



UNIVERSIDADE FEDERAL DO MARANHÃO
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO

Prova ____

Curso: Engenharia da Computação	Ano / Semestre:	Data:
Disciplina: Fundamentos em Redes Neurais	Professor: Thales Levi Azevedo Valente	
Aluno:	Código:	

OBS: RESPOSTAS DEVEM SER ENTREGUES À CANETA AZUL OU PRETA. CASO O ALUNO USE LÁPIS, O MESMO NÃO TERÁ DIREITO A REVISÃO DAS QUESTÕES

Descrição do Cenário

Você foi contratado como engenheiro de machine learning por uma empresa líder no setor de telecomunicações, especializada na operação de redes de fibra óptica transmitidas por dutos subaquáticos. Esses dutos possuem uma certa flexibilidade e são vitais para a infraestrutura de comunicação global. No entanto, estão sujeitos a desafios únicos, como deformações oriundas das forças do ambiente marinho, que podem afetar negativamente sua integridade e eficácia operacional, pois redes de fibra são muito sensíveis a deformações. Sensores de alta precisão, estrategicamente instalados ao longo dos dutos para coletar dados sobre a condição física dos mesmos, incluindo a integridade dos arames internos de tração dos dutos, são essenciais para prevenir deformações excessivas. A deterioração de um único arame pode precipitar falhas exponenciais, potencializando riscos críticos ao sistema. Portanto, o monitoramento em tempo real e a análise preditiva se fazem cruciais.

Para mitigar riscos de falhas dispendiosas e reduzir a frequência de manutenções preventivas e corretivas, a empresa busca implementar um sistema avançado de monitoramento usando IA. Os dados coletados pelos sensores são armazenados em pastas e dados .csv. Não há como transmitir os dados via rede, pois vários sensores estão instalados em plataformas em alto-mar. Essas plataformas de monitoramento conectam os dutos e são gerenciadas por pessoas, pois não há como instalar dutos que atravessam continentes sem pontos de conexão. Em outras palavras, não há conectividade com a internet nas plataformas que coletam dados. Cada plataforma possui seu padrão de organização de dados e possui seu contexto ambiental.

A plataforma deve incorporar algoritmos de machine learning para analisar padrões nos dados coletados e identificar sinais precoces de possíveis falhas, permitindo intervenções proativas para manter a operacionalidade do sistema. A análise preditiva ajudará a empresa a planejar melhor suas operações de manutenção e inspeção, reduzindo custos e maximizando a eficiência. A Empresa acredita que os anos de dados coletados são suficientes para desenvolver um ótimo modelo.

Com base na descrição do cenário baseado em uma entrevista com o cliente, responda:

1. (Pipeline de Machine Learning)

(a) Explique como você estruturaria o processo de classificação de falhas, em um OU mais sensores, dentre os passos de implantação do modelo (A), engenharia de características (B), pré-processamento (C), treinamento do modelo(D), aquisição dos dados(E), avaliação do modelo(F) e seleção de características(G).

Resposta:

- **(E) Aquisição dos dados:** os dados são coletados por sensores instalados ao longo dos dutos e salvos como arquivos .csv em dispositivos físicos, já que não há conexão de rede. O primeiro passo seria adquirir esses dados brutos.
- **(C) Pré-processamento:** se cada plataforma de monitoramento tem seu próprio padrão de organização, o primeiro passo seria colocar essas dados em um único padrão. Em seguida, aplicar outras técnicas como remoção de sensores defeituosos (dados nulos), Interpolação de pequenas falhas de leitura de sensor em temos específicos, normalização, etc.
- **(B) Engenharia de características:** A partir dos dados crus dos sensores, extraem-se variáveis derivadas como média, mediana, desvio padrão, entropia, entre outros.
- **(G) Seleção de características:** Após gerar dezenas de variáveis, é necessário reduzir a dimensionalidade. Esse passo é opcional caso o modelo esteja overfitando.
- **(D) Treinamento do modelo:** A escolha depende do volume de dados e da disponibilidade de rótulos de falha. O treinamento pode ser feito por plataforma baseado nas localizações das mesmas ou por época do ano (devido a mudança de contexto ambiental).
- **(F) Avaliação do modelo:** Como o problema é desbalanceado (poucas falhas), **acurácia não é suficiente**. Métricas importantes incluem precisão, recall e f1-score:
 - **Implantação do modelo:** O modelo precisa ser exportado em formato leve (como .pkl, .onnx ou .tflite). A predição em tempo real deve gerar sinais claros de alerta local (ex.: arquivos de alarme ou dashboards embarcados) e logs que possam ser auditados posteriormente).

(b) Discuta quais estratégias de divisão (hold-out ou cross-validation) poderiam ser usadas no contexto de ter poucos dados e no contexto de ter muitos dados, justificando sua escolha.

Resposta:

- **Poucos dados:**

A melhor abordagem é a **validação cruzada (cross-validation)**, especialmente em séries temporais:

 - Se os dados forem temporais: **TimeSeriesSplit** (mantém ordem temporal, sem embaralhar);
Vantagem: usa quase todos os dados para treino e ainda fornece estimativas de variância do desempenho.
- **Muitos dados:**

Um **hold-out estratificado** com 70% treino / 15% validação / 15% teste é suficiente. Deve-se lembrar de respeitar a ordem cronológica dos dados. Com bastante amostras, a variância da avaliação é pequena, e a cross-validation se torna mais custosa e desnecessária. Pode-se ainda usar uma base externa (ex.: outra plataforma) como **conjunto de teste não visto**, garantindo uma avaliação realista da capacidade de generalização do modelo.

Resumo:

Situação	Técnica recomendada	Justificativa
Poucos dados	Cross-validation (TimeSeriesSplit)	Maximiza uso de dados; reduz viés de partição

Situação	Técnica recomendada	Justificativa
Muitos dados	Hold-out estratificado temporal	Mais simples e rápido; boa estimativa de generalização

2. (Overfitting, Underfitting e Generalização)

Em um sistema que monitora deformações em dutos subaquáticos, quais seriam as consequências de um modelo com overfitting? E de um modelo com underfitting? Explique ambos os conceitos e quais as suposições que estagiários na área de IA poderiam fazer ao treinar.

Resposta:

- Overfitting

Um modelo com overfitting aprende tão bem os dados de treino que passa a memorizar ruídos e padrões específicos daquela amostra.

Consequência no sistema de dutos:

- No contexto de dutos subaquáticos, pode levar o modelo a **falhar ao identificar deformações reais em plataformas com condições ambientais diferentes** daquelas usadas no treinamento — justamente por ter memorizado padrões específicos ao invés de aprender padrões generalizáveis de falha.
- Isso pode resultar em **ruptura inesperada do cabo** e perda total de capacidade de transmissão.

- Underfitting

Ocorre quando o modelo é **simples demais** ou mal treinado e não consegue capturar nem os padrões gerais dos dados.

Consequência no sistema:

- O modelo **não detecta sinais reais de falha**, ignorando tendências sutis nos sensores que indicariam deterioração progressiva.
- Isso pode resultar em **ruptura inesperada do cabo** e perda total de capacidade de transmissão.

- Erros comuns de estagiários em IA

- Avaliar o modelo com métricas inadequadas**, como acurácia em problemas desbalanceados.
- Assumir que métricas no treino é sinal de sucesso**, sem avaliar generalização no conjunto de teste.
- Escolher modelos complexos demais logo no início** (ex: redes neurais profundas) mesmo com poucos dados.
- Ignorar o pré-processamento**, achando que o modelo “vai aprender tudo sozinho”.

3. (Métricas e Classes Desbalanceadas)

No caso de detecção de falhas (classificação binária: “falha” vs. “não falha”), suponha que a maioria das amostras coletadas indicam funcionamento normal, e apenas uma minoria contenha falhas. Por que a acurácia sozinha pode ser enganosa? Quais métricas adicionais seriam mais adequadas para avaliar o modelo e por quê?

Resposta: Se 99 % das instâncias são **Não falha**, um classificador “sempre não falha” atinge 99 % de acurácia e **zero utilidade**. Métricas mais adequadas:

- **Recall (Sensibilidade)** – chance de captar uma falha verdadeira.
- **Precision** – entre os alertas, quantos realmente são falhas.
- **F1-score** – harmônico dos dois, útil quando ambos importam.
- **ROC-AUC e PR-AUC** – áreas sob curvas; PR-AUC destaca classe minoritária.

4. (Inspiração Biológica e Neurônios Artificiais)

Relacione o funcionamento básico de um neurônio biológico (dendritos, soma, axônio, sinapses) com o neurônio artificial em uma rede. Destaque como a soma ponderada das entradas e a função de ativação permitem modelar a ideia de excitação ou inibição do neurônio.

- Dendritos recebem sinais x_i .
- As sinapses modulam a força — em RNAs os pesos w_i .
- A soma no corpo celular (soma) corresponde à soma ponderada $z = \sum w_i x_i + b$
- Quando z ultrapassa o limiar, o axônio dispara; na RNA usa-se função de ativação $\sigma(z)$ (ReLU, sigmoid).
- Pesos positivos \leftrightarrow sinapses excitatórias, pesos negativos \leftrightarrow inibitórias.

5. Projeto de Redes Neurais Artificiais

Considere $X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$, $y = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$ e $\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Calcule **uma** atualização de gradiente descendente em lote $m=2$ com taxa $\alpha=0,1$. Mostre todos os passos até o novo $\theta(1)$.

Dados da questão:

$$X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} \\ x_0^{(2)} & x_1^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}, y = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \text{ e } \theta^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, m = 2 \text{ e } \alpha = 0,1$$

Forma expandida da hipótese:

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n, \text{ com } x_0 = 1.$$

A forma **vetorial compacta** $h_{\theta}(x) = \theta^T x$ é **válida para um único exemplo** x , e desde que x seja um **vetor coluna**:

$$h_{\theta}(x) = \theta^T x, \text{ com } \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}, \quad x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Estaria correto, mas você terá que fazer as predições exemplo a exemplo. Para calcular as predições em batch e economizar contas, vamos usar:

$$h_{\theta}(X) = X\theta$$

onde:

- X contém os exemplos como **linhas** (cada linha = um $x^{(i)}$),
- θ é um vetor coluna,
- O resultado $h_{\theta}(X)$ é um vetor com m previsões, um para cada exemplo.

Assim, para calcular as previsões podemos fazer:

$$h_{\theta}(X) = X\theta = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{ onde } y^{(1)} = 0 \text{ e } y^{(2)} = 0$$

O próximo passo é calcular os erros, fazendo:

$$e = y' - y = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} -4 \\ -5 \end{bmatrix}$$

Agora vamos calcular os gradientes:

$$\nabla_{\theta_j} J = \frac{1}{m} \sum_{i=1}^2 (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}, \text{ onde } h_{\theta}(x^{(i)}) - y^{(i)} = e^{(i)}$$

- $\nabla_{\theta_0} J = \frac{1}{m} \sum_{i=1}^2 e^{(i)} x_0^{(i)} = \frac{1}{2} [(-4 * 1) + (-5 * 1)] = -4.5$
- $\nabla_{\theta_1} J = \frac{1}{m} \sum_{i=1}^2 e^{(i)} x_1^{(i)} = \frac{1}{2} [(-4 * 2) + (-5 * 3)] = -11.5$

Por último faremos a atualização simultânea dos parâmetros:

$$\theta_0^{(1)} = \theta_0^{(0)} - \alpha \nabla_{\theta_0} J = 0 - 0.1(-4.5) = 0.45$$

$$\theta_1^{(1)} = \theta_1^{(0)} - \alpha \nabla_{\theta_1} J = 0 - 0.1(-11.5) = 1.15$$

Vetor de parâmetros após **uma** iteração:

$$\theta^{(1)} = \begin{bmatrix} 0.45 \\ 1.15 \end{bmatrix}$$

6. O trecho abaixo (adaptado de Functions/gradient_descent.py) atualiza os parâmetros:

gradient = (1/m) * (X.T @ erro)

theta = theta + alpha * gradient

- (a)** Explique o papel de alpha na convergência. O que acontece se alpha for muito pequeno; muito grande? Justifique usando o gráfico “Gradient descent intuition” (ou algum outro) visto em aula e nos exercícios.

Resposta: **Papel da α (learning-rate)**

- **Muito pequeno** → passos minúsculos, converge, mas leva horas/dias (*plateau*).
- **Muito grande** → ultrapassa o mínimo, oscila ou diverge (curva de custo “serra”).
- **Ideal** → diminui exponencialmente o custo até a região convexa, depois estabiliza.

(No gráfico “Gradient descent intuition”, curvas suaves mostram α ótimo; curvas em zigue-zague apontam α excessivo.)

(b) A Equação está correta? Explique. (Só será aceito resposta devidamente explicada).

Resposta: Não. *O sinal está invertido.* O gradiente aponta **na direção de maior aumento** de J; para **descer**, deve-se **subtrair**. Com o “+”, o algoritmo sobe o vale do erro; o custo cresce a cada época, inviabilizando o treinamento.