

Aula 24 - Redes Convolucionais (Parte I)

João Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
florindo@unicamp.br

Outline

- 1 Introdução
- 2 Rede Convolucional
- 3 Camada de *Pooling*
- 4 Outras Camadas

Visão Computacional

- Automatizar tarefas da nossa visão.
- Parece, mas não é simples!
- Variação em escala, posição, perspectiva, luz, formato, etc.; oclusão, ...
- EX.: Carro autônomo detectando pedestres e placas.
- Redes convolucionais têm se sobressaído.

Visão Computacional

- Automatizar tarefas da nossa visão.
- Parece, mas não é simples!
- Variação em escala, posição, perspectiva, luz, formato, etc.; oclusão, ...
- EX.: Carro autônomo detectando pedestres e placas.
- Redes convolucionais têm se sobressaído.

Visão Computacional

- Automatizar tarefas da nossa visão.
- Parece, mas não é simples!
- Variação em escala, posição, perspectiva, luz, formato, etc.; oclusão, ...
- EX.: Carro autônomo detectando pedestres e placas.
- Redes convolucionais têm se sobressaído.

Visão Computacional

- Automatizar tarefas da nossa visão.
- Parece, mas não é simples!
- Variação em escala, posição, perspectiva, luz, formato, etc.; oclusão, ...
- EX.: Carro autônomo detectando pedestres e placas.
- Redes convolucionais têm se sobressaído.

Visão Computacional

- Automatizar tarefas da nossa visão.
- Parece, mas não é simples!
- Variação em escala, posição, perspectiva, luz, formato, etc.; oclusão, ...
- EX.: Carro autônomo detectando pedestres e placas.
- Redes convolucionais têm se sobressaído.

Redes Neurais Convolucionais

- Teoria “antiga”, mas ganhou protagonismo a partir de 2012.
- Alex Krizhevsky reduziu erro de 26% para 15% na competição ImageNet (“olimpíada” da Visão Computacional).
- Despertou interesse de empresas como Google (busca de imagens), Amazon (recomendação de produtos), Instagram (sistema de busca), Facebook (marcação de usuários em fotos), etc.

Redes Neurais Convolucionais

- Teoria “antiga”, mas ganhou protagonismo a partir de 2012.
- Alex Krizhevsky reduziu erro de 26% para 15% na competição ImageNet (“olimpíada” da Visão Computacional).
- Despertou interesse de empresas como Google (busca de imagens), Amazon (recomendação de produtos), Instagram (sistema de busca), Facebook (marcação de usuários em fotos), etc.

Redes Neurais Convolucionais

- Teoria “antiga”, mas ganhou protagonismo a partir de 2012.
- Alex Krizhevsky reduziu erro de 26% para 15% na competição ImageNet (“olimpíada” da Visão Computacional).
- Despertou interesse de empresas como Google (busca de imagens), Amazon (recomendação de produtos), Instagram (sistema de busca), Facebook (marcação de usuários em fotos), etc.

Inspiração Biológica

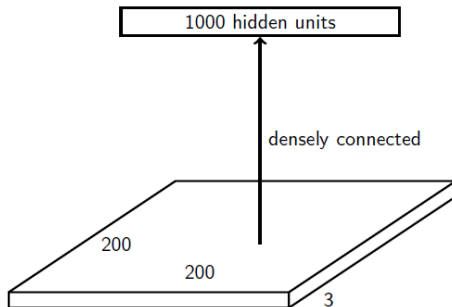
- Cérebro reconhece objetos partindo de primitivas básicas, p.ex., patas e rabo de um cachorro, e vai iterativamente construindo conceitos mais abstratos.
- Experimento de Hubel e Wiesel em 1962: neurônios corticais especializados em certas orientações.

Inspiração Biológica

- Cérebro reconhece objetos partindo de primitivas básicas, p.ex., patas e rabo de um cachorro, e vai iterativamente construindo conceitos mais abstratos.
- Experimento de Hubel e Wiesel em 1962: neurônios corticais especializados em certas orientações.

Motivação Computacional

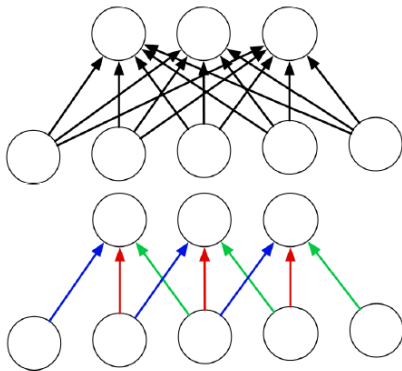
- Problema da rede neural totalmente conectada:



- 120 milhões de parâmetros!
 - Muito dado para treinar.
 - Custo computacional.

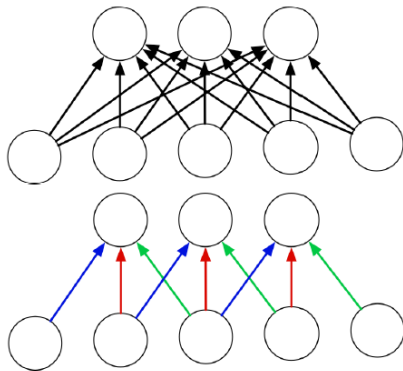
Motivação Computacional

- Atributos importantes em uma posição (bordas, cantos, contornos, etc.) tendem a ser em outras também (**estrutura compartilhada**).
- Objeto deslocado não é um novo objeto (*invariância*).
- SOLUÇÃO: Unidades escondidas olhando só para uma região e compartilhando pesos.



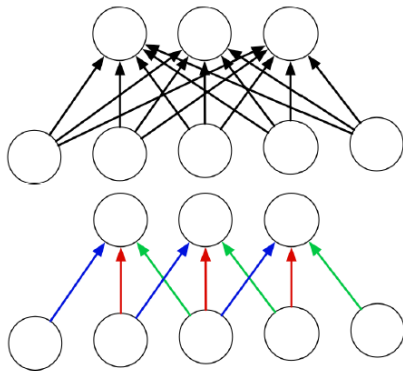
Motivação Computacional

- Atributos importantes em uma posição (bordas, cantos, contornos, etc.) tendem a ser em outras também (**estrutura compartilhada**).
- Objeto deslocado não é um novo objeto (*invariância*).
- SOLUÇÃO: Unidades escondidas olhando só para uma região e compartilhando pesos.



Motivação Computacional

- Atributos importantes em uma posição (bordas, cantos, contornos, etc.) tendem a ser em outras também (**estrutura compartilhada**).
- Objeto deslocado não é um novo objeto (*invariância*).
- SOLUÇÃO: Unidades escondidas olhando só para uma região e compartilhando pesos.

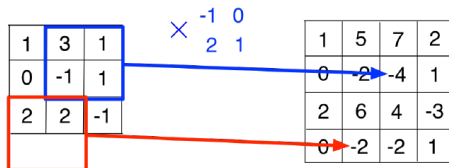


Outline

- 1 Introdução
- 2 Rede Convolucional
- 3 Camada de *Pooling*
- 4 Outras Camadas

Convolução

$$\begin{array}{|c|c|c|} \hline 1 & 3 & 1 \\ \hline 0 & -1 & 1 \\ \hline 2 & 2 & -1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 0 & -1 \\ \hline \end{array}$$



Convolução


 $*$

0	1	0
1	4	1
0	1	0


 $*$

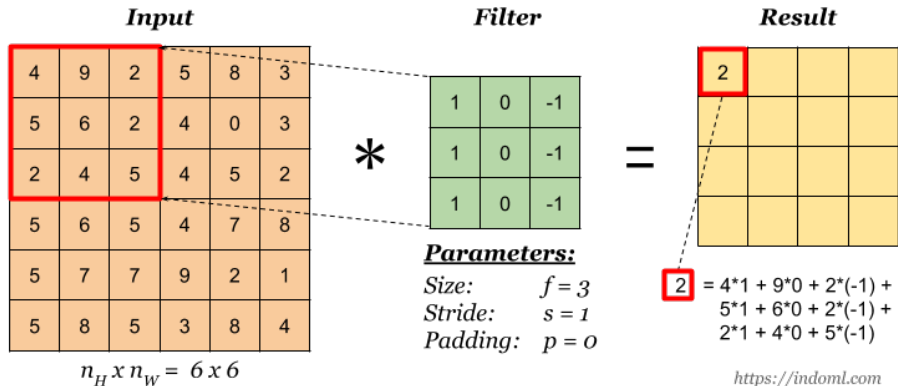
0	-1	0
-1	8	-1
0	-1	0


 $*$

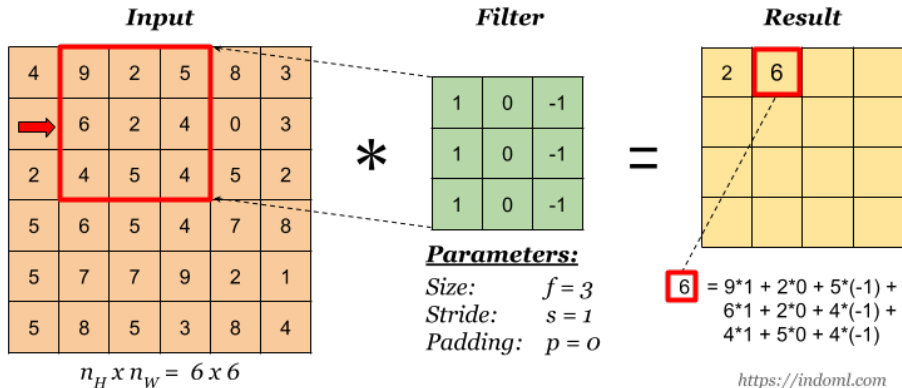
0	-1	0
-1	4	-1
0	-1	0



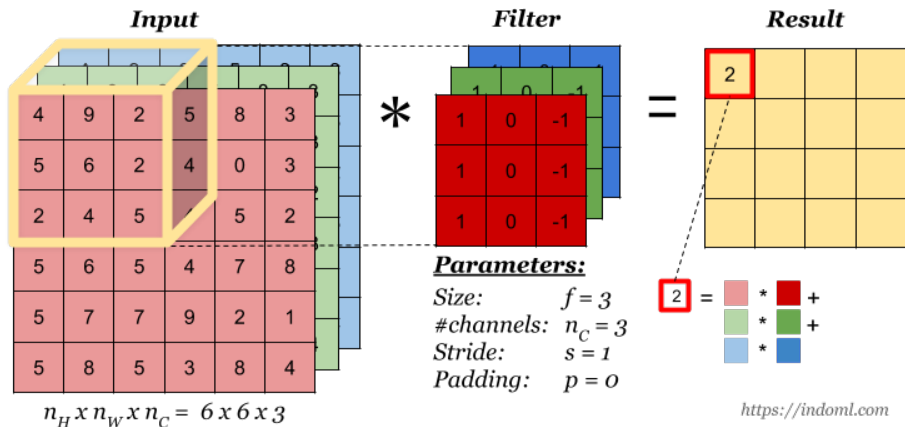
Camada Convolucional



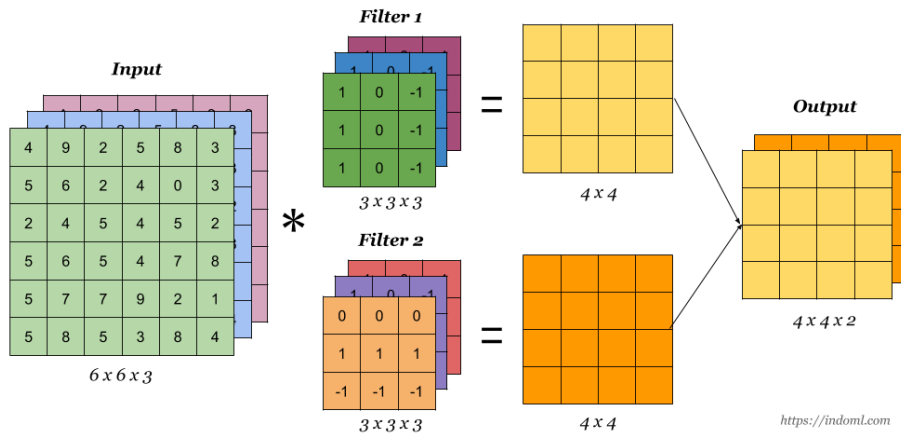
Camada Convolutiva



Camada Convolucional

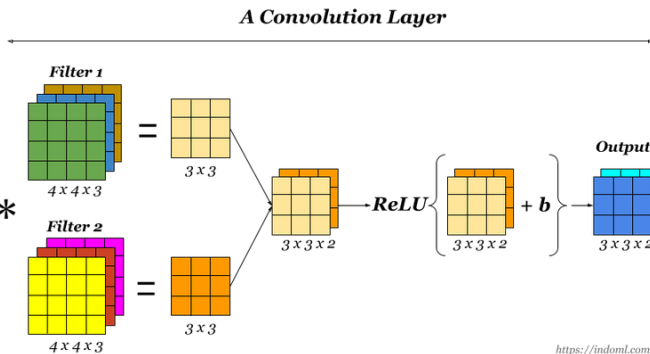


Camada Convolucional



Camada Convolucional

Convolução + $\text{ReLU}(z) = \max(0, z)$:



Camada Convolutacional

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutacional

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutacional

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutacional

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutiva

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutiva

- Imagem inteira introduzida como entrada (pode ser colorida, multicanais, etc.).
- Convolução com um filtro (**neurônio**).
- Suporte do filtro: **campo receptivo**.
- Resultado da convolução: **mapa de ativação** ou **mapa de atributos**.
- Ligeiramente menor que a imagem original porque as bordas são descartadas.
- Cada filtro pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc.

Camada Convolutiva

- Sucessivas convoluções detectam atributos de mais alto nível, p.ex., semi-círculos (reta+curva), quadrados (4 retas), etc.
- Em níveis mais altos ainda pode detectar um texto manuscrito ou um objeto de uma cor específica.
- Campos receptivos se tornam mais e mais amplos (informação mais global).

OBSERVAÇÃO

Valores (pesos) dos filtros são ajustados durante o treinamento por *backpropagation*.

Camada Convolutiva

- Sucessivas convoluções detectam atributos de mais alto nível, p.ex., semi-círculos (reta+curva), quadrados (4 retas), etc.
- Em níveis mais altos ainda pode detectar um texto manuscrito ou um objeto de uma cor específica.
- Campos receptivos se tornam mais e mais amplos (informação mais global).

OBSERVAÇÃO

Valores (pesos) dos filtros são ajustados durante o treinamento por *backpropagation*.

Camada Convolutiva

- Sucessivas convoluções detectam atributos de mais alto nível, p.ex., semi-círculos (reta+curva), quadrados (4 retas), etc.
- Em níveis mais altos ainda pode detectar um texto manuscrito ou um objeto de uma cor específica.
- Campos receptivos se tornam mais e mais amplos (informação mais global).

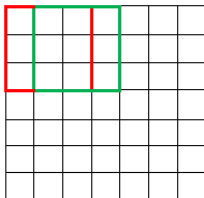
OBSERVAÇÃO

Valores (pesos) dos filtros são ajustados durante o treinamento por *backpropagation*.

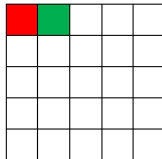
Camada Convolutiva - Detalhes

- *Stride* (deslocamento) e *Padding* (preenchimento).
- *Stride* é o espaçamento entre as máscaras.

7 x 7 Input Volume

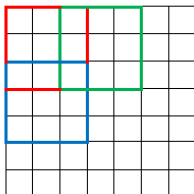


5 x 5 Output Volume



Camada Convolutiva - Detalhes

7 x 7 Input Volume

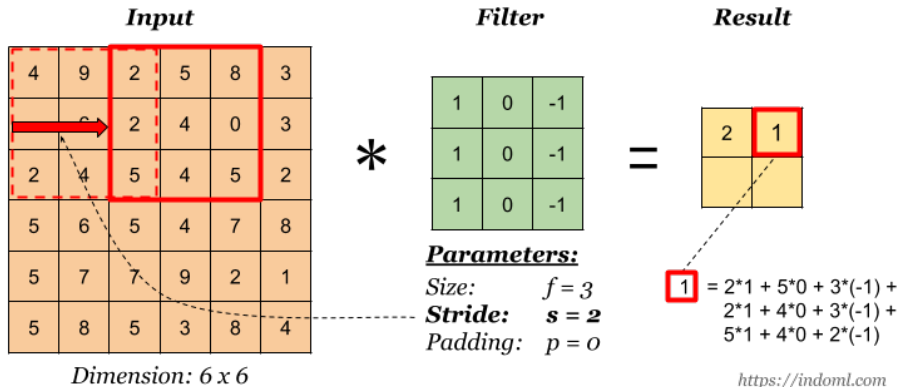


3 x 3 Output Volume



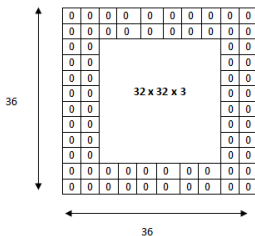
- Maior *stride* = menos sobreposição entre campos receptivos e saída com menor dimensão.

Camada Convolutiva - Detalhes



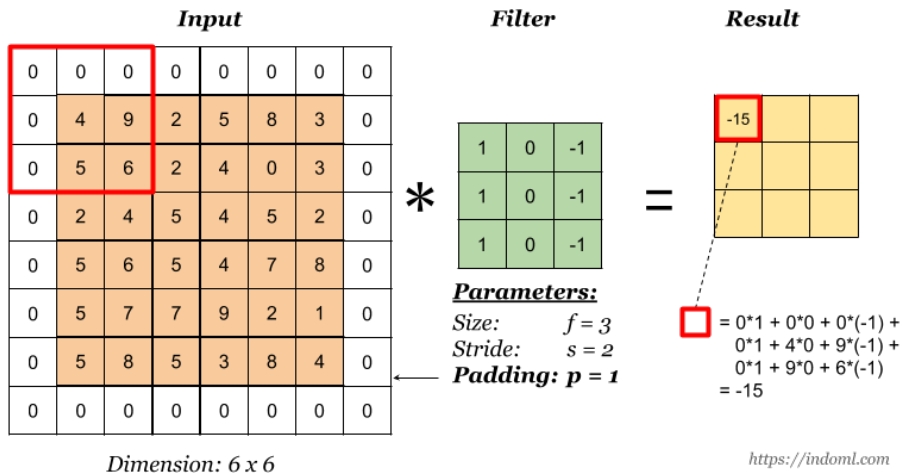
Camada Convolutiva - Detalhes

- *Padding* é o preenchimento (com zeros por exemplo) para deixar a imagem convoluída com o mesmo tamanho da original.



The input volume is $32 \times 32 \times 3$. If we imagine two borders of zeros around the volume, this gives us a $36 \times 36 \times 3$ volume. Then, when we apply our conv layer with our three $5 \times 5 \times 3$ filters and a stride of 1, then we will also get a $32 \times 32 \times 3$ output volume.

Camada Convolutiva - Detalhes



Hiper-parâmetros

- Número de camadas convolucionais e totalmente conectadas, tamanho dos filtros, *stride*, *padding*, etc.
- Depende do problema: tamanho, número, complexidade e tipo de imagem, processamento envolvido, etc.
- Altamente empírico: conjunto de validação.

Hiper-parâmetros

- Número de camadas convolucionais e totalmente conectadas, tamanho dos filtros, *stride*, *padding*, etc.
- Depende do problema: tamanho, número, complexidade e tipo de imagem, processamento envolvido, etc.
- Altamente empírico: conjunto de validação.

Hiper-parâmetros

- Número de camadas convolucionais e totalmente conectadas, tamanho dos filtros, *stride*, *padding*, etc.
- Depende do problema: tamanho, número, complexidade e tipo de imagem, processamento envolvido, etc.
- Altamente empírico: conjunto de validação.

Camada ReLU

- *Rectified Linear Units:*

$$f(x) = \max(0, x).$$

- Superaram a sigmoide, tanh e outras funções de ativação.
- Treinamento muito mais rápido com desempenho parecido.
- Atenua efeito de dissipação do gradiente (*vanishing gradient*): gradiente diminui exponencialmente ao longo das camadas tornando o aprendizado muito lento.

Camada ReLU

- *Rectified Linear Units:*

$$f(x) = \max(0, x).$$

- Superaram a sigmoide, tanh e outras funções de ativação.
- Treinamento muito mais rápido com desempenho parecido.
- Atenua efeito de dissipação do gradiente (*vanishing gradient*): gradiente diminui exponencialmente ao longo das camadas tornando o aprendizado muito lento.

Camada ReLU

- *Rectified Linear Units:*

$$f(x) = \max(0, x).$$

- Superaram a sigmoide, tanh e outras funções de ativação.
- Treinamento muito mais rápido com desempenho parecido.
- Atenua efeito de dissipação do gradiente (*vanishing gradient*): gradiente diminui exponencialmente ao longo das camadas tornando o aprendizado muito lento.

Camada ReLU

- *Rectified Linear Units*:

$$f(x) = \max(0, x).$$

- Superaram a sigmoide, tanh e outras funções de ativação.
- Treinamento muito mais rápido com desempenho parecido.
- Atenua efeito de dissipação do gradiente (*vanishing gradient*): gradiente diminui exponencialmente ao longo das camadas tornando o aprendizado muito lento.

Outline

- 1 Introdução
- 2 Rede Convolutacional
- 3 Camada de *Pooling*
- 4 Outras Camadas

Camada de *Pooling*

- Camadas convolucionais são **equivariantes**: se a entrada é transladada, a saída também será.
- Mas desejamos que nossa predição seja **invariante**: translação não afeta.
- SOLUÇÃO: Camada de *pooling*.

Camada de *Pooling*

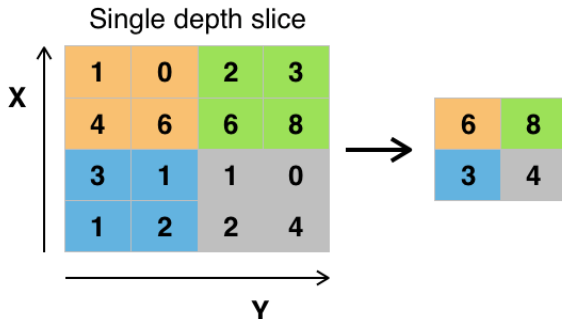
- Camadas convolucionais são **equivariantes**: se a entrada é transladada, a saída também será.
- Mas desejamos que nossa predição seja **invariante**: translação não afeta.
- SOLUÇÃO: Camada de *pooling*.

Camada de *Pooling*

- Camadas convolucionais são **equivariantes**: se a entrada é transladada, a saída também será.
- Mas desejamos que nossa predição seja **invariante**: translação não afeta.
- SOLUÇÃO: Camada de *pooling*.

Camada de *Pooling*

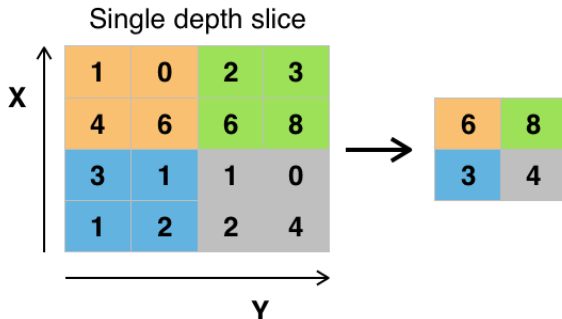
- Ou camada de sub-amostragem (*downsampling*).
- Mais comum: *maxpooling* - filtro (normalmente 2×2) e *stride* do mesmo tamanho detectando maior elemento da região.



Example of Maxpool with a 2×2 filter and a stride of 2

Camada de Pooling

- Ou camada de sub-amostragem (*downsampling*).
- Mais comum: *maxpooling* - filtro (normalmente 2×2) e *stride* do mesmo tamanho detectando maior elemento da região.



Example of Maxpool with a 2×2 filter and a stride of 2

Camada de *Pooling*

- Alternativas são, por exemplo, *average pooling* e L^2 -*pooling*.
- Motivação: dado que o atributo está na imagem convoluída (resposta alta ao filtro respectivo), não importa tanto a localização exata.
- Reduz o número de pesos (custo computacional), compactando os dados.
- Controla o sobre-treinamento (overfitting): perda de generalização.

Camada de *Pooling*

- Alternativas são, por exemplo, *average pooling* e L^2 -*pooling*.
- Motivação: dado que o atributo está na imagem convoluída (resposta alta ao filtro respectivo), não importa tanto a localização exata.
- Reduz o número de pesos (custo computacional), compactando os dados.
- Controla o sobre-treinamento (overfitting): perda de generalização.

Camada de *Pooling*

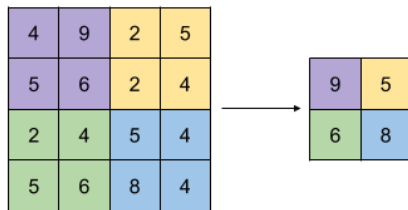
- Alternativas são, por exemplo, *average pooling* e L^2 -*pooling*.
- Motivação: dado que o atributo está na imagem convoluída (resposta alta ao filtro respectivo), não importa tanto a localização exata.
- Reduz o número de pesos (custo computacional), compactando os dados.
- Controla o sobre-treinamento (overfitting): perda de generalização.

Camada de *Pooling*

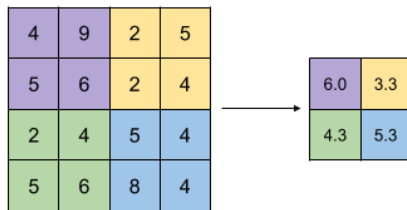
- Alternativas são, por exemplo, *average pooling* e L^2 -*pooling*.
- Motivação: dado que o atributo está na imagem convoluída (resposta alta ao filtro respectivo), não importa tanto a localização exata.
- Reduz o número de pesos (custo computacional), compactando os dados.
- Controla o sobre-treinamento (overfitting): perda de generalização.

Camada de Pooling

Max Pooling

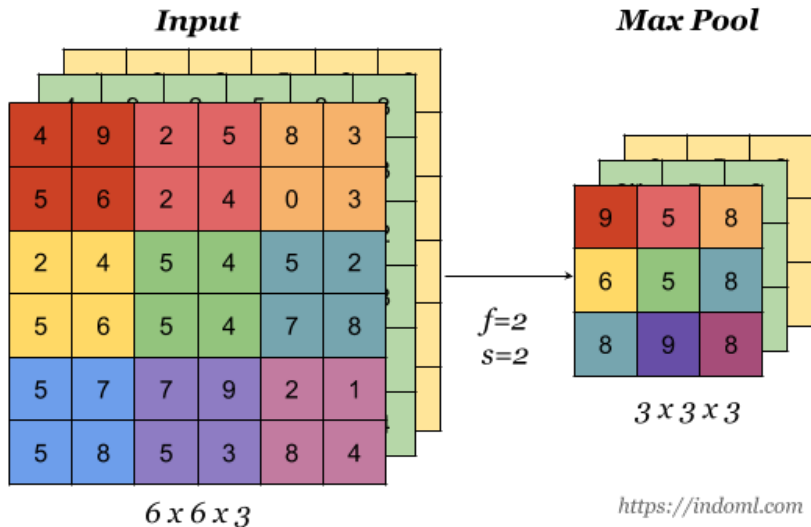


Avg Pooling



<https://indoml.com>

Camada de Pooling



Outline

- 1 Introdução
- 2 Rede Convolucional
- 3 Camada de *Pooling*
- 4 Outras Camadas

Camada de *Dropout*

- Esta camada é usada apenas na fase de treinamento, não no teste.
- Elimina um conjunto aleatório de ativações na camada setando para zero.
- Força a rede a ser “redundante”, i.e, a classificar corretamente mesmo que algumas ativações sejam zeradas.
- Atenua o *overfitting*.

Camada de *Dropout*

- Esta camada é usada apenas na fase de treinamento, não no teste.
- Elimina um conjunto aleatório de ativações na camada setando para zero.
- Força a rede a ser “redundante”, i.e, a classificar corretamente mesmo que algumas ativações sejam zeradas.
- Atenua o *overfitting*.

Camada de *Dropout*

- Esta camada é usada apenas na fase de treinamento, não no teste.
- Elimina um conjunto aleatório de ativações na camada setando para zero.
- Força a rede a ser “redundante”, i.e, a classificar corretamente mesmo que algumas ativações sejam zeradas.
- *Atenua o overfitting.*

Camada de *Dropout*

- Esta camada é usada apenas na fase de treinamento, não no teste.
- Elimina um conjunto aleatório de ativações na camada setando para zero.
- Força a rede a ser “redundante”, i.e, a classificar corretamente mesmo que algumas ativações sejam zeradas.
- Atenua o *overfitting*.

Batch Normalization

- Normalização (*z-score*) das ativações por *mini-batch*.
- Torna o treinamento mais rápido e estável.
- Atenua o *covariante shift*: efeito da aleatoriedade dos dados entrada e pesos iniciais.
- Normalmente aplicado após uma camada totalmente conectada ou convolucional e antes de uma não-linearidade (p.ex. ReLU).

Batch Normalization

- Normalização (*z-score*) das ativações por *mini-batch*.
- Torna o treinamento mais rápido e estável.
- Atenua o *covariante shift*: efeito da aleatoriedade dos dados entrada e pesos iniciais.
- Normalmente aplicado após uma camada totalmente conectada ou convolucional e antes de uma não-linearidade (p.ex. ReLU).

Batch Normalization

- Normalização (*z-score*) das ativações por *mini-batch*.
- Torna o treinamento mais rápido e estável.
- Atenua o *covariante shift*: efeito da aleatoriedade dos dados entrada e pesos iniciais.
- Normalmente aplicado após uma camada totalmente conectada ou convolucional e antes de uma não-linearidade (p.ex. ReLU).

Batch Normalization

- Normalização (*z-score*) das ativações por *mini-batch*.
- Torna o treinamento mais rápido e estável.
- Atenua o *covariante shift*: efeito da aleatoriedade dos dados entrada e pesos iniciais.
- Normalmente aplicado após uma camada totalmente conectada ou convolucional e antes de uma não-linearidade (p.ex. ReLU).

Camada Totalmente Conectada

- Ao final, temos a camada totalmente conectada.
- Estabelece quais atributos de alto nível (última camada convolucional) mais se correlacionam com a classe do objeto.
- EX.: Imagem de um cachorro teria resposta alta no mapa de ativação correspondente a patas e rabo. Um pássaro teria resposta alta para bico e asas.

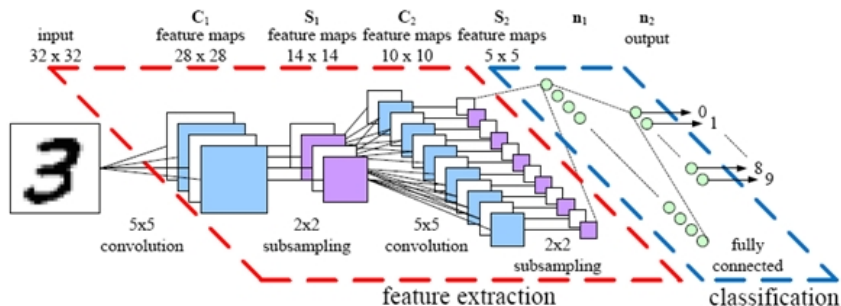
Camada Totalmente Conectada

- Ao final, temos a camada totalmente conectada.
- Estabelece quais atributos de alto nível (última camada convolucional) mais se correlacionam com a classe do objeto.
- EX.: Imagem de um cachorro teria resposta alta no mapa de ativação correspondente a patas e rabo. Um pássaro teria resposta alta para bico e asas.

Camada Totalmente Conectada

- Ao final, temos a camada totalmente conectada.
- Estabelece quais atributos de alto nível (última camada convolucional) mais se correlacionam com a classe do objeto.
- EX.: Imagem de um cachorro teria resposta alta no mapa de ativação correspondente a patas e rabo. Um pássaro teria resposta alta para bico e asas.

Redes Neurais Convolucionais



Redes Neurais Convolucionais

