Aprendizado Profundo 1

Introdução a Redes Neurais Convolucionais

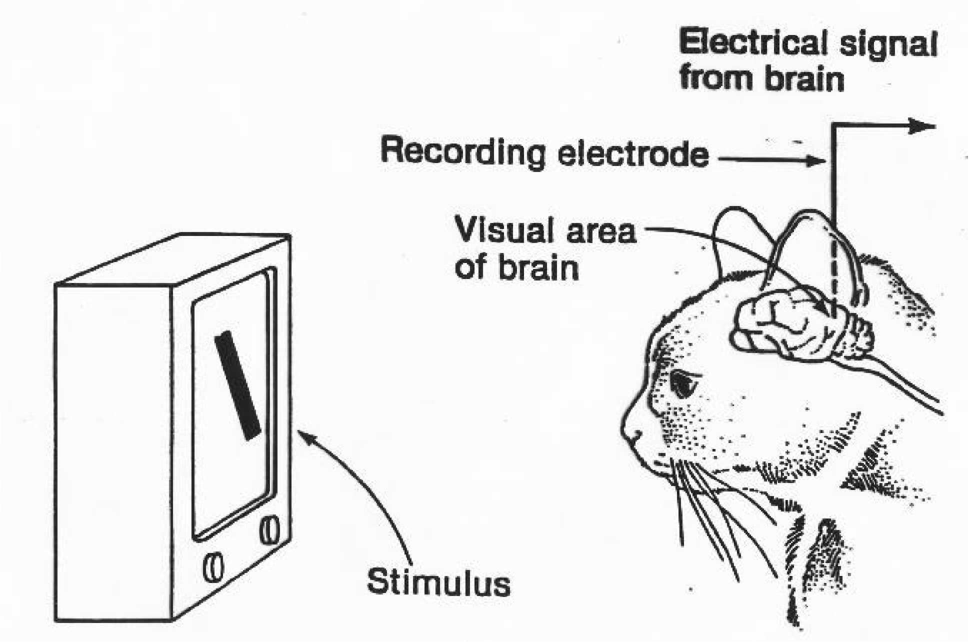
Professor: Lucas Silveira Kupssinskü

Agenda

- Motivação para CNN
- Invariância e Equivariância
- Convolução
  - *Size*
  - *Stride*
  - *Dilation*
  - *Padding*
- Campos Receptivos *(Receptive Fields)*
- *Downsampling* e *Upsampling*
- Implementação de Convolução

Motivação Histórica

- David Hubel e Torsten Wiesel realizaram experimentações implantando eletrodos no córtex visual de gatos
- Aparentemente o processamento começa identificando padrões simples como arestas
  - Vídeo

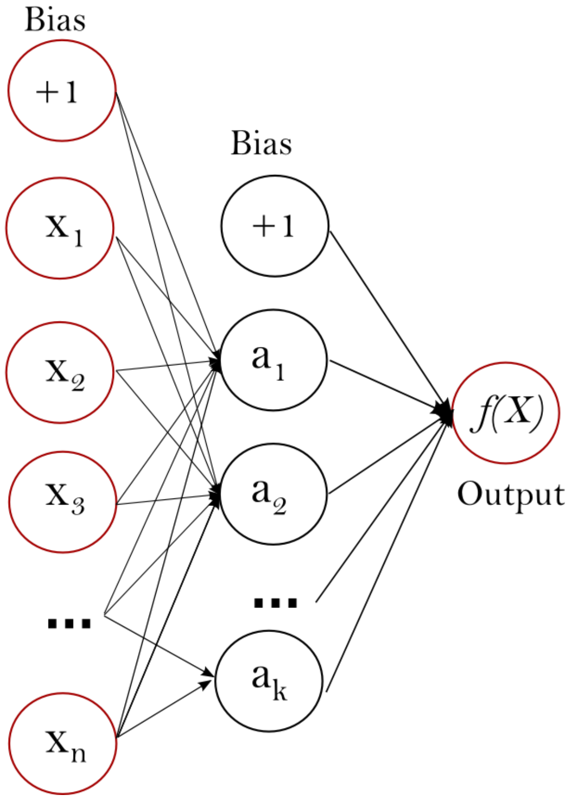


Redes Convolucionais

- São redes que incluem operações de convolução
  - Usadas principalmente em imagens
- Criadas em 1989(!) por Yann LeCun (LeNet - 5)
  - LECUN, Yann et al. Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, v. 2, 1989.
- Explodiram em popularidade pós 2012
  - AlexNet
    - KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, v. 25, 2012.
    - Treinamento em GPUs
    - +Dados
    - +Poder Computacional

Problemas com MLP

- Imagens tem três propriedades que sugerem uso de arquiteturas específicas:

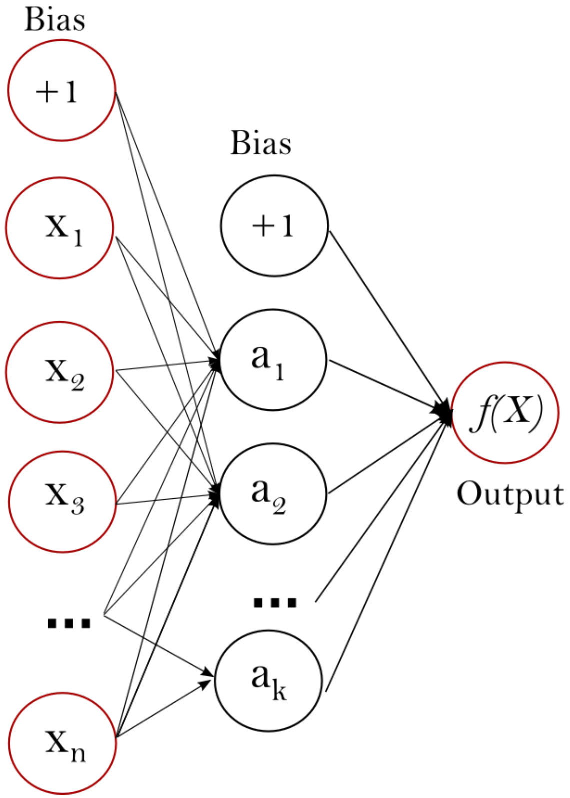


Problemas com MLP

- Imagens tem três propriedades que sugerem uso de arquiteturas específicas:
    - Alta dimensionalidade;
        - 512x512 = 262144
        - Se tivermos 100 unidades na primeira camada oculta teremos: 26214500 pesos
            - Se considerarmos float 32, ~800mb de pesos só para a primeira camada ☹

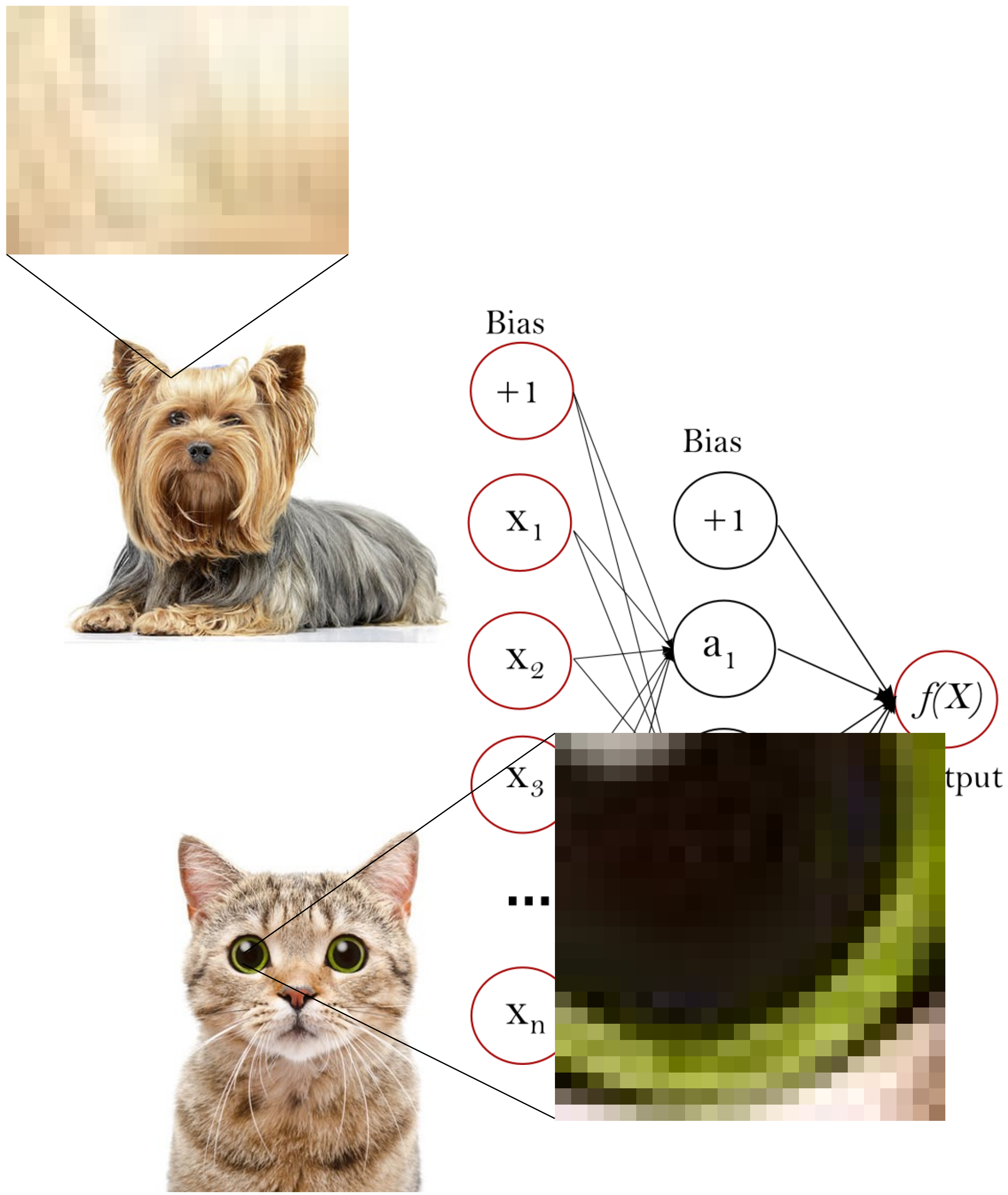
512x512

512x512



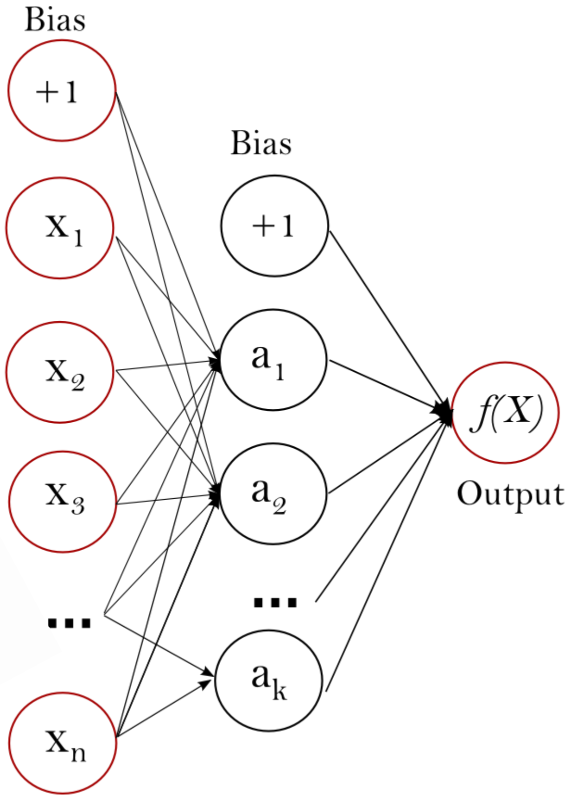
Problemas com MLP

- Imagens tem três propriedades que sugerem uso de arquiteturas específicas:
  - Alta dimensionalidade;
  - Alta coesão espacial;
    - Pixels próximos tem valores altamente correlacionados
    - Uma “característica” pode ter o mesmo significado em vários locais diferentes na imagem (orelha do cachorro ou gato)



Problemas com MLP

- Imagens tem três propriedades que sugerem uso de arquiteturas específicas:
  - Alta dimensionalidade;
  - Alta coesão espacial;
  - Semântica estável frente a transformações geométricas;



Problemas com MLP

- Imagens tem três propriedades que sugerem uso de arquiteturas específicas:

Camadas convolucionais trazem diversos benefícios:

- processam regiões da imagem separadamente compartilhando pesos
- usam menos parâmetros do que camadas totalmente conectadas
- exploram a correlação entre pixels próximos
- não precisam reaprender padrões em locais diferentes da imagem

Invariância e Equivariância

- Invariância
  - Uma função f de uma imagem x é dita invariante a uma transformação t se:

$$f\big[t[x]\big] = f[x]$$

  - ou seja, independente da transformação temos o mesmo resultado

Invariância e Equivariância

- Equivariância
  - Uma função f de uma imagem x é dita equivariante a uma transformação t se:

$$f\big[t[x]\big] = t\big[f[x]\big]$$

  - ou seja, independente da transformação temos o mesmo resultado

Invariância e Equivariância

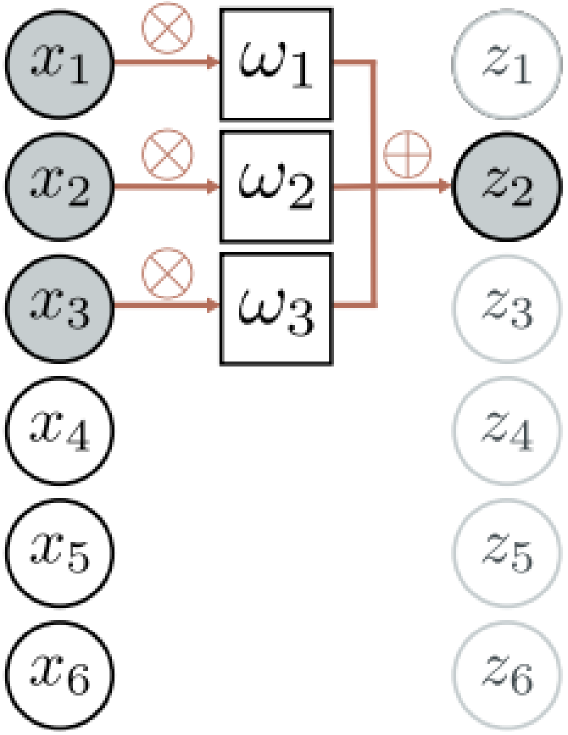
- Repare na segmentação ao lado
  - Transladar a imagem deve resultar na mesma segmentação transladada

Convolução

- Vamos iniciar com convolução 1D
- A convolução transforma um vetor de entradas x em um vetor de saídas z, tal que:

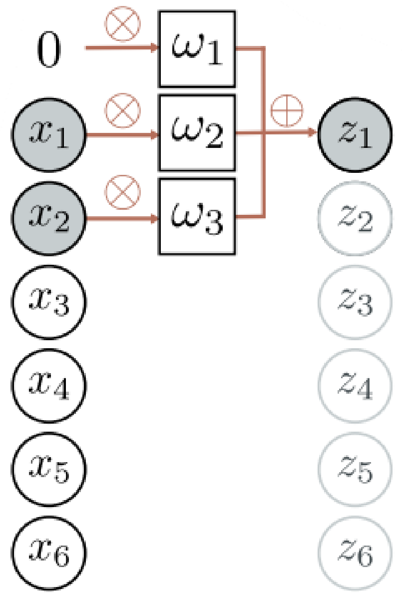
$$z_i = w_1 x_{i-1} + w_2 x_i + w_3 x_{i+1}$$

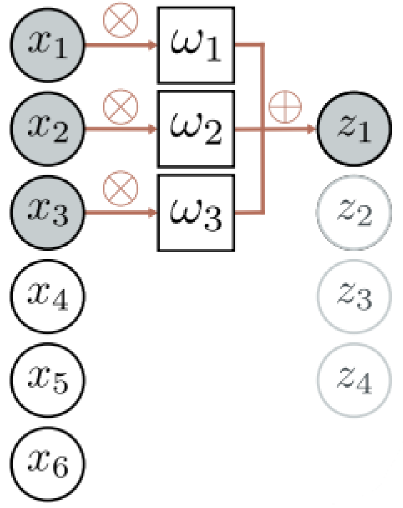
- Repare que os pesos são **compartilhados** por toda a entrada
- A esse conjunto de pesos damos o nome de *kernel* ou *filtro*



Padding

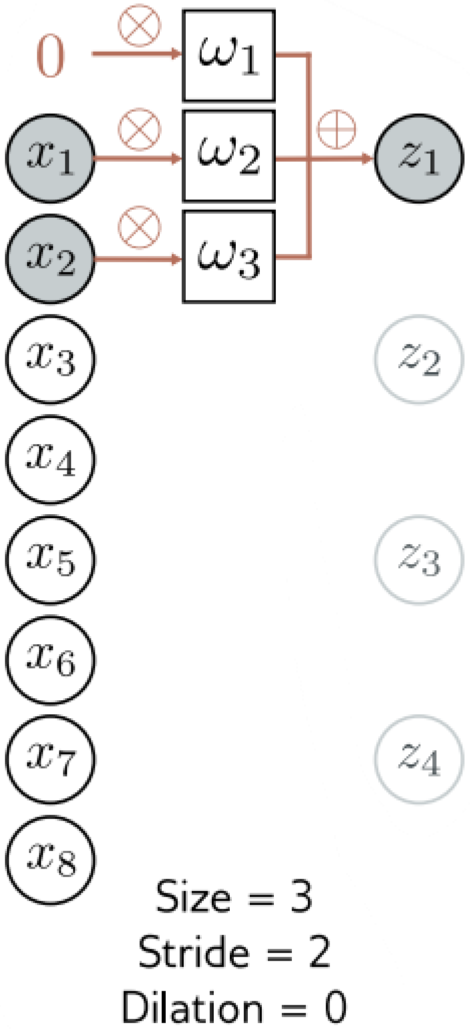
- Como lidar com os estremos da entrada?
  - *Zero Padding*
    - Completar a entrada com zeros
  - Desconsiderar os extremos (Padding=Valid)
    - Aplicar convolução apenas onde o *kernel* cabe
  - Considerar que a entrada é “circular” ou periódica
    - Pouco utilizado





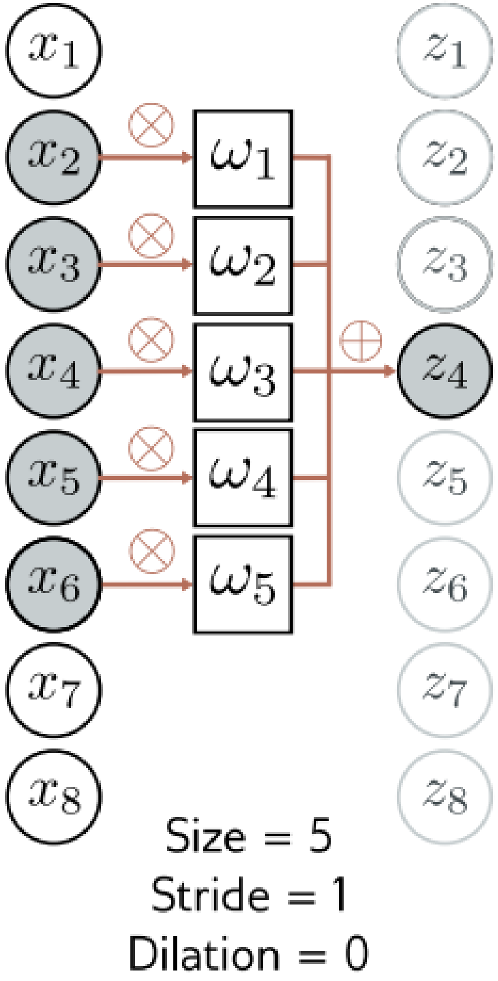
Stride, Kernel Size e *Dilation*

- *Stride*
  - É o tamanho da transladação aplicada no *kernel*
  - Pode ser interpretado como uma forma de *downsampling* quando o *stride* é maior que 1



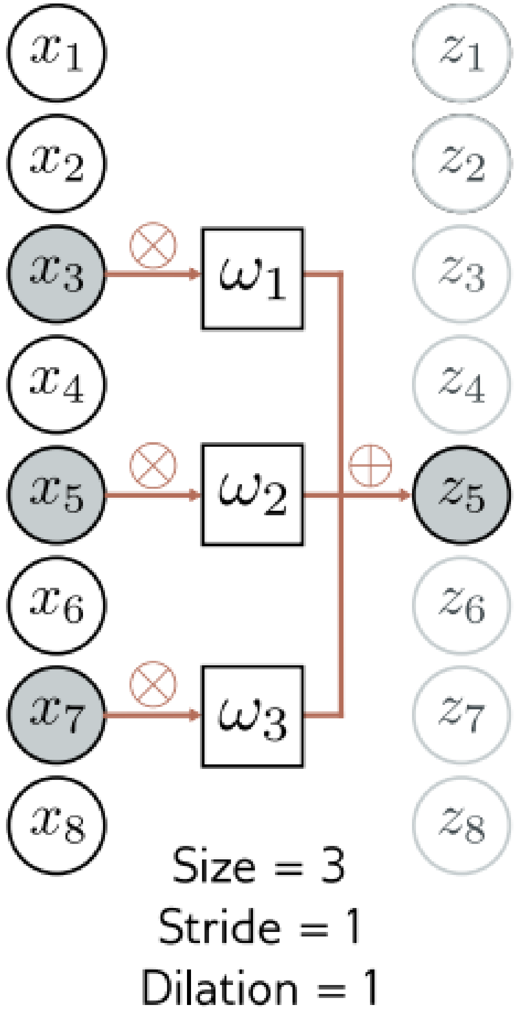
Stride, Kernel Size e *Dilation*

- *Kernel Size*
  - Tamanho do filtro/*kernel* que será usado
  - Permite que sejam integradas informações espaciais mais distantes
    - Mas aumenta o número de pesos que precisamos aprender



Stride, Kernel Size e *Dilation*

- *Dilation*
  - Permite que sejam integradas informações espaciais mais distantes sem aumento de parâmetros
    - Equivalente a aumentar o kernel mas deixar alguns pesos zerados



Camada Convolucional

- Uma camada convolucional computa a convolução, adiciona o bias e aplica uma função de ativação

$$h_i = \varphi[\beta + w_1 x_{i-1} + w_2 x_i + w_3 x_{i+1}] = \varphi\left[\beta + \sum_{j=1}^{3} w_j x_{i+j-2}\right]$$

β, w_1, w_2 e w_3 são os parâmetros aprendidos

Camada Convolucional

- Repare que a camada convolucional é um caso especial de camada totalmente conectada

$$h_i = \varphi\left[\beta + \sum_{j=1}^{3} w_j x_{i+j-2}\right]$$

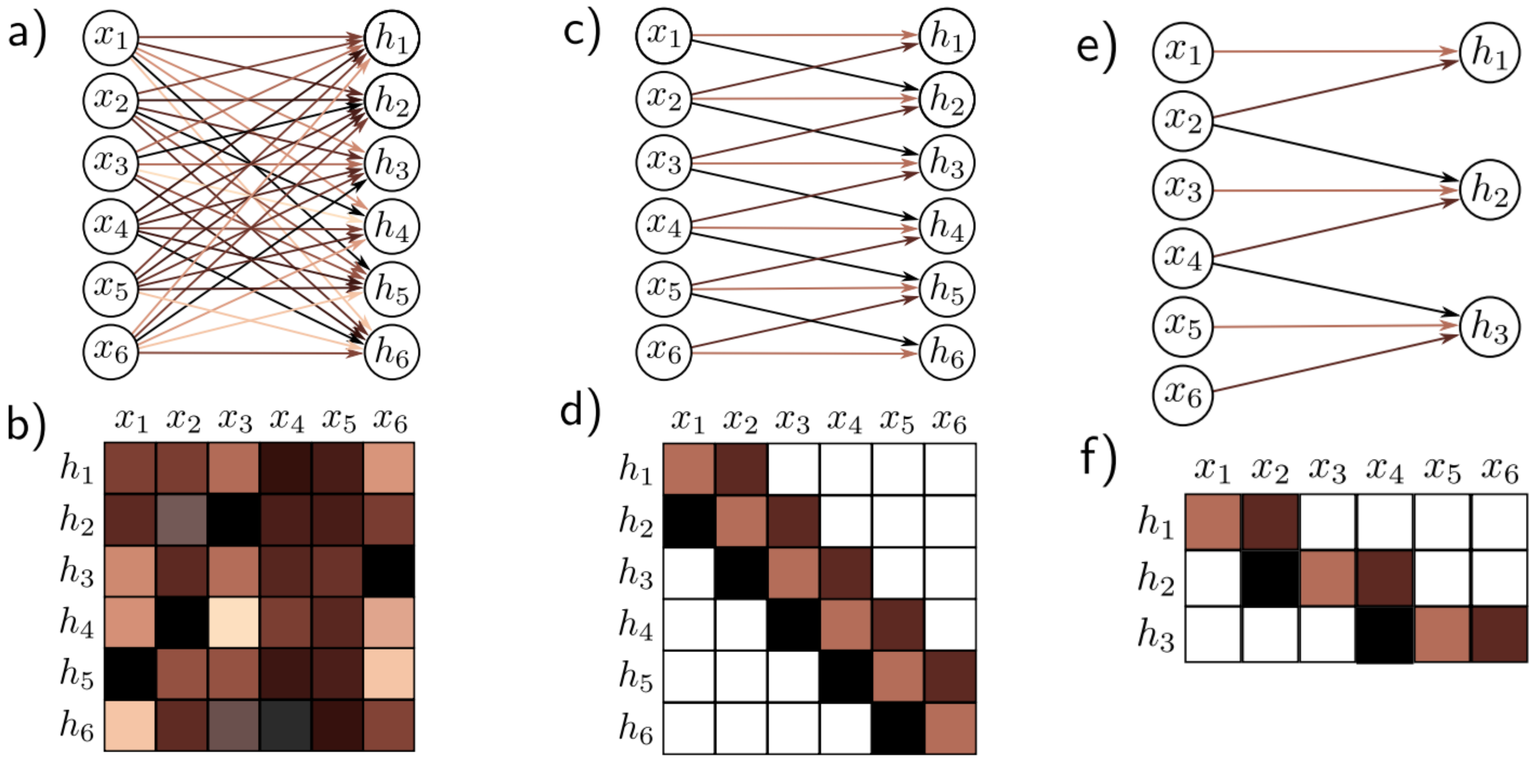
Convolucional

$$h_i = \varphi\left[\beta_i + \sum_{j=1}^{D} w_{ij} x_j\right]$$

Totalmente Conectada

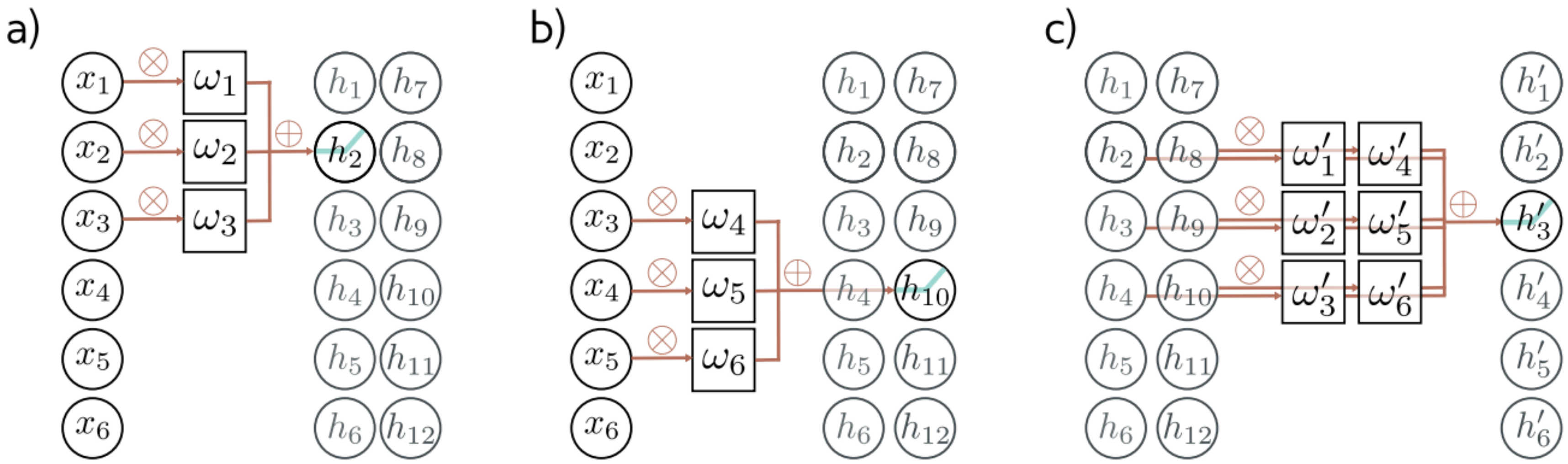
Camada Convolucional

- Existe uma forma inteligente de organizar a matriz de pesos para obter o resultado da convolução sem precisar iterar pela imagem



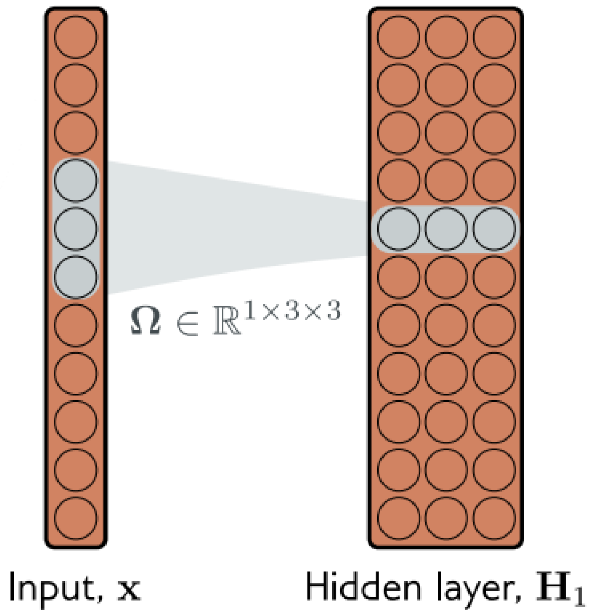
Camada Convolucional

- Tipicamente temos diversos canais de entradas e saída quando estamos aplicando uma convolução
  - As ativações resultantes de um filtro são armazenadas em unidades ocultas tipicamente denominadas de *feature maps* (ou canal)
  - a) e b) aplicação de duas convoluções diferentes geram feature maps diferentes
  - c) a convolução 1D da camada posterior atua em todos os feature maps



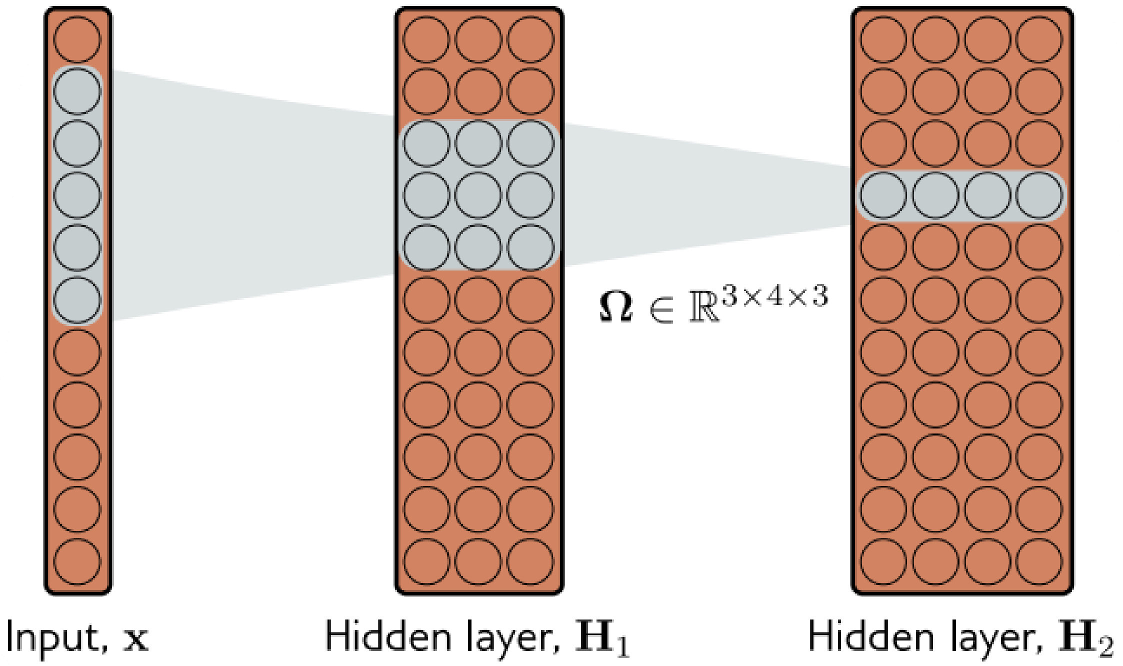
Campo Receptivo (*Receptive Field)*

- Repare que, ao realizar uma convolução uma unidade oculta (do *feature map*) foi gerada levando em conta apenas algumas das entradas
  - Damos o nome de Campo Receptivo a região considerada na unidade oculta



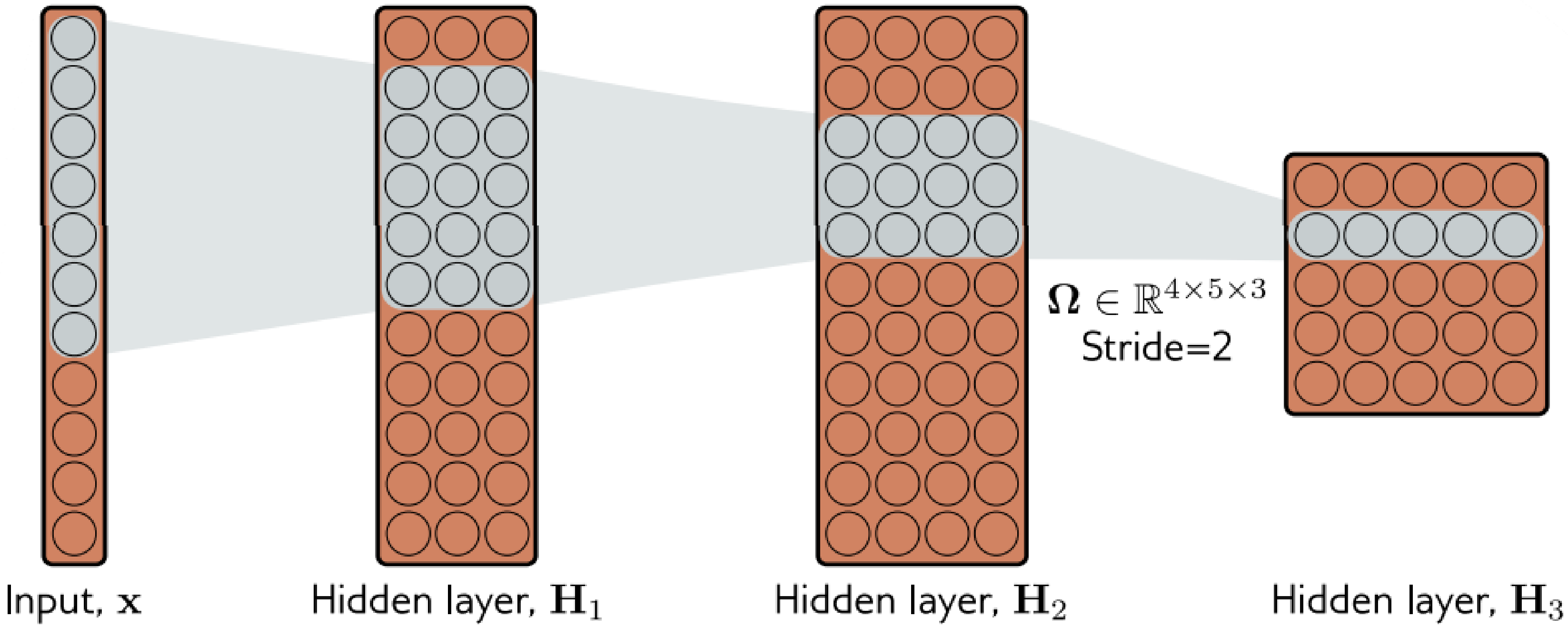
Campo Receptivo (*Receptive Field)*

- Repare que, ao realizar uma convolução uma unidade oculta (do *feature map*) foi gerada levando em conta apenas algumas das entradas
  - Damos o nome de Campo Receptivo a região considerada na unidade oculta



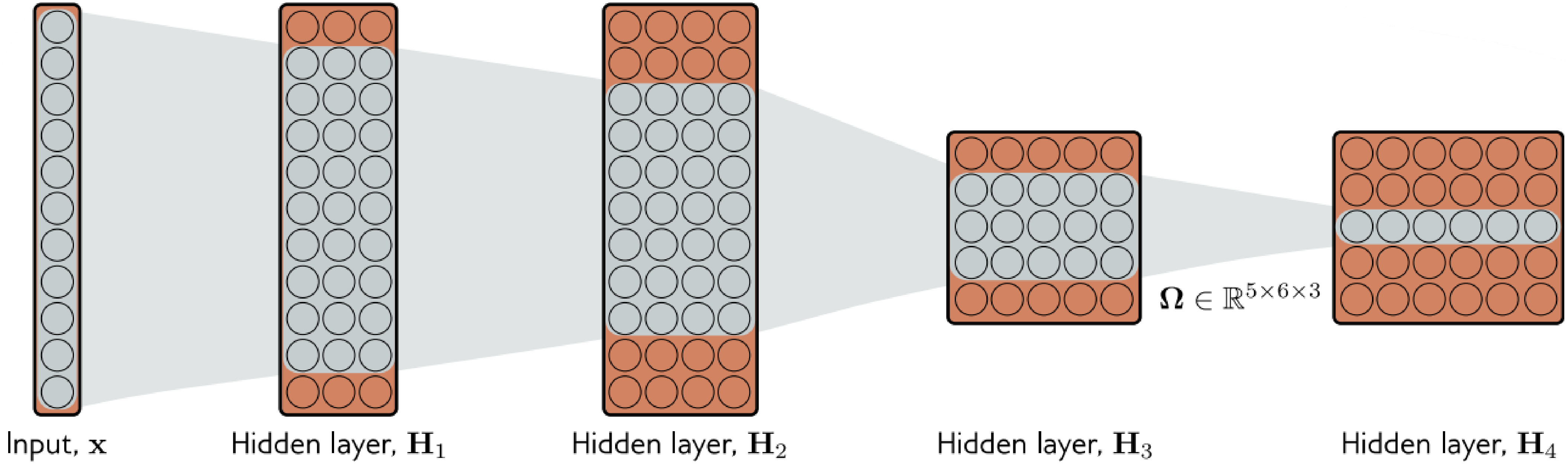
Campo Receptivo (*Receptive Field)*

- Repare que, ao realizar uma convolução uma unidade oculta (do *feature map*) foi gerada levando em conta apenas algumas das entradas
  - Damos o nome de Campo Receptivo a região considerada na unidade oculta



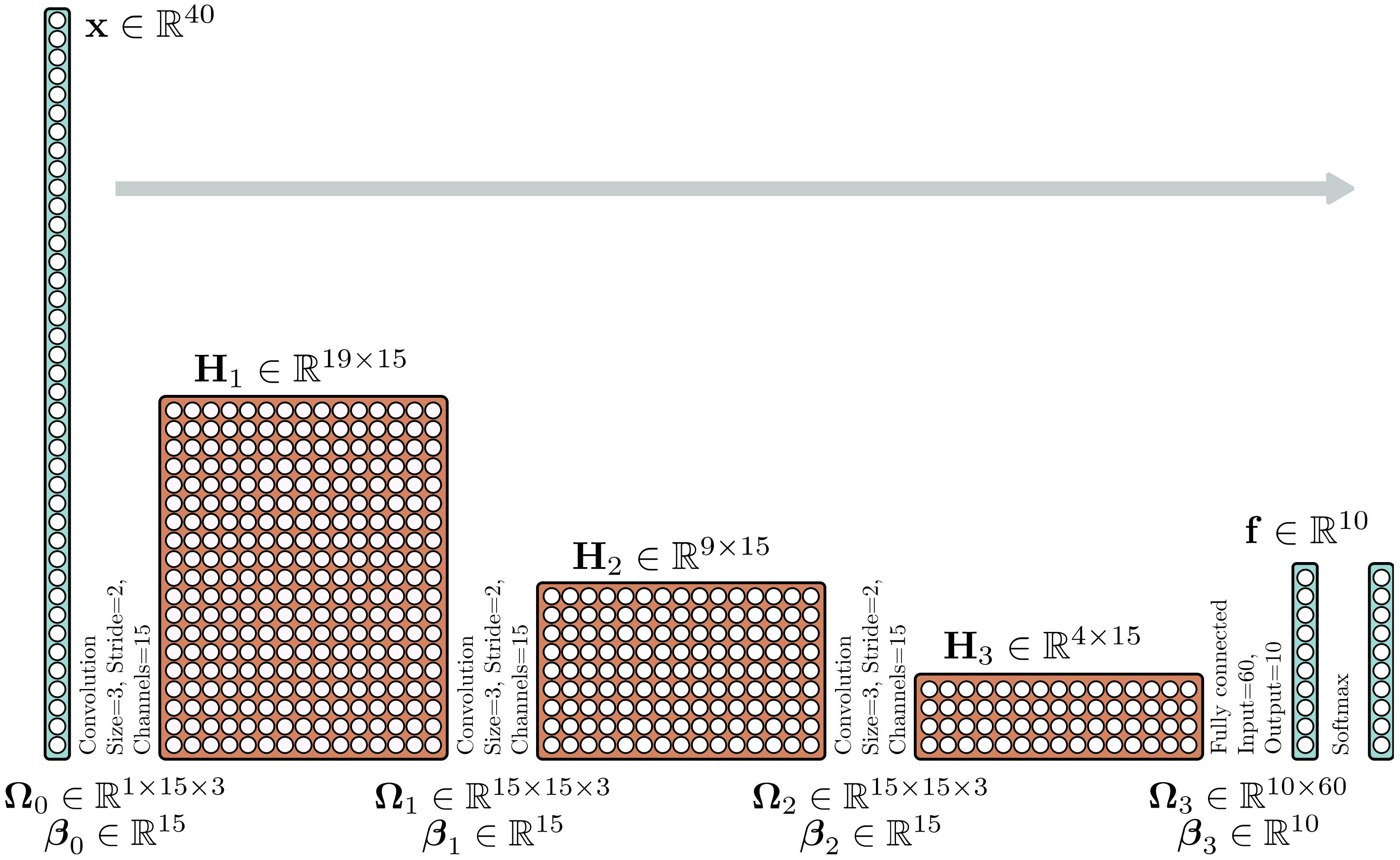
Campo Receptivo (*Receptive Field)*

- Repare que, ao realizar uma convolução uma unidade oculta (do *feature map*) foi gerada levando em conta apenas algumas das entradas
  - Damos o nome de Campo Receptivo a região considerada na unidade oculta



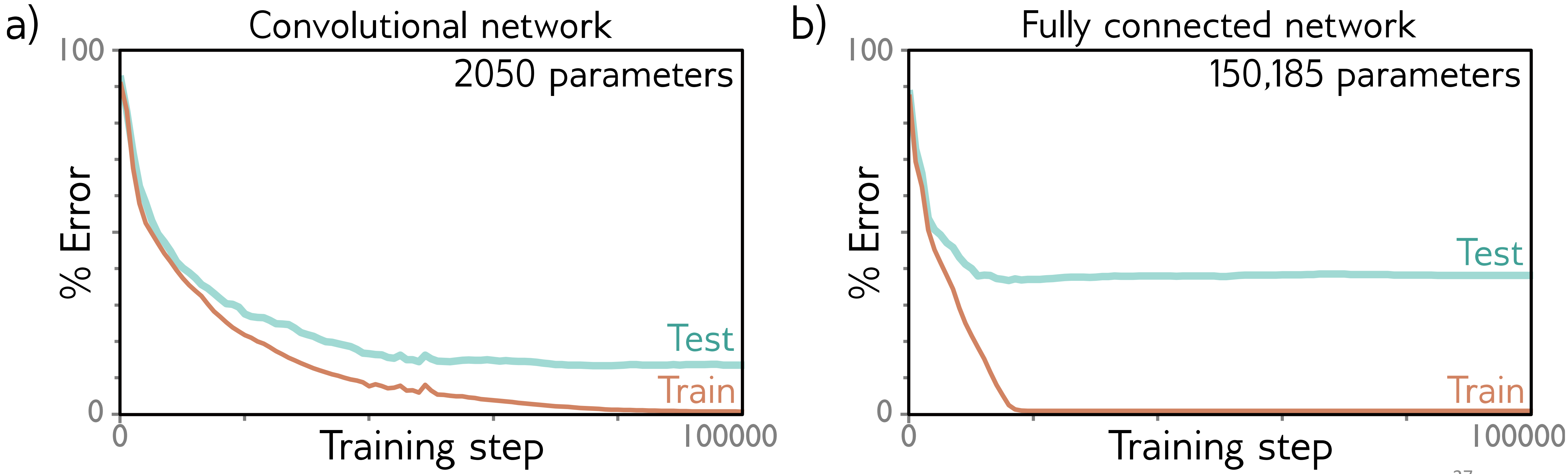
Uma exemplo de CNN

- Quatro camadas ocultas
  - 3 *Conv*
  - 1 *Fully Connected*
  - 2050 parâmetros
- 100000 iterações
- SGD
- LR=0.01
- Batch Size=100



Uma exemplo de CNN

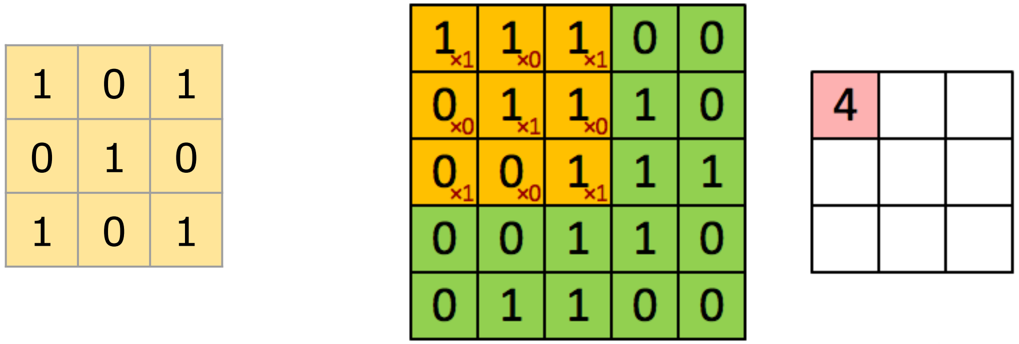
- Uma CNN atinge resultados melhores que um MLP no MNIST 1D mesmo tendo duas ordens de magnitude menos parâmetros treináveis.
- Por quê?



CNN para 2D

- Estender o conceito de CNN para 2D é natural

$$h_i = \varphi\left[\beta + \sum_{m=1}^{3}\sum_{n=1}^{3} w_{m,n} x_{i+m-2,j+n-2}\right]$$

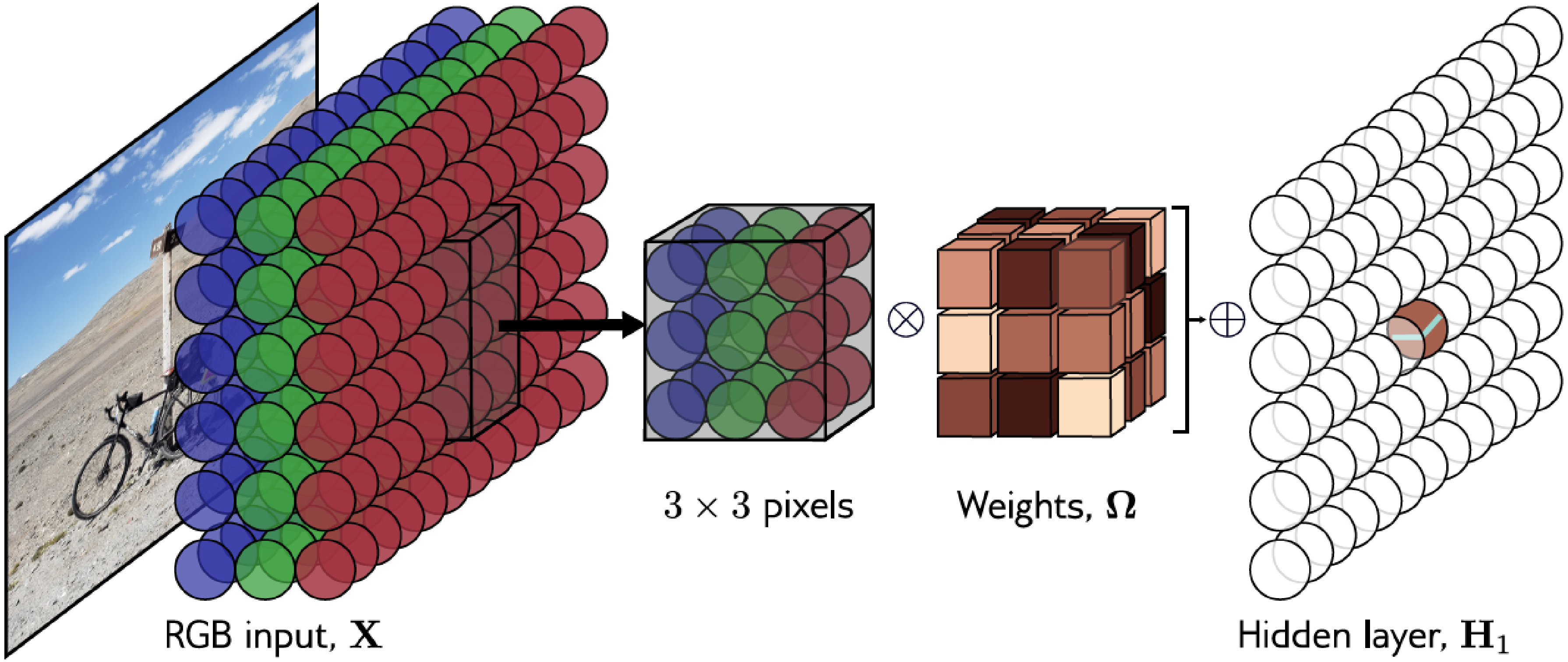


Image

Convolved Feature

CNN para 2D

- Estender o conceito de CNN para 2D é natural



Parâmetros Treináveis

- Considerando C_i o número de canais de entrada e C_o o número de canais de saída temos

$$w \in \mathbb{R}^{C_i \times C_o \times K \times K}$$

$$\beta \in \mathbb{R}^{C_o}$$

Parâmetros Treináveis

- Considerando C_i o número de canais de entrada e C_o o número de canais de saída temos

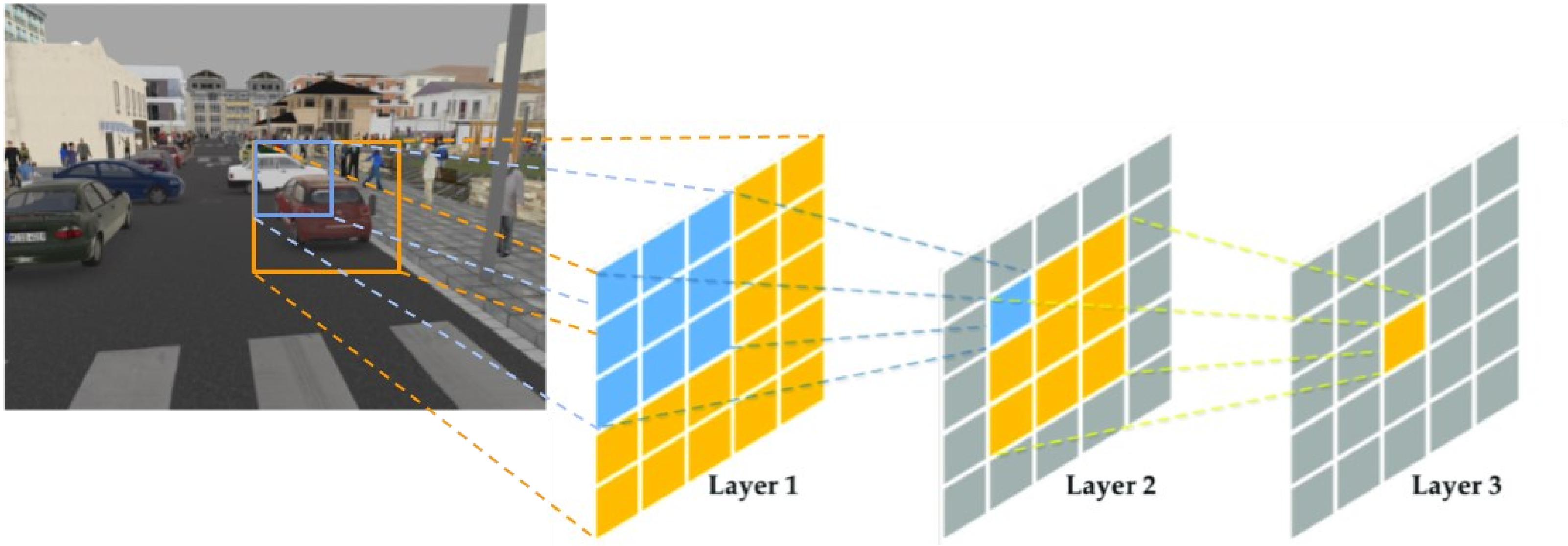
$$w \in \mathbb{R}^{C_i \times C_o \times K \times K}$$

$$\beta \in \mathbb{R}^{C_o}$$

Por exemplo:
Para 3 canais de entrada, 5 de saída e conv 3x3
Temos 3*5*3*3 = 185 parâmetros nos *kernels* + 5 bias

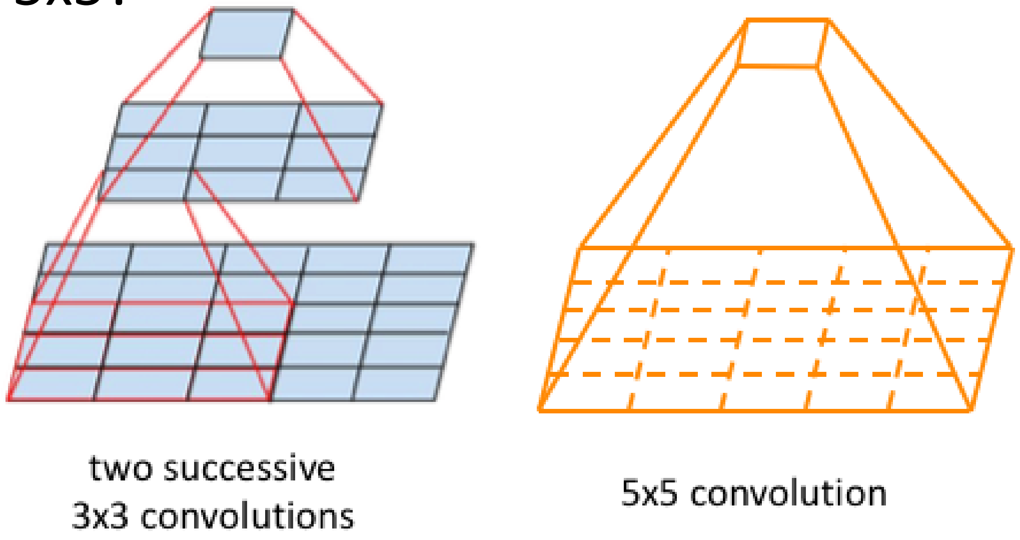
Campos Receptivos (2D)

- Mesma ideia anterior
  - Exemplo da figura abaixo usar filtros 3x3



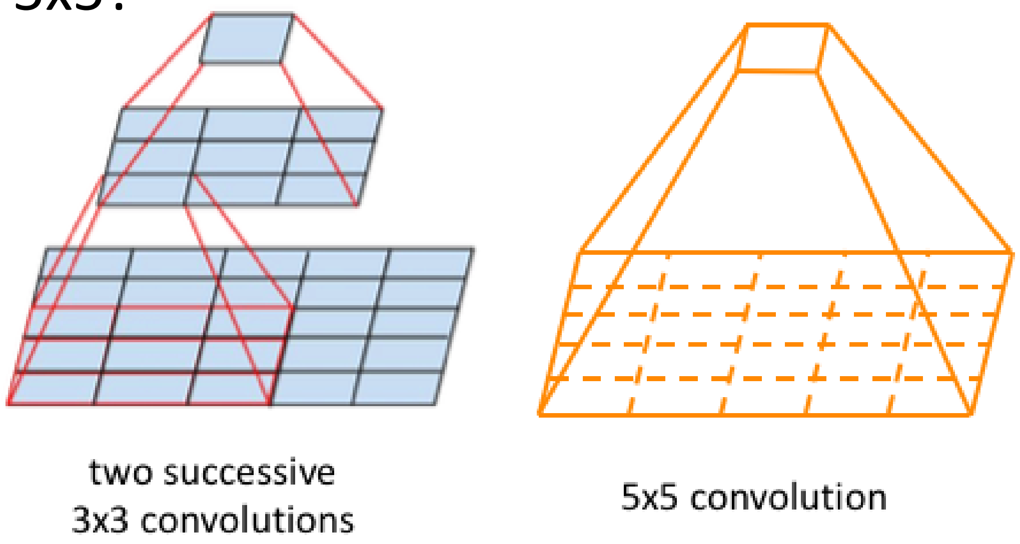
Campos Receptivos (2D)

- Repare que o campo receptivo de duas convs 3x3 é igual ao de uma conv 5x5
- Um campo receptivo pequeno pode perder informações relevantes
- Um campo receptivo muito grande (eventualmente) se torna igual a uma camada densa (totalmente conectada)
- É melhor aplicar duas 3x3 sucessivas ou uma 5x5?



Campos Receptivos (2D)

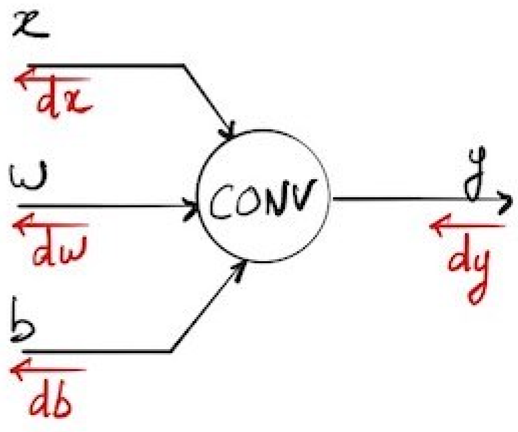
- Repare que o campo receptivo de duas convs 3x3 é igual ao de uma conv 5x5
- Um campo receptivo pequeno pode perder informações relevantes
- Um campo receptivo muito grande (eventualmente) se torna igual a uma camada densa (totalmente conectada)
- É melhor aplicar duas 3x3 sucessivas ou uma 5x5?
  - R: Duas 3x3
    - Menos parâmetros
    - Mais não-linearidade



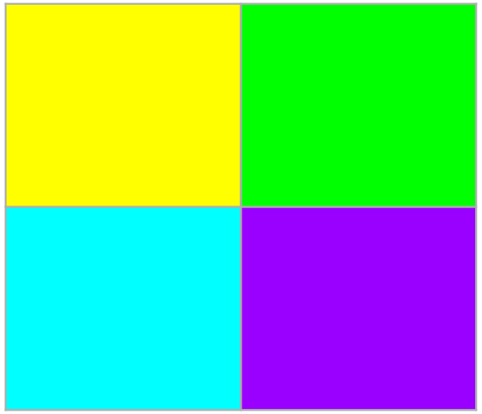
SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

Treinamento

- *Backprop* funciona para convoluções
  - As derivadas parciais precisam ser acumuladas (somadas) para cada entrada



X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}



∂h_{11}	∂h_{12}
∂h_{21}	∂h_{22}

$$\partial W_{11} = X_{11}\partial h_{11} + X_{12}\partial h_{12} + X_{21}\partial h_{21} + X_{22}\partial h_{22}$$

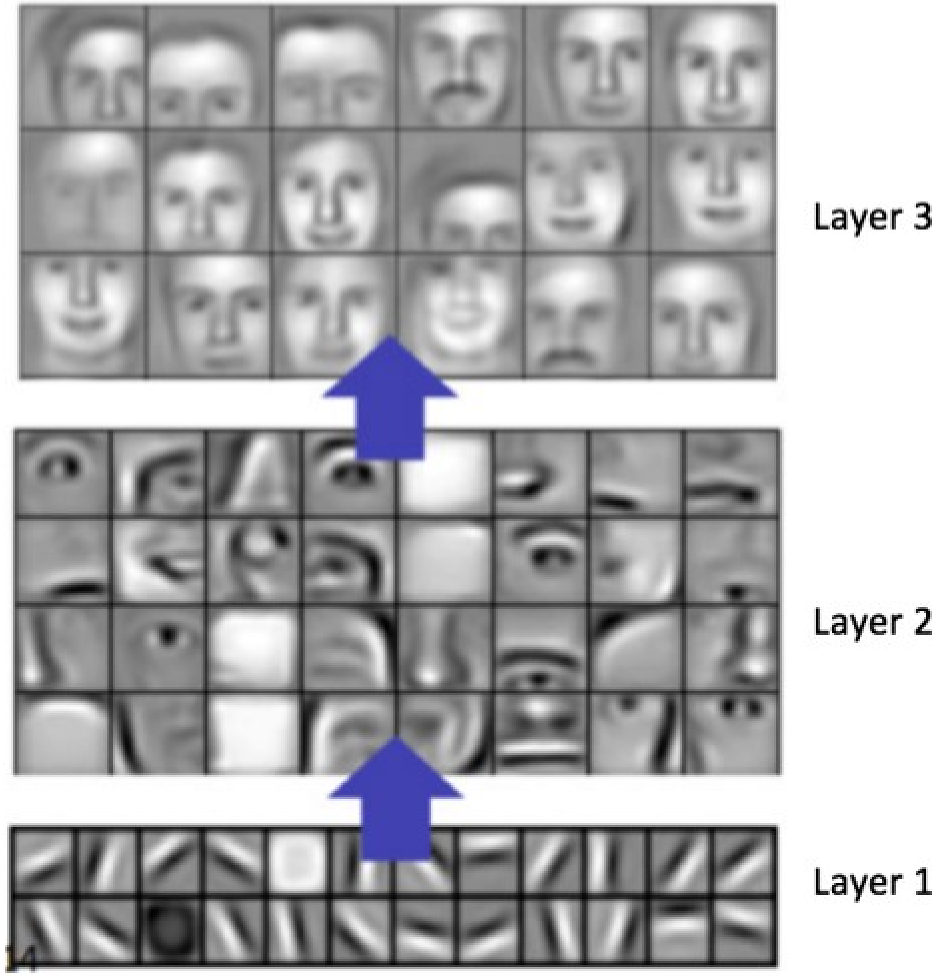
$$\partial W_{12} = X_{12}\partial h_{11} + X_{13}\partial h_{12} + X_{22}\partial h_{21} + X_{23}\partial h_{22}$$

$$\partial W_{21} = X_{21}\partial h_{11} + X_{22}\partial h_{12} + X_{31}\partial h_{21} + X_{32}\partial h_{22}$$

$$\partial W_{22} = X_{22}\partial h_{11} + X_{23}\partial h_{12} + X_{32}\partial h_{21} + X_{33}\partial h_{22}$$

O resultado do treinamento

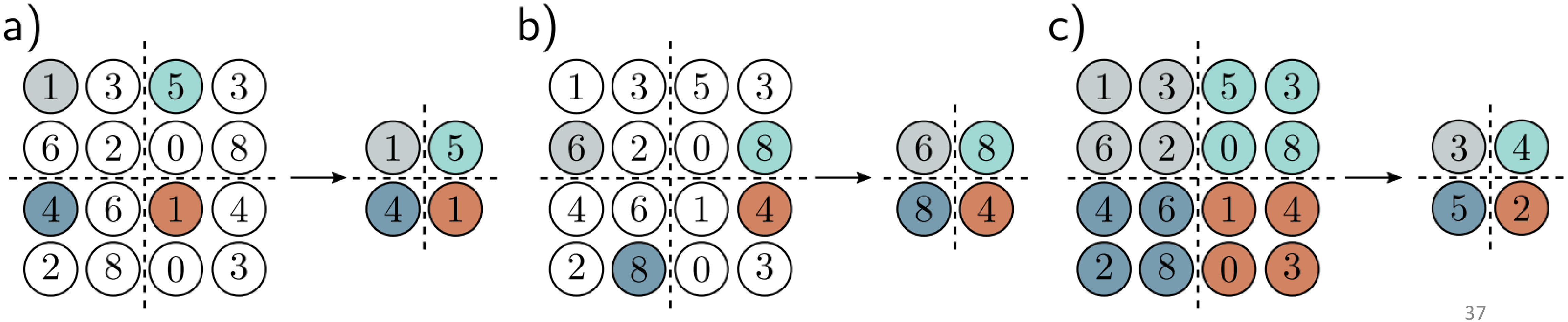
- Os *feature maps* das primeiras camadas mostram arestas, cantos e formas primitivas
- Em camadas mais internas vemos formas mais complexas
  - Repare que aumenta também o campo receptivo



Albawi, S., Mohammed, T.A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1-6.

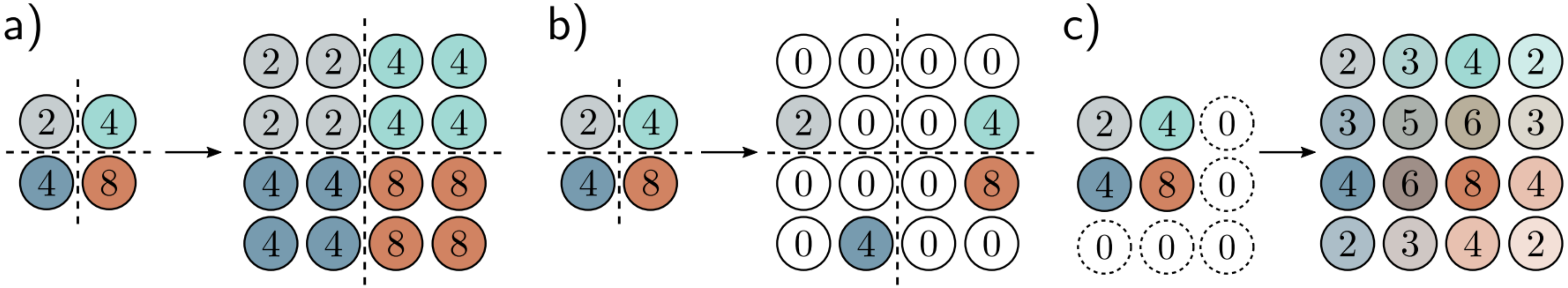
Downsampling

- Subamostrar espacialmente a imagem pode ter o efeito de aumentar o campo receptivo de uma unidade oculta
- As principais formas de subamostragem em CNN são:
  - a) Usar *Stride*
  - b) Max Pooling
  - c) *Average Pooling*



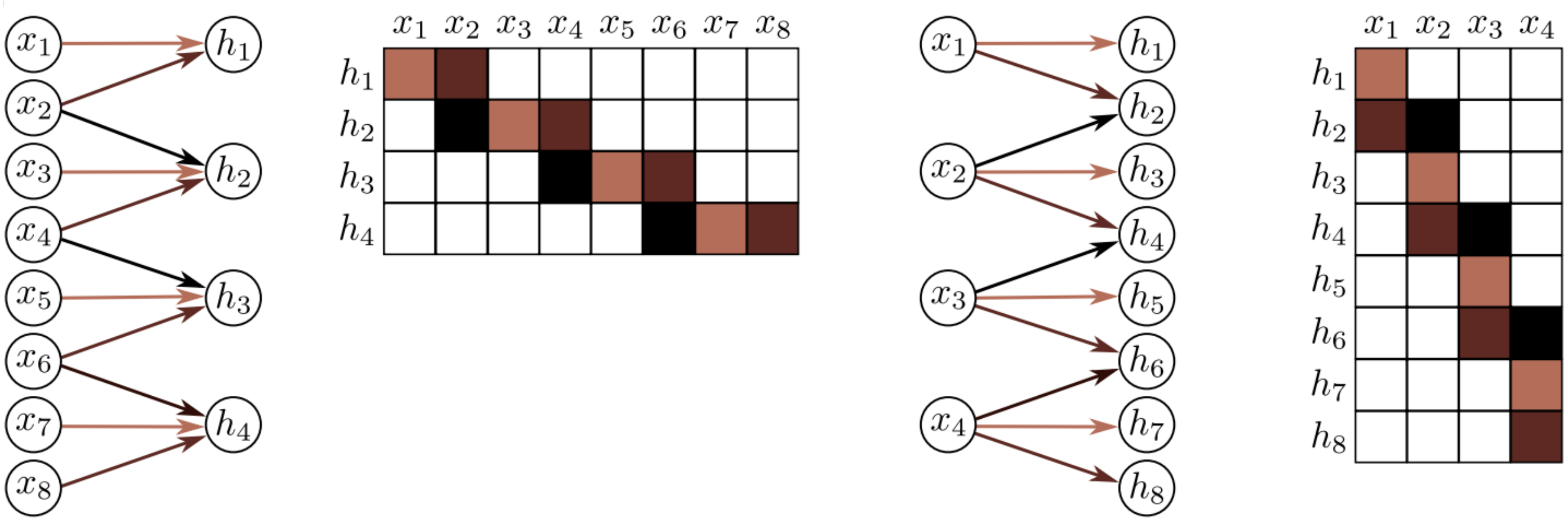
Upsampling

- Sobreamostrar espacialmente a imagem pode ser útil quando a saída do processamento também é uma imagem
- As principais formas de sobreamostragem em CNN são:
  - a) Duplicar os valores conhecidos
  - b) Max Unpooling
  - c) Interpolação Bilinear



Upsampling

- As principais formas de sobreamostragem em CNN são:
  - a) Duplicar os valores conhecidos
  - b) Max Unpooling
  - c) Interpolação Bilinear
  - d) Convolução Transposta



Convolução Transposta - Ex

- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

2	1
3	2

resultado

Fonte: Deep Learning Specialization Andrew NG

Convolução Transposta - Ex

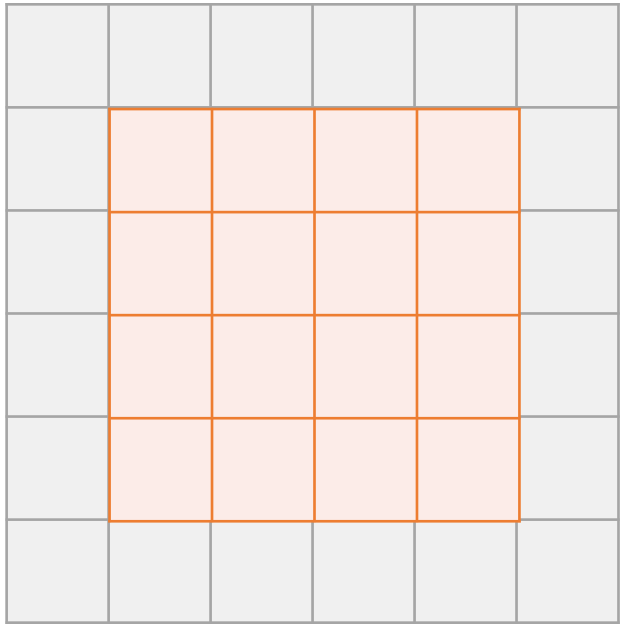
- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

2	1
3	2



resultado

Convolução Transposta - Ex

- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

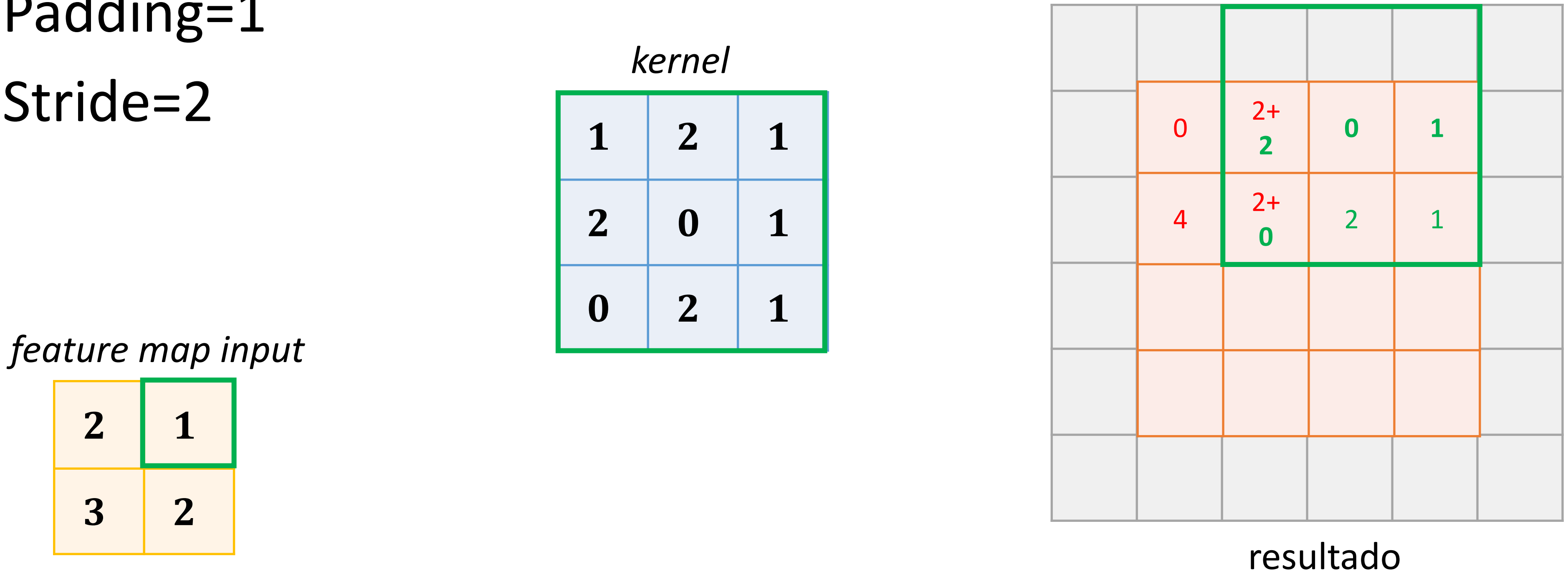
2	1
3	2

	0	2			
	4	2			

resultado

Convolução Transposta - Ex

- Filtro 3x3
- Padding=1
- Stride=2



Convolução Transposta - Ex

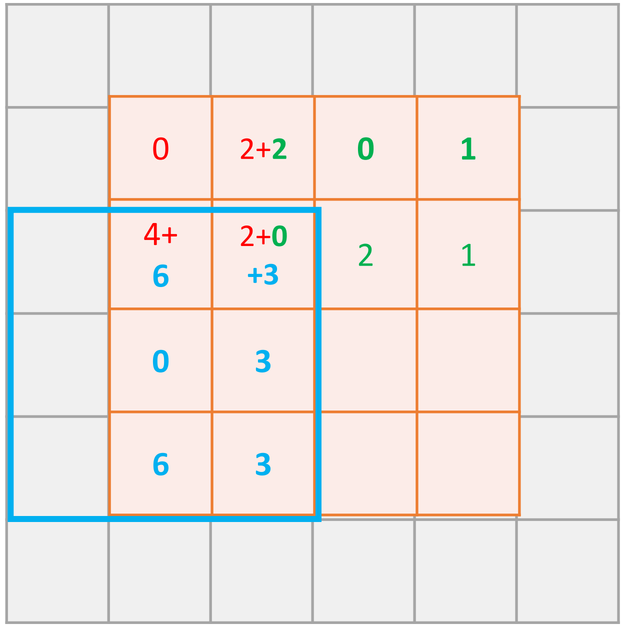
- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

2	1
3	2



resultado

Convolução Transposta - Ex

- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

2	1
3	2

	0	2+2	0	1	
	4+ 6	2+0+ 3+2	2+4	1+2	
	0	3+4	0	2	
	6	3+0	4	2	

resultado

Convolução Transposta - Ex

- Filtro 3x3
- Padding=1
- Stride=2

kernel

1	2	1
2	0	1
0	2	1

feature map input

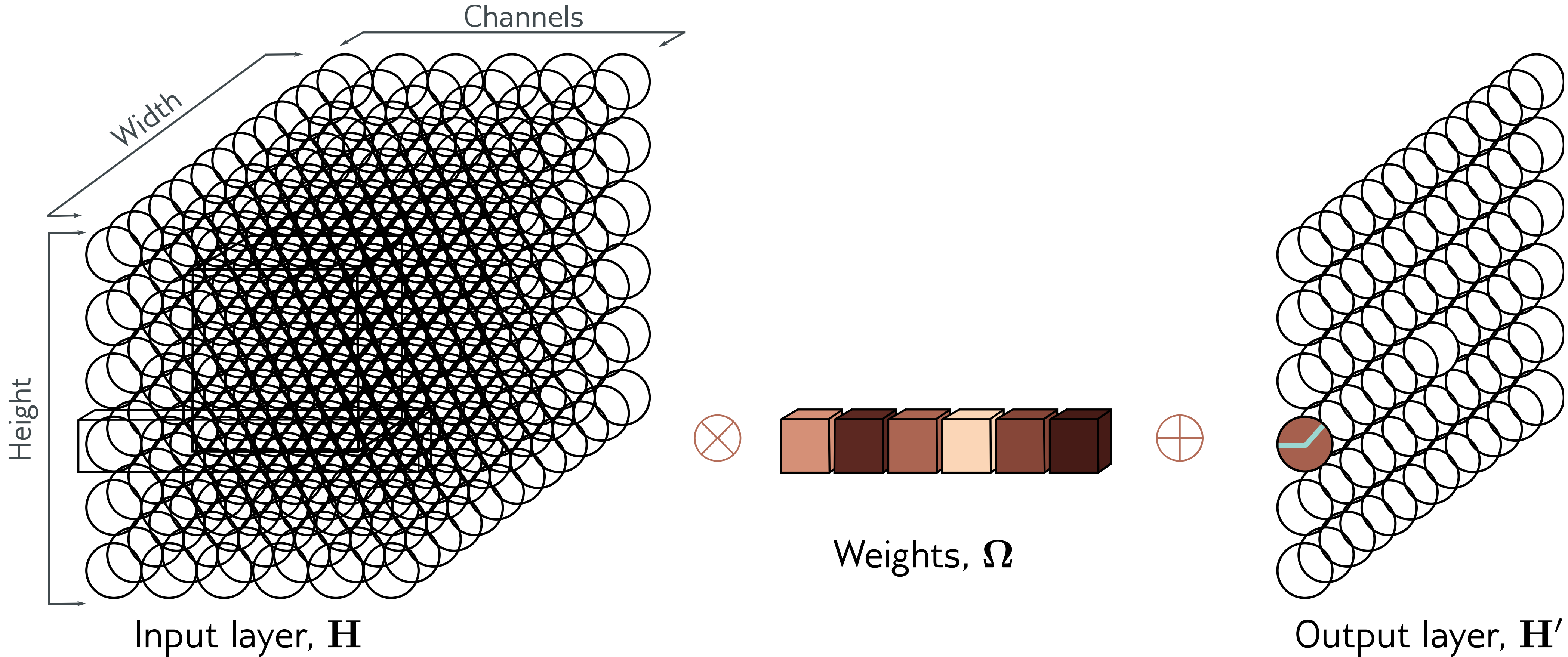
2	1
3	2

0	4	0	1
10	7	6	3
0	7	0	2
6	3	4	2

resultado

Convolução 1x1

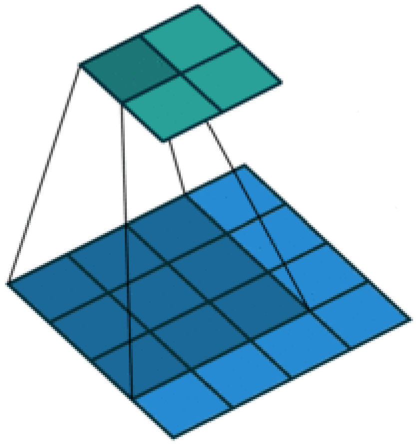
- Algumas vezes queremos mudar o número de canais sem agrupamento espacial
  - Nesses casos podemos aplicar a convolução 1x1



Implementação

- Versão ingênua

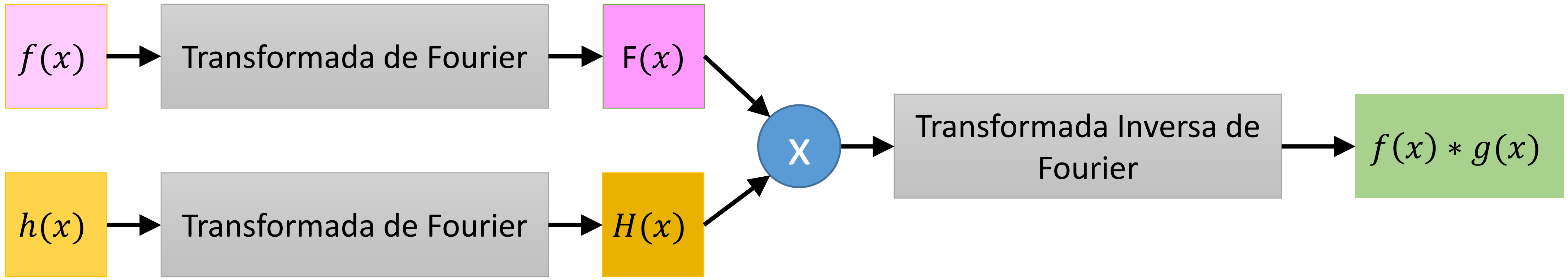
```
• for m in range(0, M): #linhas
    • For n in range(0, N): #colunas
        • For c in range(0, C): #canais
            • For i in range(0,K): #tamanho kernel
                • For j in range(0, K): #tamanho kernel
                    • feat_map_out[m,n] += kernel[i, j, c]*feat_map_in[m+i, n+j, c]
        • feat_map_out[m,n] += bias
```



Implementação

- Versão complexa (e raramente utilizada)
  - Transformada de Fourier
    - Convolução no domínio espacial é igual a multiplicação no domínio da frequência
    - Útil para convoluções com kernels grandes (>11x11)

$$g(x) = f(x) * h(x) \qquad \mathrm{G}(u) = F(u)H(u)$$



VASILACHE, Nicolas et al. Fast convolutional nets with fbfft: A GPU performance evaluation. **arXiv preprint arXiv:1412.7580**, 2014.

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
- Temos que definir matrizes de Toeplitz (matriz de diagonais constantes)
- Uma Matriz de Toeplitz é uma matriz qualquer na qual os valores da diagnonal principal e das subdiagonais são iguais
  - Abaixo um exemplo de uma matriz de Toeplitz

$$\begin{bmatrix} 5 & 27 & 2 & -56 & 0 & 1 \\ -3 & 5 & 27 & 2 & -56 & 0 \\ 8 & -3 & 5 & 27 & 2 & -56 \\ 1 & 8 & -3 & 5 & 27 & 2 \\ -9 & 1 & 8 & -3 & 5 & 27 \\ 3 & -9 & 1 & 8 & -3 & 5 \end{bmatrix}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz ou ainda Matriz de Diagonais Constantes)
  - Passo 1: Definir a entrada e o filtro

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \qquad w = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 2: Calcular o tamanho da saída
    - $\left(X_{heigth} + w_{heigth} - 1\right) \times \left(X_{heigth} + w_{heigth} - 1\right)$
    - $(3 + 2 - 1) \times (3 + 2 - 1) = 4 \times 4$

$$X = \begin{array}{|c|c|c|} \hline \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \boldsymbol{x_{1,3}} \\ \hline \boldsymbol{x_{2,1}} & \boldsymbol{x_{2,2}} & \boldsymbol{x_{2,3}} \\ \hline \boldsymbol{x_{3,1}} & \boldsymbol{x_{3,2}} & \boldsymbol{x_{3,3}} \\ \hline \end{array} \qquad w = \begin{array}{|c|c|} \hline \boldsymbol{w_{1,1}} & \boldsymbol{w_{1,2}} \\ \hline \boldsymbol{w_{2,1}} & \boldsymbol{w_{2,2}} \\ \hline \end{array}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 3: Aplicar *Zero Padding* no filtro para ficar com o tamanho da saída
    - 4×4

$$X = \begin{array}{|c|c|c|} \hline \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \boldsymbol{x_{1,3}} \\ \hline \boldsymbol{x_{2,1}} & \boldsymbol{x_{2,2}} & \boldsymbol{x_{2,3}} \\ \hline \boldsymbol{x_{3,1}} & \boldsymbol{x_{3,2}} & \boldsymbol{x_{3,3}} \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|c|c|} \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline 0 & 0 & 0 & 0 \\ \hline \boldsymbol{w_{1,1}} & \boldsymbol{w_{1,2}} & 0 & 0 \\ \hline \boldsymbol{w_{2,1}} & \boldsymbol{w_{2,2}} & 0 & 0 \\ \hline \end{array}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 4: Criar uma Toeplitz Matrix para cada linha do filtro após o *zero-padding*
    - Importante:
      - Começar da última linha
      - O número de colunas deve ser igual a imagem de entrada da convolução

$$w = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 \\ \boldsymbol{W_{1,1}} & \boldsymbol{W_{1,2}} & 0 & 0 \\ \boldsymbol{W_{2,1}} & \boldsymbol{W_{2,2}} & 0 & 0 \end{bmatrix}$$

$$F_0 = \begin{bmatrix} W_{2,1} & 0 & 0 \\ W_{2,2} & W_{2,1} & 0 \\ 0 & W_{2,2} & W_{2,1} \\ 0 & 0 & W_{2,2} \end{bmatrix} \quad F_1 = \begin{bmatrix} W_{1,1} & 0 & 0 \\ W_{1,2} & W_{1,1} & 0 \\ 0 & W_{1,2} & W_{1,1} \\ 0 & 0 & W_{1,2} \end{bmatrix} \quad F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad F_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 5: Criar uma matriz de bloco de Toeplitz (basicamente reorganizar os filtros anteriores em uma grande matriz
    - Importante:
      - O número de colunas da matriz abaixo deve ser igual ao número de linhas da imagem original
      - O número de linhas deve ser igual as linhas da imagem de saída

Matriz de Bloco Toeplitz $= \begin{bmatrix} F_0 & 0 & 0 \\ F_1 & F_0 & 0 \\ F_2 & F_1 & F_0 \\ F_3 & F_2 & F_1 \end{bmatrix}$

$W_{2,1}$	0	0	0	0	0	0	0	0
$W_{2,2}$	$W_{2,1}$	0	0	0	0	0	0	0
0	$W_{2,2}$	$W_{2,1}$	0	0	0	0	0	0
0	0	$W_{2,2}$	0	0	0	0	0	0
$W_{1,1}$	0	0	$W_{2,1}$	0	0	0	0	0
$W_{1,2}$	$W_{1,1}$	0	$W_{2,2}$	$W_{2,1}$	0	0	0	0
0	$W_{1,2}$	$W_{1,1}$	0	$W_{2,2}$	$W_{2,1}$	0	0	0
0	0	$W_{1,2}$	0	0	$W_{2,2}$	0	0	0
0	0	0	$W_{1,1}$	0	0	$W_{2,1}$	0	0
0	0	0	$W_{1,2}$	$W_{1,1}$	0	$W_{2,2}$	$W_{2,1}$	0
0	0	0	0	$W_{1,2}$	$W_{1,1}$	0	$W_{2,2}$	$W_{2,1}$
0	0	0	0	0	$W_{1,2}$	0	0	$W_{2,2}$
0	0	0	0	0	0	$W_{1,1}$	0	0
0	0	0	0	0	0	$W_{1,2}$	$W_{1,1}$	0
0	0	0	0	0	0	0	$W_{1,2}$	$W_{1,1}$
0	0	0	0	0	0	0	0	$W_{1,2}$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 6: Converter a imagem de entrada em um vetor
    - Cada linha da imagem deve ser transposta e concatenada, da última para a primeira

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix}$$

$$\begin{bmatrix} x_{3,1} \\ x_{3,2} \\ x_{3,3} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \end{bmatrix}$$

Implementação

- Versão Utilizada (im2col, Matriz de Toeplitz)
  - Passo 7: Multiplicar a matriz de Toeplitz de bloco pela imagem vetorizada

16×9

$w_{2,1}$	0	0	0	0	0	0	0	0
$w_{2,2}$	$w_{2,1}$	0	0	0	0	0	0	0
0	$w_{2,2}$	$w_{2,1}$	0	0	0	0	0	0
0	0	$w_{2,2}$	0	0	0	0	0	0
$w_{1,1}$	0	0	$w_{2,1}$	0	0	0	0	0
$w_{1,2}$	$w_{1,1}$	0	$w_{2,2}$	$w_{2,1}$	0	0	0	0
0	$w_{1,2}$	$w_{1,1}$	0	$w_{2,2}$	$w_{2,1}$	0	0	0
0	0	$w_{1,2}$	0	0	$w_{2,2}$	0	0	0
0	0	0	$w_{1,1}$	0	0	$w_{2,1}$	0	0
0	0	0	$w_{1,2}$	$w_{1,1}$	0	$w_{2,2}$	$w_{2,1}$	0
0	0	0	0	$w_{1,2}$	$w_{1,1}$	0	$w_{2,2}$	$w_{2,1}$
0	0	0	0	0	$w_{1,2}$	0	0	$w_{2,2}$
0	0	0	0	0	0	$w_{1,1}$	0	0
0	0	0	0	0	0	$w_{1,2}$	$w_{1,1}$	0
0	0	0	0	0	0	0	$w_{1,2}$	$w_{1,1}$
0	0	0	0	0	0	0	0	$w_{1,2}$

9×1

$x_{3,1}$
$x_{3,2}$
$x_{3,3}$
$x_{2,1}$
$x_{2,2}$
$x_{2,3}$
$x_{1,1}$
$x_{1,2}$
$x_{1,3}$

16×1

$x_{3,1}w_{2,1}$
$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$
$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$
$x_{3,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$
$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$
$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$
$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$
$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$
$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$
$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{1,1}w_{1,1}$
$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$
$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$
$x_{1,3}w_{1,2}$

Implementação

- Passo 8: Fazer reshape do resultado da multiplicação

16×1

$x_{3,1}w_{2,1}$
$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$
$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$
$x_{3,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$
$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$
$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$
$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$
$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$
$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$
$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{1,1}w_{1,1}$
$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$
$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$
$x_{1,3}w_{1,2}$

4×4

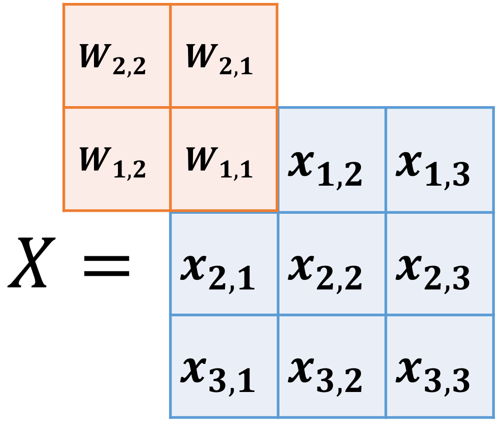
$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \qquad w = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado



$$w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$\boxed{x_{1,1}w_{1,1}}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c}
\hline
w_{2,2} & w_{2,1} & \\
\hline
w_{1,2} & w_{1,1} & \multicolumn{1}{|c|}{x_{1,3}} \\
\hline
x_{2,1} & x_{2,2} & \multicolumn{1}{|c|}{x_{2,3}} \\
\hline
x_{3,1} & x_{3,2} & \multicolumn{1}{|c|}{x_{3,3}} \\
\hline
\end{array}
\qquad\qquad
w = \begin{array}{|c|c|}
\hline
w_{2,2} & w_{2,1} \\
\hline
w_{1,2} & w_{1,1} \\
\hline
\end{array}$$

$x_{1,1}w_{1,1}$	$\boxed{x_{1,1}w_{1,2} + x_{1,2}w_{1,1}}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|} \hline & w_{2,2} & w_{2,1} \\ \hline x_{1,1} & w_{1,2} & w_{1,1} \\ \hline x_{2,1} & x_{2,2} & x_{2,3} \\ \hline x_{3,1} & x_{3,2} & x_{3,3} \\ \hline \end{array} \qquad w = \begin{array}{|c|c|} \hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$\boxed{x_{1,2}w_{1,2} + x_{1,3}w_{1,1}}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|c|} \hline & & w_{2,2} & w_{2,1} \\ \hline x_{1,1} & x_{1,2} & w_{1,2} & w_{1,1} \\ \hline x_{2,1} & x_{2,2} & x_{2,3} & \\ \hline x_{3,1} & x_{3,2} & x_{3,3} & \\ \hline \end{array} \qquad w = \begin{array}{|c|c|} \hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$\boxed{x_{1,3}w_{1,2}}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X \quad \begin{array}{|c|c|c|c|} \hline w_{2,2} & w_{2,1} & x_{1,2} & x_{1,3} \\ \hline w_{1,2} & w_{1,1} & x_{2,2} & x_{2,3} \\ \hline & x_{3,1} & x_{3,2} & x_{3,3} \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$\boxed{x_{2,1}w_{1,1} + x_{1,1}w_{2,1}}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{bmatrix} w_{2,2} & w_{2,1} & x_{1,3} \\ w_{1,2} & w_{1,1} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \qquad w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$\boxed{x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|} \hline x_{1,1} & w_{2,2} & w_{2,1} \\ \hline x_{2,1} & w_{1,2} & w_{1,1} \\ \hline x_{3,1} & x_{3,2} & x_{3,3} \\ \hline \end{array} \qquad w = \begin{array}{|c|c|} \hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$\boxed{x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & w_{2,2} & w_{2,1} \\ x_{2,1} & x_{2,2} & w_{1,2} & w_{1,1} \\ x_{3,1} & x_{3,2} & x_{3,3} & \end{bmatrix} \qquad w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$\boxed{x_{2,3}w_{1,2} + x_{1,3}w_{2,2}}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

		$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
X	$w_{2,2}$	$w_{2,1}$	$x_{2,2}$	$x_{2,3}$
	$w_{1,2}$	$w_{1,1}$	$x_{3,2}$	$x_{3,3}$

$$w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1}$ $+ x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1}$ $+ x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$\boxed{x_{3,1}w_{1,1} + x_{2,1}w_{2,1}}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1}$ $+ x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1}$ $+ x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|}\hline x_{1,1} & x_{1,2} & x_{1,3} \\ \hline w_{2,2} & w_{2,1} & x_{2,3} \\ \hline w_{1,2} & w_{1,1} & x_{3,3} \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|}\hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1}$ $+ x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1}$ $+ x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1}$ $+ x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1}$ $+ x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & w_{2,2} & w_{2,1} \\ x_{3,1} & w_{1,2} & w_{1,1} \end{bmatrix} \qquad w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$\boxed{x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

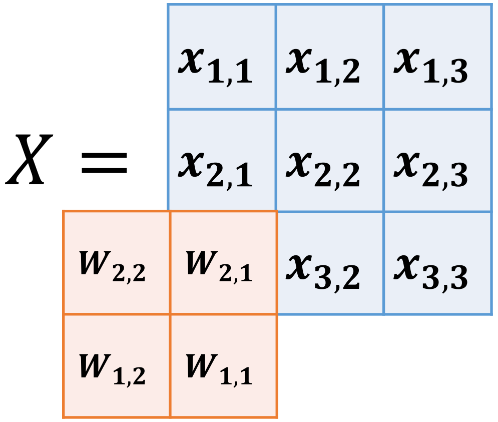
Vamos comparar o resultado

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \\ x_{2,1} & x_{2,2} & w_{2,2} & w_{2,1} \\ x_{3,1} & x_{3,2} & w_{1,2} & w_{1,1} \end{bmatrix}$$

$$w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$\boxed{x_{3,3}w_{1,2} + x_{2,3}w_{2,2}}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado



$$w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$\boxed{x_{3,1}w_{2,1}}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$X =$

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
$w_{2,2}$	$w_{2,1}$	$x_{3,3}$
$w_{1,2}$	$w_{1,1}$	

$$w = \begin{bmatrix} w_{2,2} & w_{2,1} \\ w_{1,2} & w_{1,1} \end{bmatrix}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1}$ $+ x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1}$ $+ x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1}$ $+ x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1}$ $+ x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$\boxed{x_{3,1}w_{2,2} + x_{3,2}w_{2,1}}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|} \hline x_{1,1} & x_{1,2} & x_{1,3} \\ \hline x_{2,1} & x_{2,2} & x_{2,3} \\ \hline x_{3,1} & w_{2,2} & w_{2,1} \\ \hline & w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$\boxed{x_{3,2}w_{2,2} + x_{3,3}w_{2,1}}$	$x_{3,3}w_{2,2}$

Vamos comparar o resultado

$$X = \begin{array}{|c|c|c|c|}\hline x_{1,1} & x_{1,2} & x_{1,3} & \\ \hline x_{2,1} & x_{2,2} & x_{2,3} & \\ \hline x_{3,1} & x_{3,2} & w_{2,2} & w_{2,1} \\ \hline & & w_{1,2} & w_{1,1} \\ \hline \end{array} \qquad w = \begin{array}{|c|c|}\hline w_{2,2} & w_{2,1} \\ \hline w_{1,2} & w_{1,1} \\ \hline \end{array}$$

$x_{1,1}w_{1,1}$	$x_{1,1}w_{1,2} + x_{1,2}w_{1,1}$	$x_{1,2}w_{1,2} + x_{1,3}w_{1,1}$	$x_{1,3}w_{1,2}$
$x_{2,1}w_{1,1} + x_{1,1}w_{2,1}$	$x_{2,1}w_{1,2} + x_{2,2}w_{1,1} + x_{1,1}w_{2,2} + x_{1,2}w_{2,1}$	$x_{2,2}w_{1,2} + x_{2,3}w_{1,1} + x_{1,2}w_{2,2} + x_{1,3}w_{2,1}$	$x_{2,3}w_{1,2} + x_{1,3}w_{2,2}$
$x_{3,1}w_{1,1} + x_{2,1}w_{2,1}$	$x_{3,1}w_{1,2} + x_{3,2}w_{1,1} + x_{2,1}w_{2,2} + x_{2,2}w_{2,1}$	$x_{3,2}w_{1,2} + x_{3,3}w_{1,1} + x_{2,2}w_{2,2} + x_{2,3}w_{2,1}$	$x_{3,3}w_{1,2} + x_{2,3}w_{2,2}$
$x_{3,1}w_{2,1}$	$x_{3,1}w_{2,2} + x_{3,2}w_{2,1}$	$x_{3,2}w_{2,2} + x_{3,3}w_{2,1}$	$\boxed{x_{3,3}w_{2,2}}$

Referências:

- Sugere-se ***fortemente*** a leitura de:
  - Capítulo 10 de Understanding Deep Learning
    - https://udlbook.github.io/udlbook/
- Aula baseada nos seguintes materiais:
  - Capítulo 10 de Understanding Deep Learning. Simon. J. Prince
  - Machine Learning @ Berkeley
  - https://github.com/alisaaalehi/convolution_as_multiplication/tree/main
  - https://github.com/vdumoulin/conv_arithmetic/tree/master