

Redes Neurais Convolucionais e Recorrentes

Aula 6

Magno TF Severino

PADS - Aprendizagem Estatística de Máquina II

Objetivos de aprendizagem

- Compreender o funcionamento dos modelos de redes neurais convolucionais (CNN) e como implementá-los;
- Compreender o funcionamento dos modelos de redes neurais recorrentes (RNN) e como implementá-los;

Rede neural com uma camada

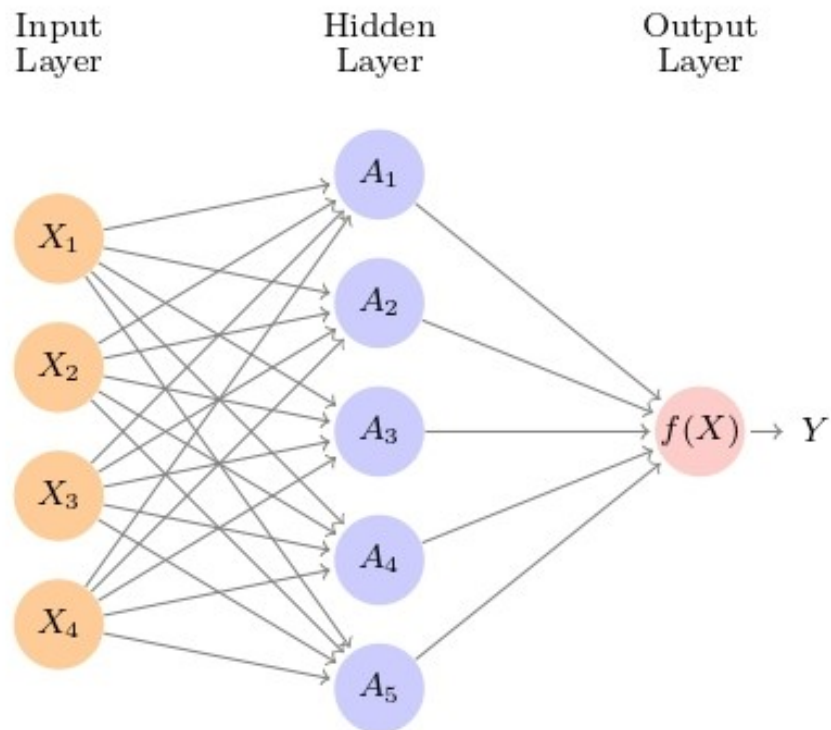


Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Rede neural com várias camadas

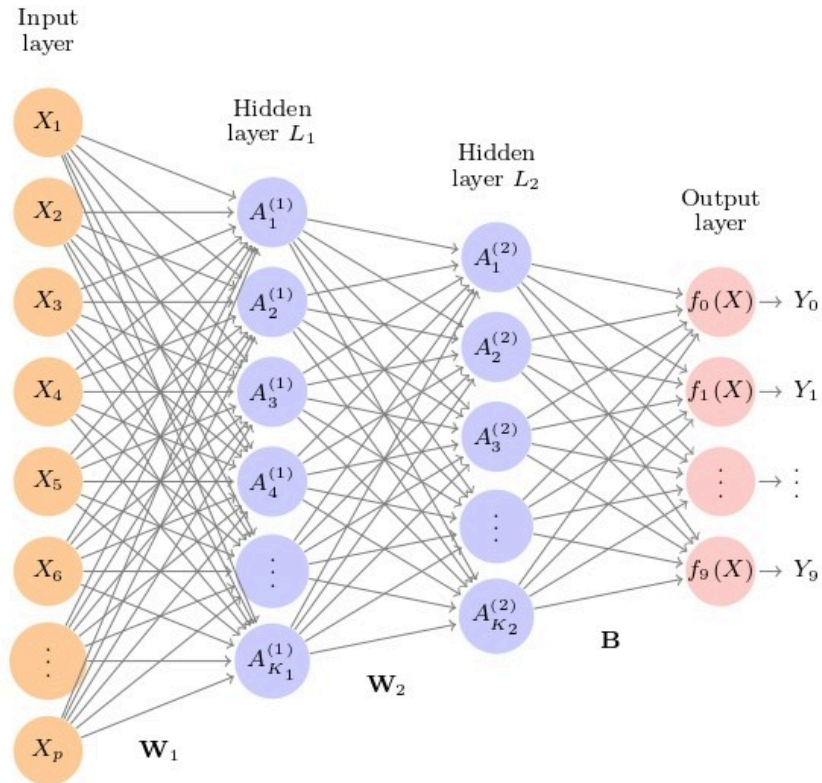
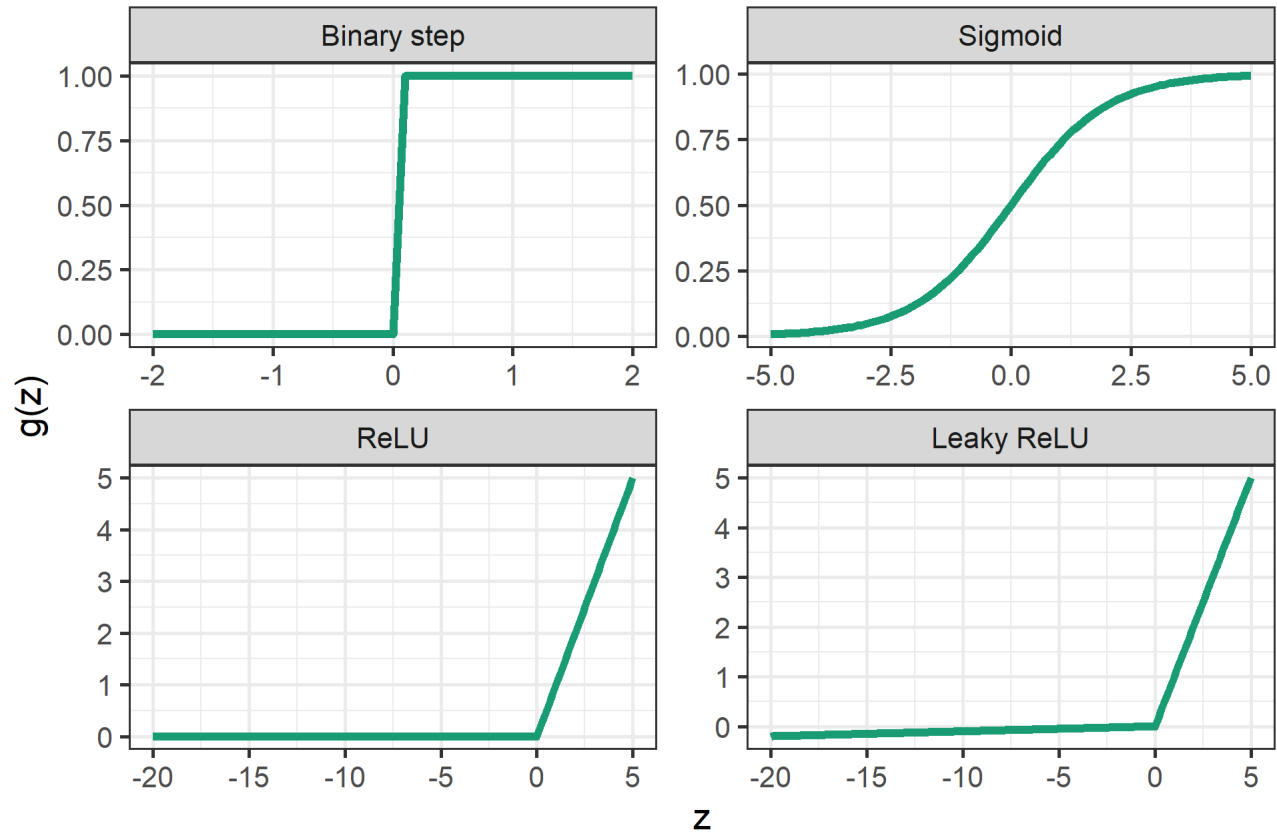


Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Funções de ativação



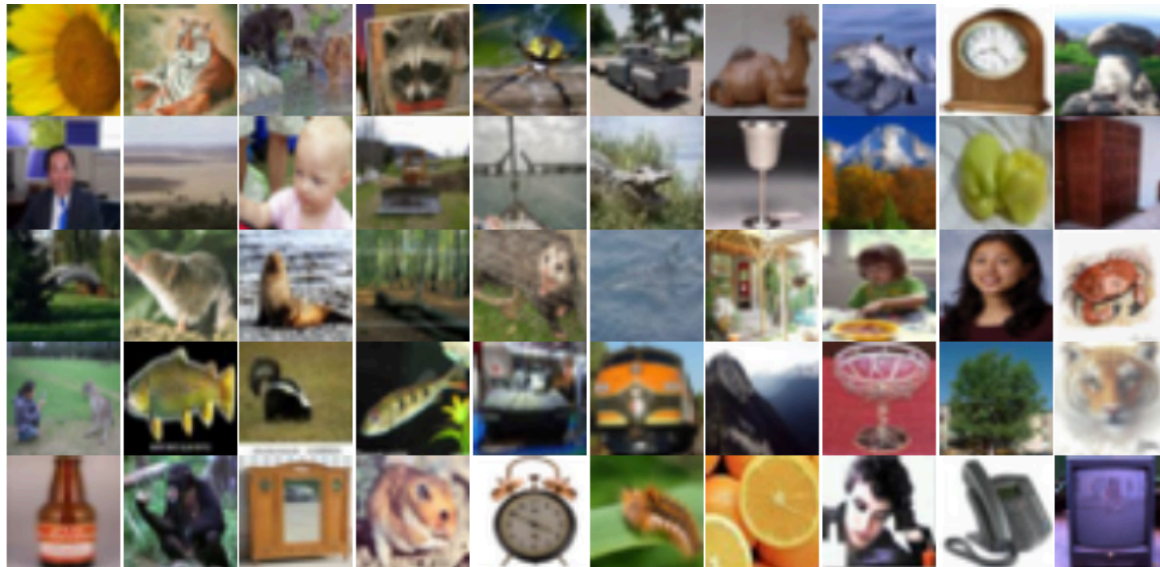
Lista de funções de ativação na [Wikipedia](#).

Redes neurais convolucionais (CNN)

- Se popularizaram por volta de 2010;
- Grandes bases de dados de imagens classificadas;
- As *redes neurais convolucionais* (*convolutional neural networks* - *CNNs*) se mostraram poderosas para classificar imagens;
- Em certo grau, as CNNs simulam a maneira de seres humanos classificam imagens, reconhecendo características e padrões específicos que distinguem cada classe.

Base de dados cifar100

- 60000 imagens classificadas em 20 superclasses, com cinco classes por subclasse;
- Exemplo: mamíferos aquáticos: (castor, golfinho, lontra, foca, baleia);
- Cada imagem tem 32×32 pixels com números entre 0 e 255 cada, representando os canais *red*, *green* and *blue* (RGB).
- Veja uma amostra das imagens:



Como representar uma imagem?

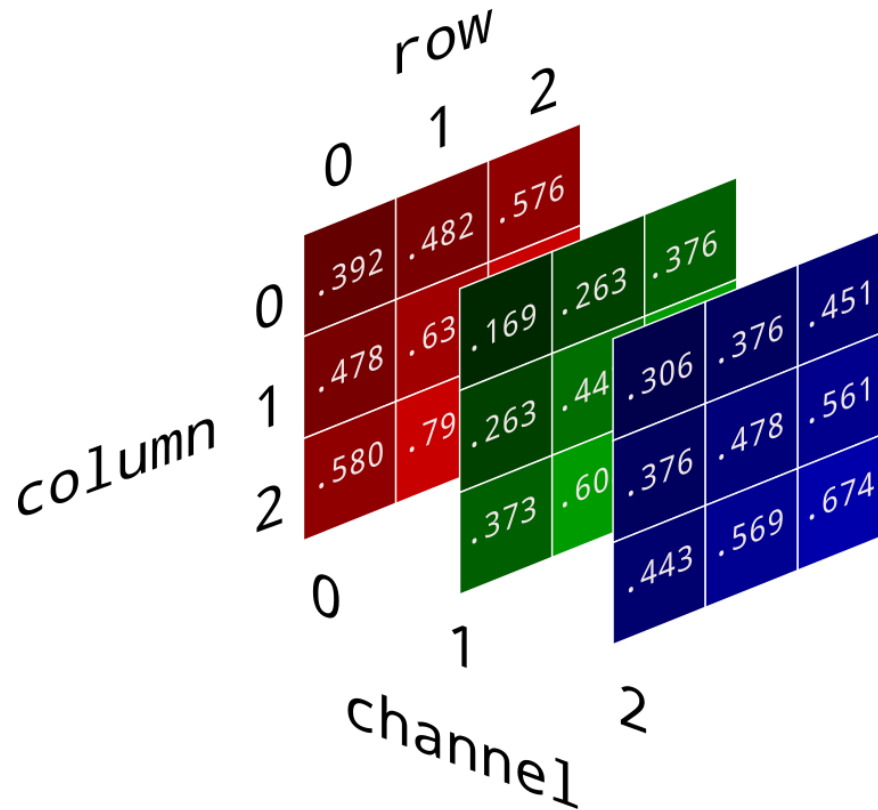


Figura da página [KD Nuggets](#).

A ideia por traz das CNNs

- Primeiro, a rede identifica características de baixo nível da imagem (bordas, marcas de cores, etc);
- Essas características de baixo nível são então combinadas para formar características de alto nível (orelhas, olhos, etc);
- Eventualmente, a presença/falta dessas características contribui com a probabilidade de classificação em uma determinada categoria.
- Mas como uma CNN faz isso?
Utilizando dois tipos de camadas ocultas: *convolution layers* e *pooling layers*.

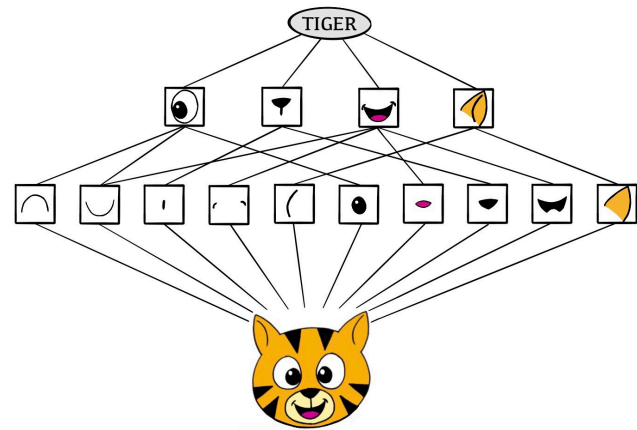


Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Convolution layers

- Uma camada de convolução é composta por um grande número de filtros de convolução;
- Cada filtro é um template que determina se uma característica local está presente em uma imagem;
- Operação de convolução: multiplicar elementos de matrizes e somá-los posteriormente.

Convolution layers

Considere a seguinte imagem 4×3 :

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}.$$

E o seguinte filtro 2×2 :

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}.$$

A operação de convolução gera o seguinte resultado

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}.$$

O elemento na primeira linha e primeira coluna da matriz é obtido ao multiplicar cada

Convolution layers - exemplo

Imagem original

$$\begin{bmatrix} a & a & a & 0 & 0 & 0 \\ a & a & a & 0 & 0 & 0 \\ a & a & a & 0 & 0 & 0 \\ a & a & a & 0 & 0 & 0 \end{bmatrix}.$$

Filtro de convolução

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}.$$

Imagem convolucionada

$$\begin{bmatrix} 0 & 3a & 3a & 0 \\ 0 & 3a & 3a & 0 \end{bmatrix}.$$

Convolution layers - exemplo

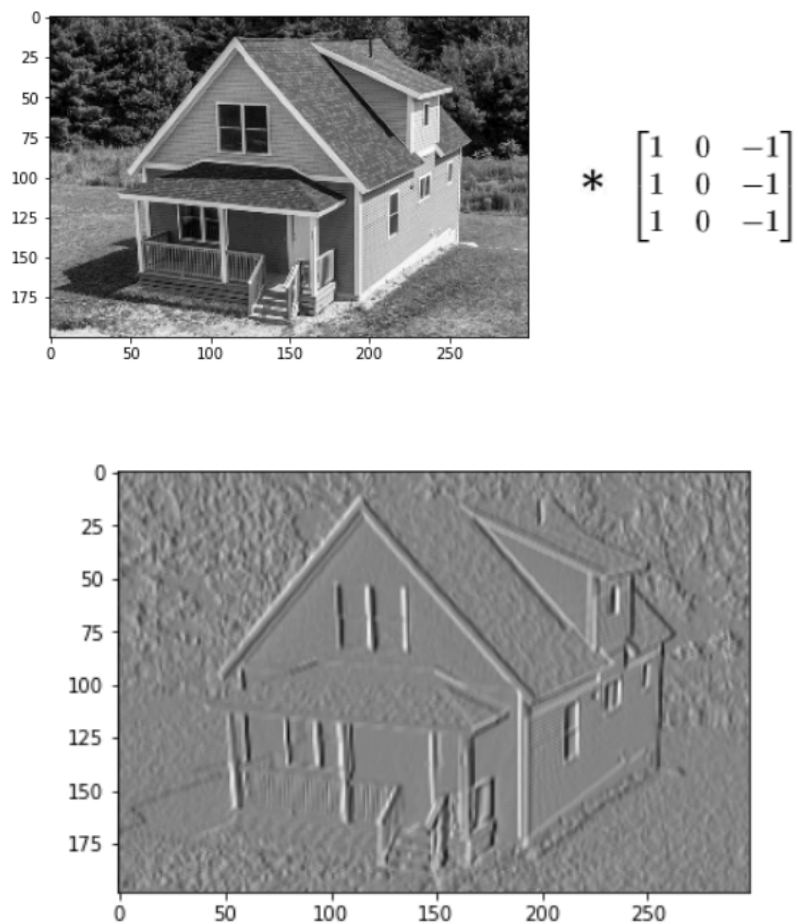


Figura do TCC de Diego Marques, aluno do IME-USP.

Convolution layers - exemplo

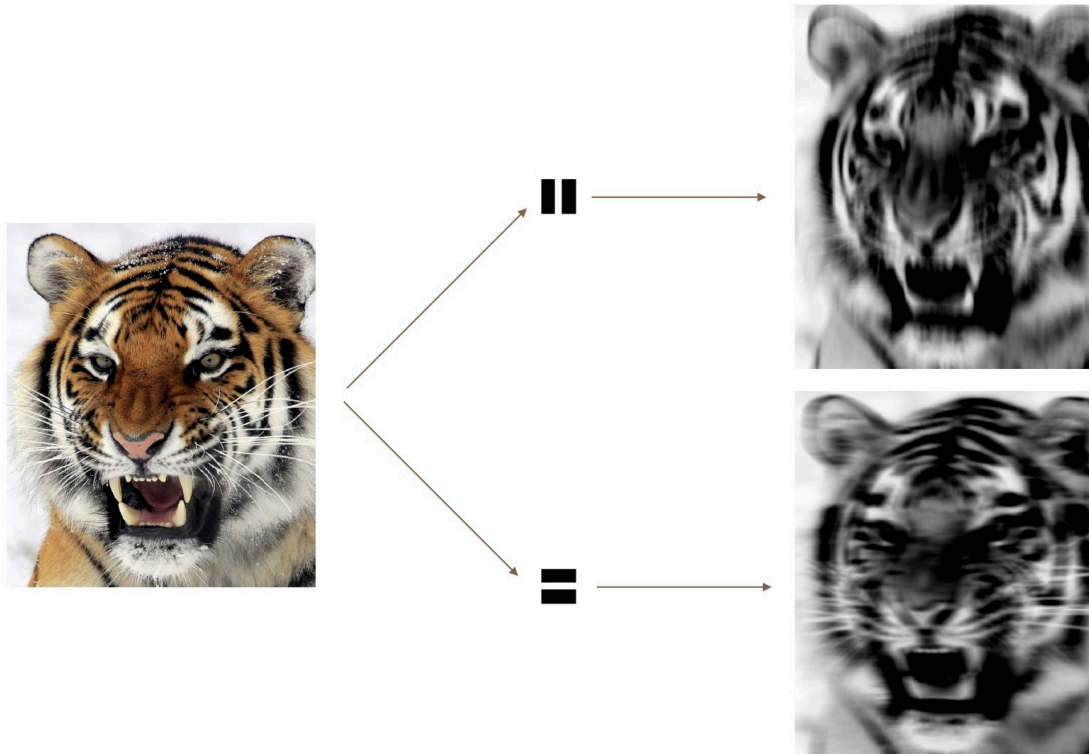
- Se uma submatriz 2×2 da imagem original se assemelha ao filtro de convolução, ela terá um *valor grande* na imagem convolucionada. Caso contrário, terá um *valor pequeno*.
- No exemplo anterior, usamos uma matriz 3×3 , mas em geral, filtros de convolução são matrizes $\ell_1 \times \ell_2$, em que ℓ_1 e ℓ_2 são números inteiros positivos pequenos, não necessariamente iguais.

Padding

Para que a imagem convolucionada mantenha a mesma dimensão da imagem original, deve-se realizar uma operação de *padding*, que consiste em adicionar uma borda de 0's ao redor da imagem de entrada antes de aplicar a operação de convolução.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a & b & c & 0 \\ 0 & d & e & f & 0 \\ 0 & g & h & i & 0 \\ 0 & j & k & l & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Convolution layers - exemplo



Os filtros acima são matrizes 15×15 contendo zeros (preto), com uma faixa estreita de uns (branco), orientada vertical ou horizontalmente.

Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Convolution layers - alguns detalhes

- Lembre-se que uma imagem é representada em três canais (RGB);
- Um filtro de convolução também terá três canais (um para cada cor), com possíveis diferentes valores de peso;
- O resultado das três operações é somado para formar uma saída em duas dimensões;
- Observe que neste momento as informações de cor foram usadas e não são passadas para camadas seguintes.
- Se usarmos K filtros de convolução diferentes na primeira camada oculta, obteremos K saídas bidimensionais (matrizes), que juntas são tratadas como um único objeto tridimensional (array);
- Então, cada uma das K saídas é um canal separado de informação;
- Após a operação de convolução, temos K canais em contraste com os três canais de cores da imagem original.
- Normalmente, aplicamos a função de ativação ReLU após a operação de convolução.

Pooling Layers

- Maneira de condensar uma imagem em uma imagem menor sumarizada.
- Essa operação é especificada, não aprendida pela rede.
- Operações mais comuns: *max pooling* e *average pooling*.
- Uma operação max pooling considerando um bloco de 2×2 reduz o tamanho da imagem pela metade em cada dimensão.

Pooling Layers - exemplo

Max pooling 2×2 :

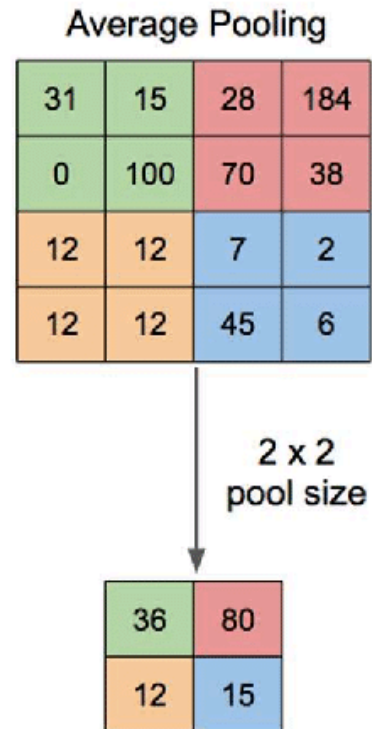
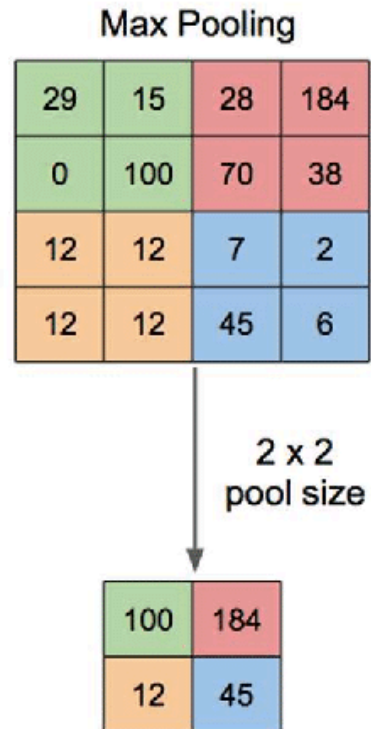


Figura da página [Research Gate](#).

In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change.

Arquitetura de uma CNN

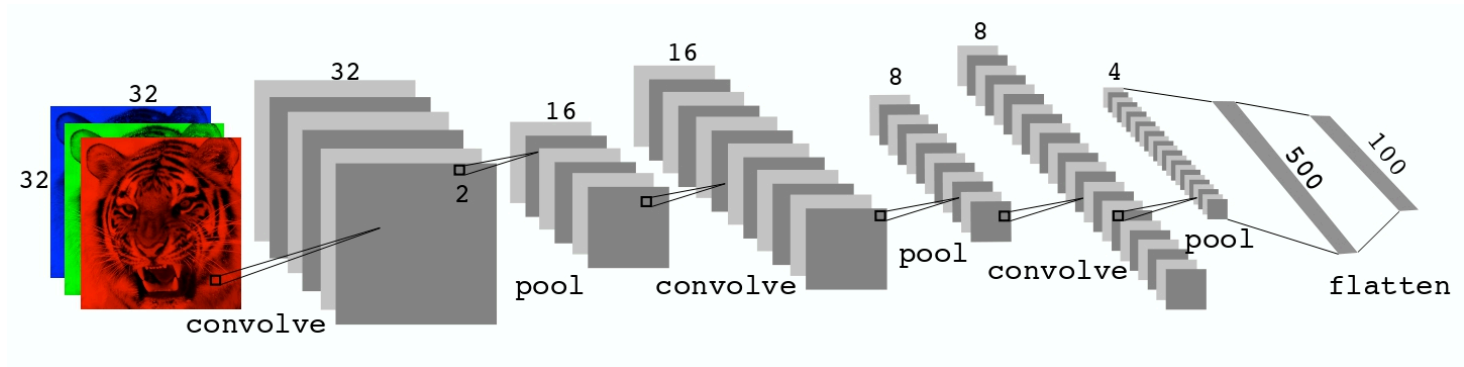


Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Data Augmentation

Cada imagem do conjunto de treino é replicada diversas vezes, cada uma delas aleatoriamente distorcidas de uma maneira natural que não altera o reconhecimento humano.



Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Código R: cifar100.r

Redes neurais para dados sequenciais

Redes neurais recorrentes (RNNs)

Várias fontes de dados tem uma natureza sequencial, como por exemplo:

- documentos textuais, como livros, artigos, reviews de filmes e tweets (a posição relativa e a sequência capturam a narrativa e o tema, que podem ser explorados em tarefas como classificação por tópicos, análise de sentimentos, tradução, etc);
- séries temporais de temperatura, chuva, velocidade do vento, qualidade do ar, etc, com o objetivo de fazer previsões para os próximos dias;
- séries temporais financeiras, para rastrear índices de mercado, volume de negociações, preços de ações, etc. Aqui, a previsão geralmente é difícil.
- discurso gravado, músicas e outros registros sonoros, com o objetivo de fazer transcrição e até tradução do conteúdo;
- texto manuscrito, para fazer digitalização.

Redes neurais recorrentes (RNNs)

- Especificamente desenhadas para dados sequenciais;
- Por exemplo, um documento pode ser representado como uma sequência L de palavras, $X = (X_1, X_2, \dots, X_L)$, em que X_ℓ é uma palavra;
- Portanto, a ordem das palavras é importante;
- A variável resposta Y também pode ser uma sequência (como em uma tradução) ou uma classe (como em análise de sentimentos).

Arquitetura das redes neurais recorrentes

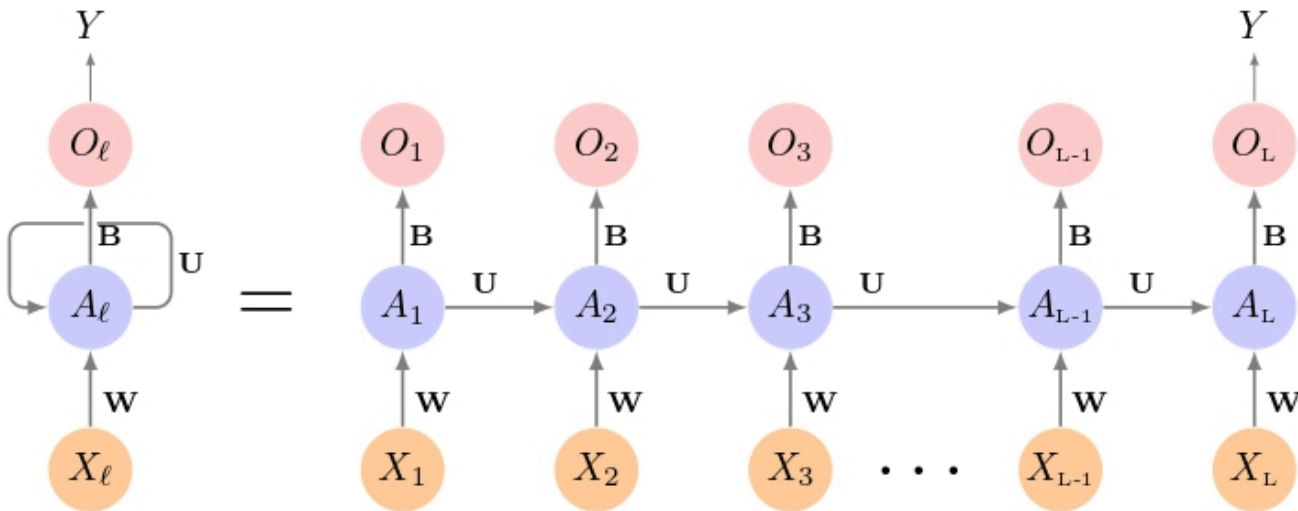


Figura adaptada do livro [An Introduction to Statistical Learning](#) (James et al, 2021).

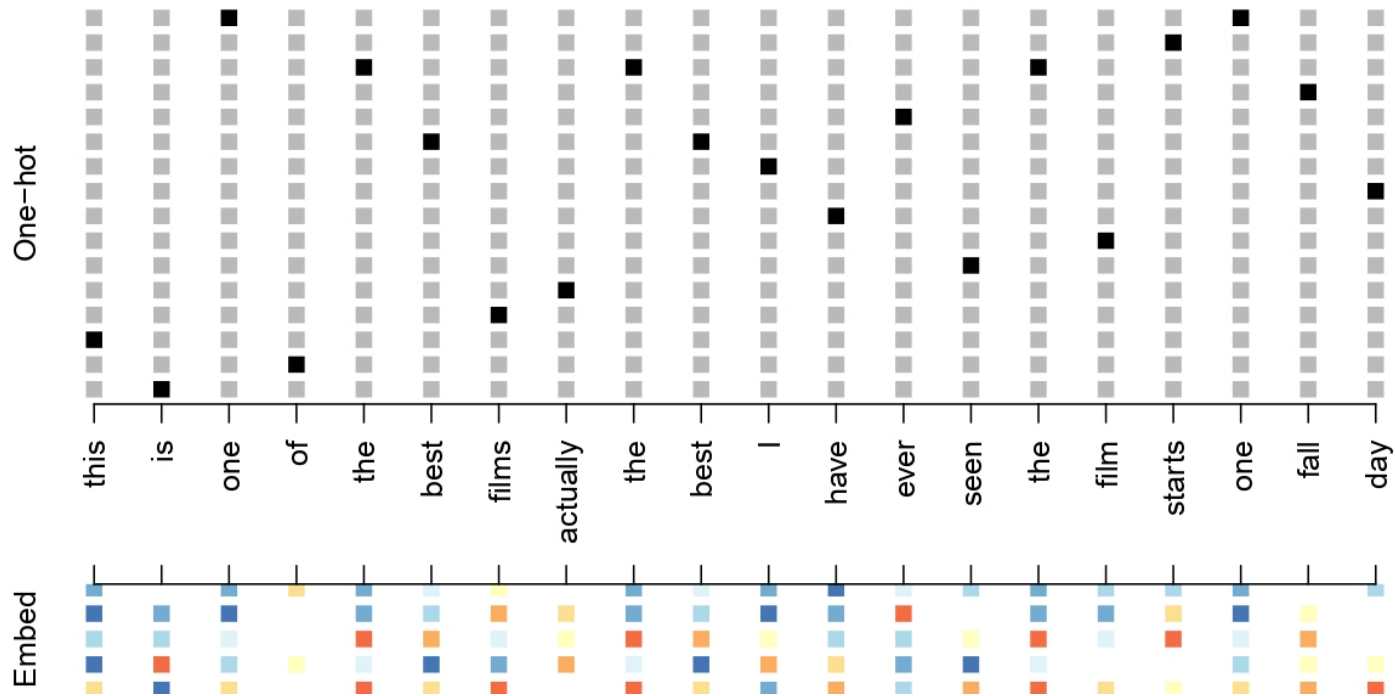
Embeddings de texto

- Representação numérica de palavras ou sequências de texto.
- Transformação de dados textuais em vetores de números densos.
- Facilita o processamento de linguagem natural em algoritmos de aprendizado de máquina.
- Motivação: limitações da representação tradicional de texto (one-hot encoding).
 - Esparsidade e dimensionalidade alta.
 - Dificuldades em capturar semântica e relações entre palavras.

Funcionam de embeddings de texto

- Mapeamento de palavras para vetores contínuos.
- Aprendizado de representações semânticas.
- Preservação de similaridades e relações entre palavras.
- Uso de modelos pré-treinados (ex: Word2Vec, GloVe, FastText).
- Vantagem: Redução do tempo de treinamento e melhoria na performance.

Embeddings - exemplo



Sequência de 20 palavras representadas como um único documento: *one-hot-encoding* com um dicionário de 16 palavras (imagem superior) e *embedding* em um espaço m -dimensional, com $m = 5$.

Figura do livro *An Introduction to Statistical Learning* (James et al, 2021).

Código R: imdb.r

Resumindo

- **Redes Neurais Convolucionais (CNNs):**
 - Excelência em tarefas de visão computacional.
 - Processamento eficiente através de camadas convolucionais e de pooling.
 - Aplicação prática em reconhecimento de padrões e classificação de imagens.
- **Redes Neurais Recorrentes (RNNs):**
 - Ideal para dados sequenciais como texto e séries temporais.
 - Capacidade de manter a informação ao longo do tempo através de loops internos.
 - Utilização em análise de sentimentos, tradução automática e outras tarefas de processamento de linguagem natural.

Obrigado!

`magnotfs@insper.edu.br`