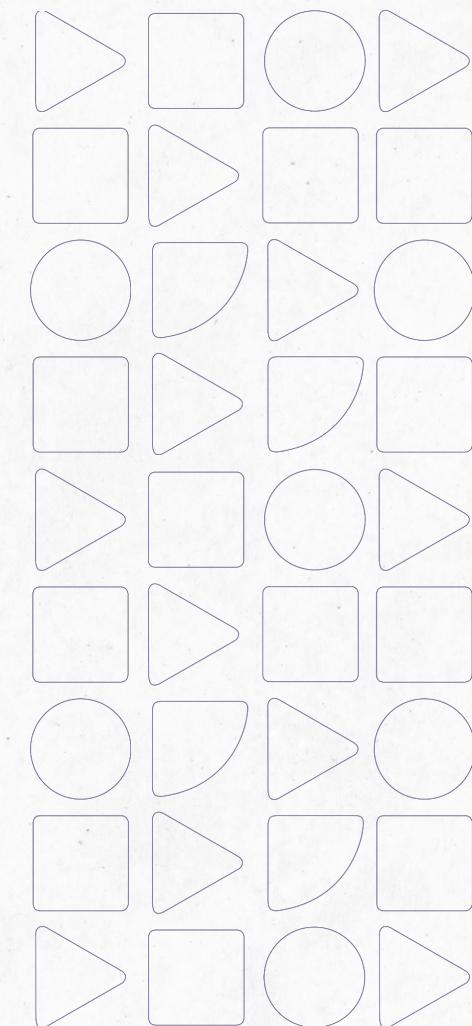


Deep learning

Disciplina: Sistemas Inteligentes



Conteúdos:

Deep learning.

Habilidade(s):

Aplicar os conceito de *deep learning* em diversas situações.

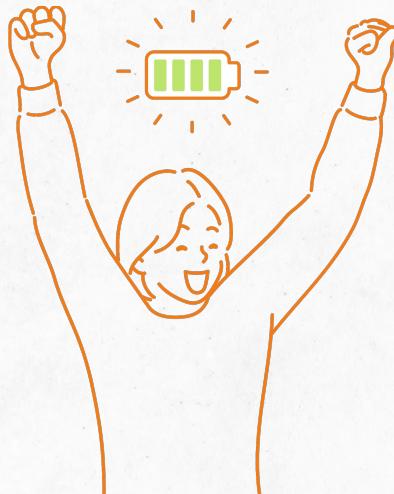
Bloco 1

Redes neurais: matematizando o biológico.

Abertura

Qual é o seu *mood* de hoje?

Com os neurônios funcionando a todo vapor!



Tentando fazer o neurônio que me resta funcionar.





Redes neurais

Redes neurais têm sido estudadas desde a década de 1940 com o *Perceptron*. No entanto, o seu desenvolvimento foi limitado pelo poder computacional da época.

Com o aumento do poder computacional, especialmente com o uso de GPUs, as redes neurais emergiram como uma poderosa ferramenta. Hoje, elas são aplicadas com sucesso em diversos problemas sob o rótulo de *deep learning*.

Estrutura de uma rede neural

Neurônios artificiais

Uma rede neural é composta por uma grande quantidade de neurônios artificiais. Cada neurônio realiza cálculos matemáticos em suas entradas e aplica uma função de ativação para gerar uma saída.

Pesos e funções de ativação

Cada entrada é multiplicada por um peso. As entradas ponderadas são somadas e passam por uma função de ativação. As funções de ativação podem variar, incluindo degrau, sigmoide, tangente hiperbólica e softmax.

Aprendizado em redes neurais

Ajuste de pesos (*backpropagation*): o aprendizado na maioria das redes neurais envolve o ajuste dos pesos das entradas. O processo de *backpropagation* compara a saída da rede com a resposta esperada e ajusta os pesos para minimizar o erro.

Gradiente descendente: o gradiente descendente é um método comum para ajustar os pesos durante o *backpropagation*.

Tipos de arquiteturas de redes neurais

Multilayer perceptrons (MLP)

Usado para tratar problemas não lineares.
Combinação de *perceptrons* em camadas.

Redes convolucionais (CNN)

Amplamente utilizadas em processamento de imagens.
Transformam dados de entrada em matrizes numéricas tridimensionais.

Redes recorrentes (RNN)

Adequadas para dados sequenciais, como texto e áudio. Uso da predição de um dado para prever o próximo (séries temporais).

Long short-term memory (LSTM)

Introdução de unidades LSTM para aprender correlações em séries temporais longas.

Tipos de arquiteturas de redes neurais

Máquinas de Boltzmann

Redes neurais recorrentes com abordagem estatística (estocástica). Usadas para resolver problemas combinatórios complexos.

Deep belief network

Alternativa ao aprendizado supervisionado. Agrupa dados de entrada em classes aprendidas.

Deep auto-encoders

Composta por redes simétricas (*encoder* e *decoder*). Mapeia uma representação em outra.

Generative adversarial network

Rede profunda composta por duas redes adversárias. Usada para aprender a gerar representações, como imagens, texto ou música.

Combinação de arquiteturas de redes neurais

As arquiteturas podem ser combinadas para resolver uma variedade de problemas. A escolha depende do tipo de dados e das metas de predição.

Para iniciantes, é crucial experimentar várias arquiteturas, desde as simples até as complexas. A prática ajuda a escolher a melhor abordagem para problemas do mundo real.

Vamos praticar?

Formem grupos para realizar a atividade.

Primeiro momento

15 minutos

Escolham um dos tipos de arquiteturas de redes neurais, pesquisem uma aplicação desse tipo e criem um *folder* sobre essa aplicação.

Segundo momento

10 minutos

Mostrem para a turma o que vocês produziram.



Bloco 2

O conceito de aprendizagem profunda.

Deep learning

Deep learning (DL) é uma forma de *machine learning* (ML) que utiliza redes multicamadas complexas para extrair e modelar recursos de dados. Ela é capaz de resolver problemas complexos, como visão computacional e processamento de linguagem natural (PNL).



Deep learning

VANTAGENS

- DL pode lidar com problemas onde outros métodos falham;
- Permite modelos mais precisos e reduz o esforço de engenharia de recursos.

VS

DESVANTAGENS

- Treinar e otimizar modelos de DL pode exigir considerável poder computacional e dados;
- Os resultados dos modelos de DL podem ser difíceis de interpretar.

Diferenças entre DL e ML tradicional

Deep learning

Em DL, ocorre a extração automática de recursos nas camadas de rede, resultando em uma hierarquia de aprendizagem de representação.

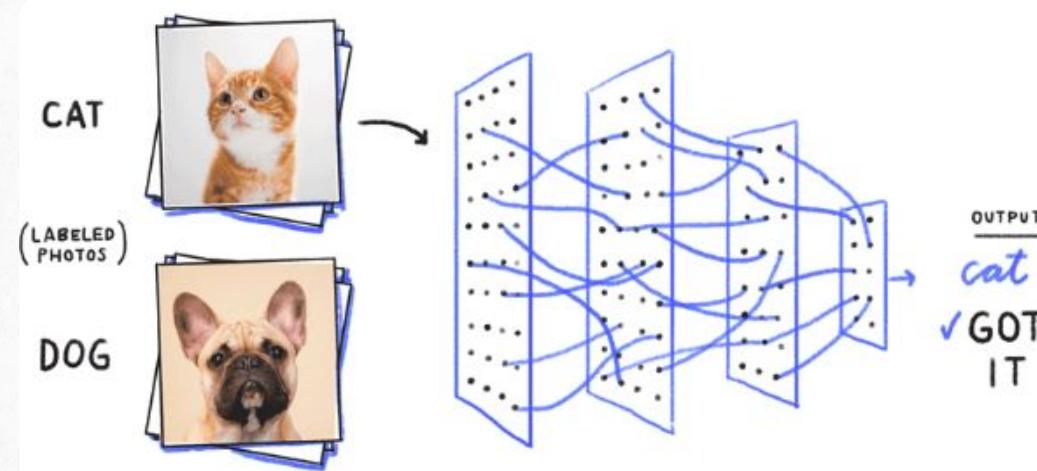
VS

Machine learning

Em ML tradicional, os modelos aprendem a classificar ou prever com recursos pré-selecionados.

Estruturas de *deep learning*

Uma característica essencial da DL é ter múltiplas camadas de processamento com pesos entre entrada e saída. As camadas iniciais aprendem ou extraem recursos, enquanto camadas posteriores fazem a classificação ou previsão.



Aplicações de *deep learning*

A *deep learning* tem alcançado resultados de ponta em reconhecimento de imagem, áudio, PNL e tradução de idiomas. Também é aplicada em jogos automáticos e outros campos.

Exemplos de redes DL

Existem redes DL especializadas para diferentes fontes de dados, incluindo imagens, áudio, dados semi-estruturados e dados estruturados.

DNNs (Redes Neurais Profundas)

Extensão da arquitetura clássica de MLP (*multilayer perceptrons*).

CNNs (Redes Neurais Convolucionais)

Especializadas em aprendizado de recursos de imagem.

TNNs (Redes Neurais de Transformador)

Utilizadas em aprendizado de sequência, especialmente em tarefas de tradução de idiomas e processamento de linguagem natural (PNL).

Vamos praticar?

Primeiro momento

5 minutos

Escolha um dos tipos de redes DL e pesquise uma aplicação do tipo escolhido. Em seguida, leia um pouco sobre ela.

Segundo momento

20 minutos

Fale para a turma sobre o que você pesquisou.

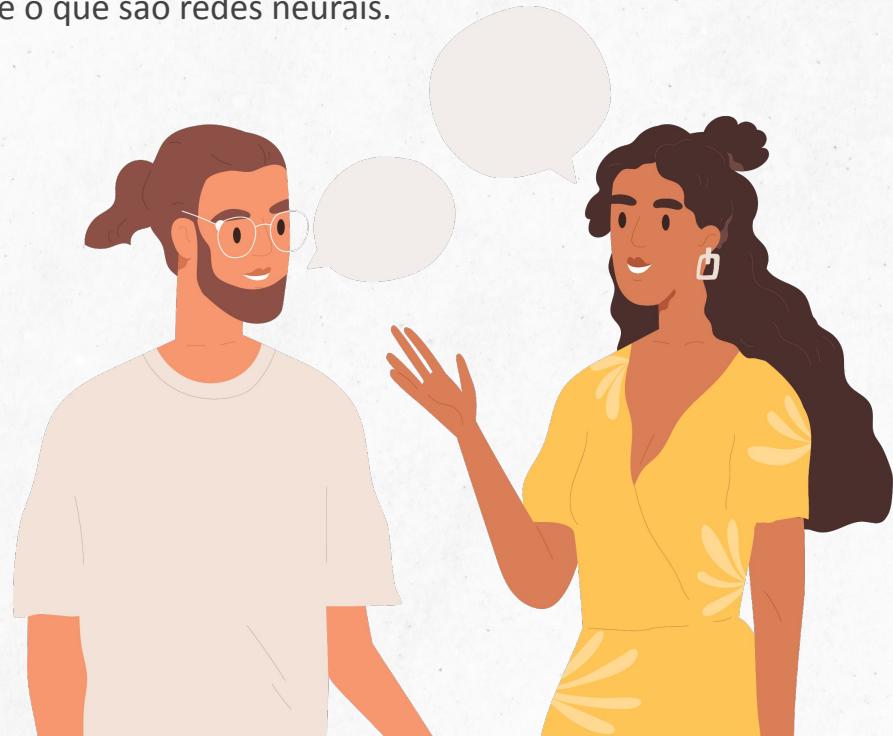


Bloco 3

Redes neurais profundas de *feedforward*.

Fofoca do bem

Conte para a turma, em forma de fofoca, um pouco sobre o que são redes neurais.



Redes neurais profundas (DNNs)

Em DNNs (*deep neural networks*) *feedforward*, os neurônios são as unidades básicas de processamento, organizados em camadas separadas: entrada, camadas ocultas e saída.

Fluxo de informação em DNNs

Propagação direta e função de ativação

As saídas de cada camada passam pela propagação direta e aplicam uma função de ativação. A função de ativação introduz não linearidade na rede.

Equação de propagação direta

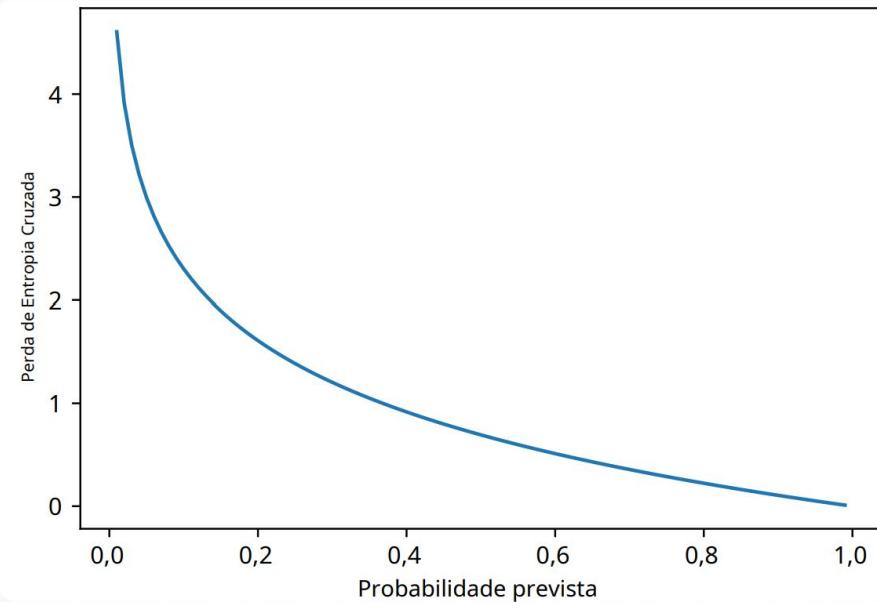
Mostra a operação de propagação direta seguida pela aplicação da função de ativação. Matrizes Zo, UMA e Ceu são elementos-chave no processo.

Função de ativação softmax

O aprendizado supervisionado em DNNs segue etapas semelhantes ao ML tradicional e utiliza uma função de perda para comparar a saída da rede com a saída desejada. A função de perda gera um vetor de erro, usado para corrigir os pesos da rede.

Função de perda de entropia cruzada

A função de perda de entropia cruzada é uma ferramenta fundamental em ML e otimização. Ela define a perda considerando o número de classes (K), a probabilidade verdadeira e a distribuição prevista pelo modelo (softmax).



Vamos praticar?

Formem quatro grupos para realizar a atividade.

Atividade

20 minutos

Dois grupos vão ficar com a função de ativação softmax e os outros dois com a função de perda de entropia cruzada. Cada grupo deve criar um *folder* explicando um pouco e mostrando exemplos de aplicações dessas funções.



Bloco 4

Redes neurais profundas de *feedforward*.

Algoritmo de retropropagação em redes neurais

O algoritmo de retropropagação é fundamental no treinamento de redes neurais. É usado para calcular correções nos pesos da rede para minimizar o erro.

Já o **vetor gradiente** aponta para a direção de subida mais íngreme. É usado por algoritmos de otimização para ajustar os pesos e reduzir o erro.

Algoritmo de otimização Adam

O algoritmo

O algoritmo de otimização Adam é usado para controlar as correções dos pesos, adaptar as taxas de aprendizado e evitar mínimos locais.

Adequação a dados ruidosos

É eficaz em problemas com dados ruidosos ou esparsos, como aqueles com muitos recursos binários e *one-hot encoding*.



Mitigação do *overfitting*

Overfitting é um problema comum em modelos de aprendizado de máquina e redes neurais, que ocorre quando o modelo se ajusta demasiadamente aos dados de treinamento, mas não generaliza bem. Existem formas de mitigar o *overfitting*, como:

Aumento do conjunto de dados

Uma abordagem possível é aumentar o tamanho e a variabilidade do conjunto de dados.

Redução da complexidade do modelo

Diminuir a complexidade do modelo, incluindo camadas e parâmetros.

Dropout

O *dropout* é uma técnica que aleatoriamente zera valores de ativações para evitar co-adaptações complexas nos dados de treinamento.

Normalização em lote

A normalização em lote ajuda a evitar deslocamentos nos valores de ativação, contribuindo para estabilidade de treinamento.

Vamos praticar?

Primeiro momento

10 minutos

Escolha uma das formas de mitigar o *overfitting*, pesquise uma situação em que seja possível aplicar a mitigação que você escolheu. Em seguida, leia um pouco sobre ela.

Segundo momento

15 minutos

Fale para a turma sobre o que você pesquisou.



Estratégias para combater o *overfitting*

Dropout e normalização em lote são estratégias comuns para mitigar o *overfitting* em redes neurais.

***Underfitting* e viés alto**

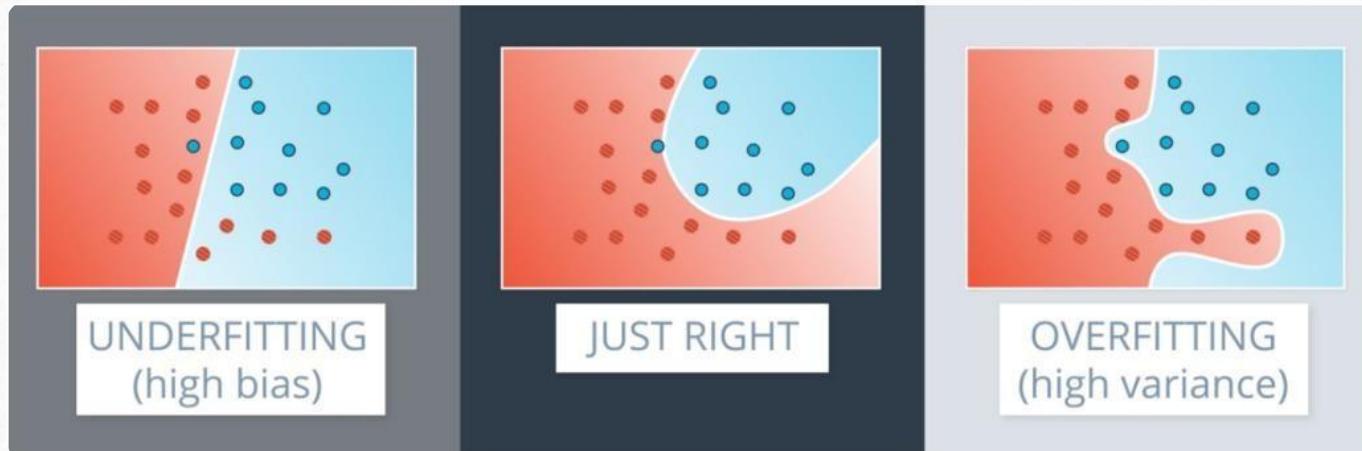
Underfitting

Underfitting é menos comum, indicando que o modelo não generaliza bem devido à sua baixa complexidade.

Limites de decisão simples

Modelos com *underfitting* geram limites de decisão simples, incapazes de separar corretamente os dados em classes.

Underfitting e viés alto



Bloco 5

Redes neurais convolucionais.

Redes neurais convolucionais (CNN)

As CNNs foram inspiradas na organização do córtex visual humano, com semelhança na conectividade dos neurônios.

Os neurônios individuais respondem a estímulos em áreas limitadas da visão, chamadas campos receptivos.

Operação de convolução em CNNs

Capturando dependências espaciais

As CNNs usam a operação de convolução para capturar dependências espaciais em imagens. É feita a aplicação de filtros ou *kernels* relevantes.

Aprendizado de filtros

Diferentemente de técnicas tradicionais, CNNs com treinamento suficiente aprendem esses filtros automaticamente.

Benefícios das CNNs

CNNs exigem menos pré-processamento em comparação com outras redes de aprendizado profundo. Também possuem melhor ajuste aos dados de imagem devido à redução de parâmetros e compartilhamento de peso.

Camadas

Camadas convolucionais

- CNNs podem ter várias camadas convolucionais;
- A primeira camada captura recursos de baixo nível, como bordas e cores;
- Camadas subsequentes capturam recursos de nível superior, como composições dos recursos de nível inferior.

Camadas de *pooling*

- A camada de *pooling* reduz o tamanho dos recursos, economizando poder computacional;
- Também extrai características dominantes invariantes à posição.

Operações de *pooling*

Max pooling

Retorna o valor máximo da parte da imagem coberta pelo filtro.

Agrupamento médio

Retorna a média de todos os valores na parte da imagem coberta pelo filtro.

Arquitetura da rede neural convolucional

O número de camadas, organização e ordenação definem a arquitetura da rede.

Após as operações de convolução e *pooling*, os recursos são achatados para classificação ou regressão.

Discussões e reflexões



Bloco 6

Redes neurais transformadoras.

Redes neurais transformadoras (TNN)

TNN é um modelo de *deep learning* que utiliza o mecanismo de atenção e que é projetado para lidar com dados sequenciais em NLP, como resumo e tradução de texto.

Esse modelo tem substituído gradualmente as RNNs em problemas de PNL, por conta de sua aprimoração e uso de grandes conjuntos de dados.

Temos como exemplos o desenvolvimento de modelos pré-treinados, como BERT e GPT.

Diferenças entre TNN e RNN

Processamento não sequencial: ao contrário das redes neurais recorrentes (RNN), a TNN não requer processamento sequencial. Utiliza-se o mecanismo de atenção para identificar significado semântico em qualquer parte da sequência.

Paralelização eficiente: a TNN permite maior paralelização, reduzindo os tempos de treinamento.

Operações nas camadas de TNN

Camadas do codificador

Camadas de codificação geram codificações com informações contextuais. As codificações são passadas para a próxima camada.

Camadas do decodificador

Camadas de decodificação usam informações contextuais das codificações para gerar a sequência de saída. Incluem mecanismo de atenção, rede neural *feed-forward*, conexões residuais e normalização.

Arquitetura de TNN

TNNs seguem uma arquitetura codificador-decodificador.

O codificador processa a entrada iterativamente, gerando informações relevantes.

Codificador



Decodificador

Aprendizado profundo multimodal

- Modalidades são canais de informação;
- Combinação de diferentes fontes de informação para melhorar a capacidade de previsão;
- Há algumas modalidades que são únicas, utilizadas no aprendizado profundo, como o áudio, imagens e texto;
- Coleta de informações de diferentes fontes para aprimorar previsões.

Estratégias de combinação de informações

Camada de fusão inicial

Une recursos antes do processamento pela rede. Facilita a combinação de informações.

Camada de fusão intermediária

Aprendizado de características de uma única fonte de informação. Constrói modelos adequados aos tipos de dados.

Camada de fusão tardia

Fusão das representações de recursos de alto nível de cada modalidade. Utiliza-se uma camada de representação compartilhada para previsões.

Vamos praticar?

Primeiro momento

10 minutos

Escolha uma das três camadas das estratégias de combinação de informações, pesquise um pouco mais sobre como utilizá-la e se existe alguma aplicação que faz uso dela. Em seguida, leia um pouco sobre.

Segundo momento

15 minutos

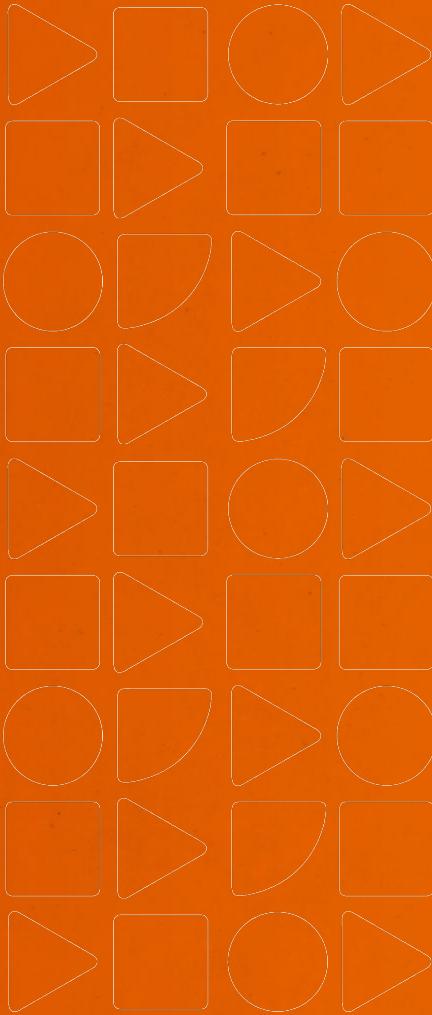
Fale para a turma sobre o que você pesquisou.



Fechamento

Diga uma palavra sobre o que você achou da aula.





Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Sistemas Inteligentes*. 2023.