

Treino e Teste de dados de uma rede neural

Definição: Divisão dos dados em dois conjuntos:

- Treino: Usado para ajustar os parâmetros do modelo.
- Teste: Usado para avaliar o desempenho final.

Exemplo:

Um modelo que prevê preços de casas:

- Treino: 70% dos dados (ex: 700 imóveis).
- Teste: 30% (300 imóveis) para verificar se o modelo generaliza.

Aplicação: Evita *overfitting* (ex: um modelo que "decora" os preços do treino mas falha em novos dados).

Relação: Conecta-se com Validação Cruzada (técnica mais robusta que usa múltiplas divisões).

Acurácia vs. Precisão vs. Recall

Definição:

- Acurácia: % de acertos totais.
- Precisão: % de positivos previstos que são reais (ex: emails marcados como spam corretamente).
- Recall: % de positivos reais detectados (ex: cânceres identificados dentre todos os casos).

Exemplo: Diagnóstico médico (100 pacientes):

- Acurácia: 90% (90 diagnósticos corretos).
- Precisão: 85% (dos 20 casos previstos como câncer, 17 eram verdadeiros).
- Recall: 75% (detectou 15 dos 20 casos reais de câncer).

Aplicação: Recall é crítico em medicina; Precisão em filtros de spam.

Relação: Métricas derivadas da Matriz de Confusão.

Overfitting Definição:

Quando o modelo se ajusta demais aos dados de treino, capturando ruídos em vez de padrões.

Exemplo: Um modelo de regressão polinomial que traça uma curva passando por todos os pontos de treino (alta complexidade), mas falha em novos dados.

Aplicação: Soluções incluem Regularização (L1/L2) e Validação Cruzada.

Relação: Oposto de Underfitting (modelo muito simples, como uma reta subajustada).

K-Means Clustering

Definição: Algoritmo não supervisionado que agrupa dados em k clusters baseado em similaridade.

Exemplo: Agrupamento de clientes de e-commerce por comportamento de compra:

- Cluster 1: Compradores frequentes de eletrônicos.
- Cluster 2: Compradores ocasionais de roupas.

Aplicação: Segmentação de mercado, análise de imagens (agrupar pixels similares).

Relação: Técnica de Aprendizado Não Supervisionado.

Redes Neurais Convolucionais (CNN)

Definição: Redes projetadas para processar dados grid-like (ex: imagens), usando filtros que detectam padrões hierárquicos (bordas → texturas → objetos).

Exemplo: Filtro de uma CNN detectando bordas verticais em uma imagem de um gato:

1. Camada 1: Bordas.
2. Camada 2: Olhos, nariz.
3. Camada 3: Rosto completo.

Aplicação: Reconhecimento facial, autônomo (detecção de pedestres).

Relação: Base para GANs (geração de imagens).

Ética: Bias em IA

Definição: Viés sistêmico em modelos que perpetuam discriminações (ex: gênero, raça).

Exemplo: Modelo de contratação que prioriza currículos masculinos para TI, pois foi treinado com dados históricos enviesados.

Aplicação: Auditoria de modelos com SHAP para identificar *features* discriminatórias.

Relação: Conecta-se com Privacidade Diferencial (proteção de dados sensíveis).

IA Generativa: Stable Diffusion

Definição: Modelo que gera imagens a partir de texto, usando difusão (adição e remoção iterativa de ruído).

Exemplo: Prompt: "Um gato astronauta no estilo Van Gogh":

1. Gera ruído.
2. Refina passo a passo até a imagem final.

Aplicação: Design de jogos, publicidade.

Relação: Usa Transformers para interpretar texto.

Validação Cruzada (k-fold)

Definição: Técnica que divide os dados em k partes iguais, usando $k-1$ para treino e 1 para teste, repetindo k vezes.

Exemplo:

Para $k=5$ (5-fold):

1. Dados divididos em 5 partes (A, B, C, D, E).
2. Rodadas:
 - Treino: A+B+C+D | Teste: E
 - Treino: A+B+C+E | Teste: D
 - ...

Média das métricas finais.

Aplicação: Evita viés em conjuntos pequenos (ex: dados médicos raros).

Relação: Alternativa superior à divisão simples treino/teste.

Regularização (L1 e L2)

Definição: Técnicas para penalizar pesos do modelo e evitar overfitting:

- L1 (LASSO): Penaliza valores absolutos → cria esparsidade (zeros).
- L2 (Ridge): Penaliza valores quadrados → reduz magnitudes.

Exemplo: Prever preços de casas:

- LASSO pode zerar pesos de *features* irrelevantes (ex: "cor da porta").

- Ridge reduz impacto de *features* correlacionadas (ex: "área útil" vs. "área total").

Aplicação: LASSO para seleção de features; Ridge para multicolinearidade.

Relação: Usada com Regressão Linear ou Redes Neurais.

Aprendizado por Reforço

Definição: Modelo aprende através de recompensas/punições em um ambiente.

Exemplo: Agente em um labirinto:

- Recompensa (+1): Chegar ao fim.
- Punição (-1): Bater na parede.

Aplicação: Jogos (AlphaGo), robótica (caminhar sem cair).

Relação: Base para Aprendizado por Reforço Profundo (uso de redes neurais).

Random Forest

Definição: Conjunto de *árvores de decisão*, onde cada árvore vota na previsão final.

Exemplo: Diagnóstico de doenças:

- Árvore 1: Analisa sintomas.
- Árvore 2: Analisa histórico familiar.
- ...

Resultado final = maioria dos votos.

Aplicação: Sistemas de recomendação (ex: Netflix).

Relação: Melhora Árvores de Decisão únicas (reduz overfitting).

Transformers

Definição: Arquitetura que usa *mecanismo de atenção* para processar sequências (ex: texto).

Exemplo: Tradução "Hello → Olá":

1. Atenção identifica "Hello" como palavra-chave.
2. Gera "Olá" com contexto completo da frase.

Aplicação: ChatGPT, tradutores (Google Translate).

Relação: Base para GPT e BERT.

Privacidade Diferencial

Definição: Adiciona ruído controlado aos dados para preservar anonimato.

Exemplo: Censo demográfico:

- Idade real: 35 anos → Idade reportada: 33 ou 37 (aleatório).

Aplicação: Proteção de dados em saúde (ex: registros de HIV).

Relação: Defesa contra Adversarial Attacks.

Redes Neurais Recorrentes (RNN)

Definição: Redes com "memória" para processar sequências (ex: séries temporais).

Exemplo: Previsão do próximo caractere em "Hello":

- Input "H" → Prevê "e"
- Input "e" → Prevê "l"
- ...

Aplicação: Previsão de ações na bolsa.

Relação: Evoluiu para LSTM (resolve problema do vanishing gradient).

Explicabilidade: SHAP

Definição: Método baseado em teoria dos jogos para atribuir importância a features.

Exemplo: Modelo de crédito nega empréstimo:

- SHAP mostra que "baixa renda" contribuiu com -30 pontos.

Aplicação: Bancos (explicações regulatórias).

Relação: Alternativa ao LIME (menos preciso mas mais rápido).

Gradient Descent

Definição: Algoritmo que ajusta pesos do modelo minimizando a função de custo.

Exemplo: Encontrar o ponto mais baixo em um vale (custo = altura):

1. Calcula gradiente (inclinação).
2. Anda na direção oposta.

Aplicação: Treino de redes neurais.

Relação: Usa Backpropagation para calcular gradientes.

1. Aprendizado Supervisionado vs. Não Supervisionado vs. Reforço

Critério	Supervisionado	Não Supervisionado	Reforço
Dados	Rotulados (ex: imagens com classe)	Não rotulados (ex: transações)	Ambiente com recompensas/punições
Objetivo	Prever rótulos	Descobrir padrões ou grupos	Aprender políticas ótimas
Exemplo	Classificar spam	Clusterizar clientes (K-Means)	Robô que aprende a andar
Quando usar	Quando há dados rotulados	Para exploração de dados	Problemas sequenciais (ex: jogos)

2. Classificação: Regressão Logística vs. Random Forest vs. SVM

Critério	Regressão Logística	Random Forest	SVM
Funcionamento	Modelo linear (função sigmoide)	Conjunto de árvores de decisão	Encontra hiperplano de margem máxima
Vantagens	Interpretável, rápido	Alta precisão, evita overfitting	Eficaz em alta dimensionalidade
Limitações	Só captura relações lineares	Menos interpretável	Sensível a kernels mal escolhidos
Aplicação típica	Risco de crédito	Diagnóstico médico	Reconhecimento de escrita

3. Redes Neurais: CNN vs. RNN vs. Transformers

Critério	CNN	RNN	Transformers
Arquitetura	Filtros convolucionais	Loops com memória temporal	Mecanismo de atenção
Melhor para	Dados grid-like (imagens)	Sequências curtas (ex: tempo)	Sequências longas (ex: texto)

Problemas	Invariância a posição	Vanishing gradient	Custo computacional alto
Exemplo	Detecção de tumores em raio-X	Previsão do próximo caractere	Tradução automática (GPT)

4. Métricas: Acurácia vs. Precisão vs. Recall

Critério	Acurácia	Precisão	Recall
Fórmula	$(VP + VN) / \text{Total}$	$VP / (VP + FP)$	$VP / (VP + FN)$
Foco	Acertos globais	Erros de falsos positivos	Erros de falsos negativos
Quando priorizar	Classes balanceadas	Custo de FP alto (ex: spam)	Custo de FN alto (ex: câncer)
Exemplo	90% acertos em teste	85% dos spam detectados são reais	75% dos casos de câncer identificados

5. Técnicas de Regularização: L1 (LASSO) vs. L2 (Ridge)

Critério	L1 (LASSO)	L2 (Ridge)
Efeito nos pesos	Zera pesos irrelevantes	Reduz magnitudes dos pesos
Uso ideal	Seleção de features	Evitar overfitting em features correlacionadas
Exemplo	Remove "cor da porta" do modelo	Reduz impacto de "área útil" e "área total"
Matemática	Penaliza $ w $	Penaliza w^2

6. IA Generativa: GAN vs. Autoencoder vs. Stable Diffusion

Critério	GAN	Autoencoder	Stable Diffusion
Funcionamento	Gerador vs. Discriminador	Encoder + Decoder	Difusão gradual de ruído
Saída	Imagens realistas	Reconstrução de dados	Imagens a partir de texto

Aplicação	Deepfakes	Detecção de anomalias	Arte digital
Dados necessários	Grandes datasets	Dados de treino específicos	Texto + imagens

7. Overfitting vs. Underfitting

Critério	Overfitting	Underfitting
Causa	Modelo muito complexo	Modelo muito simples
Sintoma	Alto desempenho no treino, baixo no teste	Desempenho ruim em ambos
Solução	Regularização, Dropout	Aumentar complexidade do modelo
Analogia	Decorar um livro inteiro para uma prova	Estudar só o título do livro

8. Pré-processamento: One-Hot vs. Embedding

Critério	One-Hot Encoding	Embedding
Uso	Categóricas com poucas classes	Categóricas com muitas classes (ex: palavras)
Dimensionalidade	Alta (1 coluna por categoria)	Baixa (vetores densos)
Exemplo	Cor: [1,0,0] = Vermelho	Palavra "cão" → [0.2, -0.5, 0.1]
Vantagem	Simplicidade	Captura relações semânticas

9. Redes Neurais: LSTM vs. GRU

Critério	LSTM	GRU
Arquitetura	Portas: Input, Forget, Output	Portas: Reset, Update
Complexidade	Mais parâmetros (3 portas)	Mais simples (2 portas)

Memória Longa	Excelente (porta <i>Forget</i> explícita)	Boa, mas menos eficaz em sequências muito longas
Velocidade	Mais lenta	Mais rápida
Aplicação Típica	Previsão de séries temporais complexas	Chatbots, tradução automática

Quando usar?

- LSTM: Quando a relação temporal é crítica (ex: previsão de ações meses à frente).
- GRU: Quando eficiência computacional é prioritária (ex: processamento em tempo real).

10. Redução de Dimensionalidade: PCA vs. t-SNE

Critério	PCA	t-SNE
Objetivo	Maximizar variância	Preservar estruturas locais
Natureza	Linear	Não-linear

Saída	Rotas globais	Clusters visíveis
Uso Comum	Pré-processamento para modelos	Visualização de dados
Exemplo	Reduzir 100 features para 10	Plotar grupos de células cancerígenas em 2D

Quando usar?

- PCA: Antes de treinar um modelo (ex: regressão linear).
- t-SNE: Para explorar dados (ex: entender agrupamentos naturais).

11. NLP: Word2Vec vs. BERT

Critério	Word2Vec	BERT
Tipo de Modelo	Pré-treinamento não supervisionado	Pré-treinamento supervisionado com atenção
Contexto	Palavra isolada	Palavra no contexto da frase

Saída	Vetores estáticos	Vetores dinâmicos
Exemplo	"Rei" - "Homem" + "Mulher" ≈ "Rainha"	"Banco" (financeiro vs. assento) tem vetores diferentes
Aplicação	Chatbots simples	Tradução, Q&A complexo

Quando usar?

- Word2Vec: Projetos com recursos limitados ou vocabulário fixo.
- BERT: Tarefas que exigem compreensão contextual (ex: análise de sentimentos em reviews).

12. Visão Computacional: YOLO vs. Mask R-CNN

Critério	YOLO	Mask R-CNN
Abordagem	Detecção em tempo real	Detecção + segmentação
Velocidade	Extremamente rápida	Mais lenta
Precisão	Boa para objetos grandes	Alta (segmenta pixel a pixel)

Uso Típico	Vídeos (ex: carros autônomos)	Medicina (ex: segmentar tumores)
Exemplo	Identificar pedestres em uma rua	Delimitar células em microscopia
Quando usar?		
<ul style="list-style-type: none"> YOLO: Aplicações em tempo real (ex: vigilância). Mask R-CNN: Quando precisão espacial é crítica (ex: robótica cirúrgica). 		

13. Otimização: SGD vs. Adam

Critério	SGD	Adam
Mecanismo	Gradiente puro	Média móvel de gradientes
Velocidade	Lento (taxa de aprendizado fixa)	Rápido (ajuste automático de LR)
Robustez	Sensível a hiperparâmetros	Mais estável

Melhor para	Convex problems	Redes neurais profundas
Exemplo	Regressão linear simples	Treinar uma ResNet
Quando usar?		
<ul style="list-style-type: none"> • SGD: Problemas convexos ou quando interpretabilidade é necessária. • Adam: Deep Learning com grandes datasets. 		

14. Explicabilidade: LIME vs. SHAP

Critério	LIME	SHAP
Abrangência	Local (explica uma previsão)	Global + Local
Base Teórica	Perturbação de dados	Teoria dos jogos (Shapley values)
Performance	Rápido	Computacionalmente custoso
Exemplo	Por que um empréstimo foi negado?	Impacto médio de "idade" em todas as previsões

Uso	Debugging rápido	Relatórios regulatórios
Quando usar?		
<ul style="list-style-type: none"> • LIME: Explicações pontuais para stakeholders não técnicos. • SHAP: Análise detalhada de viés do modelo. 		

15. Aprendizado Federado vs. Aprendizado por Reforço

Critério	Aprendizado Federado	Aprendizado por Reforço
Dados	Dados distribuídos (ex: celulares)	Ambiente simulado/real
Objetivo	Treinar modelo sem centralizar dados	Aprender políticas ótimas
Privacidade	Alta (dados não compartilhados)	Variável
Exemplo	Teclado de smartphone que aprende localmente	Robô que aprende a caminhar
Desafio	Comunicação eficiente	Definir recompensas adequadas

Quando usar?

- Federado: Saúde (dados sensíveis) ou IoT.
- Reforço: Automação industrial ou games.

16. GANs: DCGAN vs. StyleGAN

Critério	DCGAN	StyleGAN
Arquitetura	CNN básica	Controle fino de estilo/camadas
Qualidade	Boa para low-resolution	Alta resolução (faces realistas)
Controle	Limitado	Ajuste de atributos (cabelo, idade)
Exemplo	Gerar rostos 64x64 pixels	Criar avatares hiper-realistas
Complexidade	Mais simples	Requer grande poder computacional

Quando usar?

- DCGAN: Prototipagem rápida ou datasets pequenos.

- StyleGAN: Projetos profissionais (ex: arte digital).

17. Batch Normalization vs. Dropout

Critério	Batch Normalization (BN)	Dropout
Objetivo	Normalizar as ativações entre camadas	Evitar overfitting desligando neurônios aleatoriamente
Como Funciona	Padroniza médias/variâncias das saídas por <i>batch</i>	Zera aleatoriamente neurônios durante o treino
Fase de Aplicação	Durante o treino e inferência	Apenas durante o treino
Vantagem	Acelera convergência e estabiliza treino	Reduz dependências entre neurônios
Desvantagem	Sensível ao tamanho do <i>batch</i>	Pode exigir mais épocas para convergir
Hiperparâmetro	Momentum (para média móvel)	Taxa de dropout (ex: 0.2 = 20% dos neurônios desligados)

Uso Típico	Redes profundas (ex: ResNet)	Redes densas (MLPs) ou CNNs
Exemplo	Normaliza saídas de uma camada convolucional	Desliga 30% dos neurônios em uma camada oculta

Quando Usar?

- Batch Norm:
 - Modelos profundos com problemas de *vanishing/exploding gradients*.
 - Quando o treino é instável ou lento.
 - Exemplo: CNNs para classificação de imagens.
 - Dropout:
 - Redes com muitas camadas densas (MLPs) ou overfitting evidente.
 - Exemplo: Classificação de texto com embeddings.
-

Relação e Combinação

- Podem ser usados juntos (ex: CNNs com BN + Dropout nas camadas finais).
 - BN lida com a escala dos dados; Dropout impõe robustez.
-

Analogia

- BN: Como um professor que padroniza as notas da turma para facilitar comparações.
- Dropout: Como estudar com um grupo onde cada dia um membro diferente falta, forçando todos a aprenderem de forma independente.