{-1}y$$

Faisons l'hypothèse que W est orthonormale, nous avons alors:

$$W^{-1} = W^T$$

Justification en 1D:

Étant donné que la convolution s'écrit

$$y = Wx$$

Cherchons à déconvoluer en calculant:

$$W^{-1}y$$

Faisons l'hypothèse que W est orthonormale, nous avons alors:

$$W^{-1} = W^T$$

Un exemple pour lequel cette hypthèse est vraie:

$$W = [-1, 0, 0, 1]$$

Dans la vraie vie, c'est rare.

Justification en 1D:

Sous condition d'othonormalité, le signal déconvolué $\hat{x}$ s'écrit:

$$\hat{x} = W^T y$$

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{pmatrix} = \begin{pmatrix} w_1 & 0 & 0 \\ w_2 & 0 & \vdots \\ w_3 & w_1 & \\ \vdots & w_2 & 0 \\ & w_3 & w_1 \\ & 0 & w_2 \\ 0 & 0 & w_3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Justification en 1D:

Réarrangement des termes: La déconvolution est une convolution!

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{pmatrix} = \begin{pmatrix} w_3 & w_2 & w_1 & 0 & \dots & 0 \\ 0 & w_3 & w_2 & w_1 & & \vdots \\ \vdots & 0 & w_3 & w_2 & w_1 & \\ & & \dots & & & \\ 0 & & & w_3 & w_2 & w_1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ y_1 \\ 0 \\ 0 \\ y_2 \\ 0 \\ 0 \\ y_3 \\ 0 \\ 0 \end{pmatrix}$$

À retenir de cette dernière partie

- L'augmentation de la résolution peut être réalisée par une convolution en insérant des zéros
- Sous certaines hypothèses fortes, cela correspond à une déconvolution
- En pratique, ce n'est pas le cas:

=> Permet d'obtenir une visualisation (voir prochain cours sur l'intepprétation des réseaux)

=> Effet négligeable quand les filtres sont appris: Segmentation d'image.

Source: Lecture 7 du cours de deep learning CS320

Glossaire

- Padding : ajouter des lignes et des colonnes sur l'extérieur de l'image
- scaling : réduire / augmenter la taille de l'image
- shearing : scaling vertical et horyzontal de différentes amplitudes
- cropping : extraire une partie de l'image.
- stride : pas de convolution

Lab : Manipulation basiques et convolutions

Ouvrir le fichier: TP*convolutions.ipynb*