

Estudo Dirigido - MLP Back-propagation

Chegou a hora de aplicar os conhecimentos adquiridos de forma mais concreta com ferramentas reais. Mesmo as IAs de hoje ainda têm um longo caminho a percorrer para atingir objetivos aparentemente simples para os quais foram projetadas. Utilize esta avaliação como uma oportunidade de aprendizado e faça o melhor que puder. É esperado que, além de conseguir entregar os resultados pedidos, você possa aprender ainda mais sobre o assunto e descobrir novas funcionalidades e aplicações do tema.

Chamamos de Aprendizado de Máquina o conjunto de técnicas utilizadas em IA que utilizam algoritmos para aprender a realizar tarefas a partir de experiências passadas. As técnicas de Aprendizado de Máquina podem ser divididas em uma série de categorias, das quais duas se destacam: o Aprendizado Supervisionado e o Aprendizado Não-supervisionado. Com o Aprendizado Supervisionado a máquina aprende por meio de exemplos rotulados, ou seja, podemos resolver problemas de Classificação (separar exemplos em classes) ou Regressão (aproximar funções matemáticas complexas).

Por exemplo, se quisermos avaliar quais alunos de uma determinada escola estão usando máscara, estamos diante de um problema de classificação. As classes possíveis são: aluno com máscara ou aluno sem máscara. Neste caso, cada exemplo rotulado será composto por uma foto e seu rótulo (representado pelo nome da classe a qual pertence). Tipicamente são necessários diversos exemplos rotulados para que o algoritmo possa aprender os padrões necessários para resolver uma tarefa de classificação.

Existem vários algoritmos de Classificação, dentre eles podemos citar: Árvore de Decisão (do inglês Decision Tree), k-Vizinhos Mais Próximos (k-Nearest Neighbours), Naïve Bayes, Máquina de Vetores Suporte (Support Vector Machine), as Redes Neurais MLP Back-propagation, etc. Esses algoritmos podem ser implementados em diversas linguagens de programação. Muitas bibliotecas, para diversas linguagens de programação, disponibilizam implementações desses métodos de aprendizado. Além disso, para os que não tem conhecimentos de programação, existem plataformas que possibilitam a resolução de um problema de Aprendizado de Máquina de modo bastante intuitivo. A nível de referência, alguns links de recursos que você pode utilizar para resolver esta avaliação:

- Sem programação: <http://www.aispace.org/downloads.shtml>
- Sem programação: Orange (<https://orangedatamining.com/>)
- Programação em Python: Scikit-learn (<https://scikit-learn.org/stable/>)

Para utilizar o software Orange você não precisa ter conhecimento prévio em programação, já para utilizar a biblioteca Scikit-learn você precisa ter conhecimento prévio de programação na linguagem Python, a mais comumente utilizada na comunidade de IA. Abaixo, alguns conteúdos sobre a utilização do Orange e da Scikit-learn que podem ser utilizados para a construção da solução desta avaliação:

- [Salvando Predições no Orange Data Mining](#)
- [Machine Learning sem código. Usando Orange Data Mining para criar um... | by BrunoBatista | Ensina.AI](#)

- -04 PREDIÇÕES - ORANGE CANVAS
- [Introdução ao Colab e Scikit-learn](#)
- [Implementando um Modelo de Classificação no Scikit-Learn*](#)

Para exemplificar, considere uma aplicação de IA visando elevar a produtividade do agronegócio. Atualmente, além da tradicional previsão do tempo, muitos agricultores já têm acesso às pré-medições de temperatura, precipitação, direção e velocidade dos ventos e de mais fenômenos climáticos que podem influenciar positivamente ou negativamente nas plantações. Tendo em vista este contexto, uma fazenda de arroz da região Sul do Brasil disponibilizou dados sobre fenômenos climáticos coletados diariamente durante 7 anos, os quais devem ser utilizados para a construção de um modelo capaz de prever se os fenômenos climáticos são ou não são ideais para plantar. A planilha de treinamento fornecida tem 1.794 instâncias (linhas) e 15 características das instâncias (colunas).

id	col_1	col_2	col_3	col_4	col_5	col_6	col_7	col_8	col_9	col_10	col_11	col_12	col_13	col_14	col_15	target
1	6.338431	8.550782	5.624511	8.536958	8.536958	8.346047	8.399787	8.473381	8.473381	8.634096	8.489434	9.728270	8.665812	2.804327	8.260737	0
2	9.146805	16.022925	4.934298	7.067574	7.067574	7.034689	6.556745	5.447100	5.447100	7.562939	5.264155	4.817431	7.822631	3.762136	6.954110	0
3	11.315154	19.245600	4.014962	2.828132	2.828132	8.991663	9.886247	8.060166	8.060166	8.557616	6.300734	8.535757	6.857591	2.978528	6.761856	0
4	7.206610	14.908040	5.526639	7.857803	7.857803	8.876916	6.837429	5.250751	5.250751	7.325638	8.010435	5.240311	10.009655	1.365952	5.066801	1
5	8.756336	20.039245	6.071641	1.803756	1.803756	10.584192	7.530467	7.652314	7.652314	9.953948	3.804962	5.478215	3.667689	1.836328	7.778353	0

Figura 1: Exemplo dos dados de treinamento.

Figura 1 apresenta um pedaço do conjunto original de dados disponível para o experimento. A coluna id identifica individualmente cada uma das 1.794 instâncias, enquanto a coluna target apresenta os rótulos associados: valor 1 para instâncias que representam momentos ideais para o plantio, e valor 0 caso contrário. As demais colunas contêm valores numéricos que representam dados climáticos, os quais devem embasar a decisão do modelo sobre o plantio.

Recomenda-se dividir o conjunto original em duas partes: uma para treinamento e outra para teste. Essa divisão permite avaliar corretamente o desempenho do modelo, garantindo que ele seja testado em dados inéditos, com rótulos conhecidos. Por exemplo, pode-se alocar 70% das instâncias para o treinamento e 30% para o teste, estratificando a amostragem para manter a proporção de rótulos nas duas divisões. A partir disso, o modelo pode ser treinado com os dados de treinamento e avaliado com os dados de teste, garantindo uma análise confiável da sua capacidade de generalização.

Nesta avaliação você vai trabalhar com um problema de Classificação detalhado em: [GitHub - zalandoresearch/fashion-mnist: A MNIST-like fashion product database. Benchmark](#). Fashion-MNIST é um conjunto de dados de imagens de artigos de Zalando que consiste em um conjunto de treinamento de 60.000 exemplos e um conjunto de teste de 10.000 exemplos. Cada exemplo é uma imagem em tons de cinza 28x28, associada a um rótulo de 10 classes ([fashion_mnist | TensorFlow Datasets](#)). Implemente a sequência de passos a seguir.

1. Exploração Inicial – Avaliando Pesos Iniciais e Funções de ativação

Considere uma topologia de rede. Escolha as funções de ativação para os neurônios, justificando-as de acordo com a natureza do problema. Para as camadas ocultas, recomenda-se o uso de funções como ReLU, sigmoide ou tanh, conforme o comportamento esperado das ativações. Para a camada de saída, utilize a função Softmax, que converte os valores de saída em uma distribuição de probabilidades sobre as classes, sendo a escolha adequada para problemas de classificação multiclasse, como o Fashion-MNIST.

Execute cinco treinamentos para a rede MLP, cada um com uma inicialização de pesos diferente (reinic peace o gerador de números aleatórios em cada treinamento). Avaliar diferentes inicializações é essencial para compreender a estabilidade do modelo e o impacto do ponto de partida no processo de aprendizado.

Durante a Exploração Inicial, apenas o conjunto de treinamento deve ser utilizado. O objetivo é verificar se o modelo consegue aprender padrões básicos a partir dos dados rotulados fornecidos. O conjunto de teste não deve ser utilizado nesta etapa.

Métricas a Serem Avaliadas

- Função de Perda (Loss): utilizar a Entropia Cruzada Categórica (Categorical Cross-Entropy) para monitorar a convergência do modelo e ajustar os pesos durante o treinamento.
- Acurácia (Accuracy): medir a proporção de classificações corretas sobre o total de exemplos no conjunto de treinamento, avaliando se o modelo está de fato aprendendo os padrões das classes.
- Curva de Convergência: traçar gráficos da perda e da acurácia em função das épocas para observar se o modelo está aprendendo corretamente ou apresentando sinais de overfitting ou underfitting.

Relato no relatório final

Ao final da Questão 1, o estudante deve registrar e analisar os resultados obtidos, apresentando um breve relato técnico estruturado, contendo os seguintes elementos:

a. Descrição da configuração experimental:

- Arquitetura da rede (número de camadas e neurônios por camada);
- Funções de ativação utilizadas (ReLU e Softmax);
- Função de perda (Entropia Cruzada Categórica) e otimizador empregado;
- Número de épocas e taxa de aprendizado;
- Justificativa das escolhas feitas.

b. Apresentação dos resultados:

- Tabela com os resultados dos cinco treinamentos, incluindo perda final e acurácia final de cada um;
- Gráficos da perda (loss) e da acurácia (accuracy) ao longo das épocas, comparando as diferentes inicializações.

c. Análise e discussão:

- Comparação entre os cinco treinamentos (diferenças de convergência, estabilidade e desempenho);
- Identificação de eventuais indícios de underfitting ou overfitting;

- Discussão sobre a sensibilidade do modelo à inicialização dos pesos;
- Observações sobre o comportamento geral do aprendizado.

d. Conclusão da etapa:

- Síntese dos principais achados;
- Indicação de que o modelo foi capaz (ou não) de aprender padrões básicos;
- Avaliação da adequação das funções de ativação e da estabilidade do treinamento.

A topologia definida e validada nesta etapa (com funções de ativação e estrutura comprovadamente estáveis) servirá de base para a Questão 2, que investigará a **influência da taxa de aprendizado e do termo momento** na eficiência e estabilidade do processo de treinamento dessa mesma arquitetura.

2. Exploração de Hiperparâmetros de Aprendizado – Taxa de Aprendizado e Termo Momento

Explore a relação entre taxa de aprendizado e termo momento e a velocidade de convergência do treinamento para cada configuração, quando o erro de treinamento escolhido para parar o processo de treinamento for 0.001. Utilize uma abordagem sistemática como Grid Search, que explora combinações de parâmetros (por exemplo, taxas de aprendizado [0.01, 0.1, 0.5] e momentos [0.5, 0.7, 0.9]). Explorar uma gama mais ampla de valores pode revelar tendências mais detalhadas. Neste contexto, é interessante incluir taxas de aprendizado menores, como 0.01 ou 0.001, e momentos intermediários (e.g., 0.7). O conjunto de treinamento deve ser o principal conjunto utilizado nesta etapa, já que o foco é analisar como diferentes configurações de hiperparâmetros impactam o aprendizado do modelo nos dados disponíveis para treinamento. Um **conjunto de validação** (opcional) pode ser separado a partir do conjunto de treinamento para auxiliar na escolha de hiperparâmetros, monitorando a generalização do modelo e evitando que ele apenas memorize os dados de treinamento. O conjunto de teste deve ser reservado exclusivamente para a avaliação final do modelo, após a escolha dos melhores hiperparâmetros.

Métricas a Serem Avaliadas

- Função de Perda (Loss): métrica principal para avaliar a eficácia do modelo durante o treinamento. O objetivo é observar como a perda diminui em função do número de épocas para diferentes combinações de hiperparâmetros. A curva da perda ao longo das épocas fornece insights sobre a estabilidade e velocidade de convergência.
- Velocidade de Convergência: determinada pelo número de épocas necessárias para atingir o critério de parada do erro (e.g., perda ≤ 0.001). Isso ajuda a identificar combinações de hiperparâmetros que promovem um aprendizado eficiente.
- Curvas de Convergência: gráficos da perda (loss) ao longo das épocas são úteis para visualizar a relação entre os hiperparâmetros e o desempenho do modelo. Permitem identificar problemas como divergência (quando a taxa de aprendizado é muito alta) ou aprendizado muito lento (taxa de aprendizado muito baixa).
- Estabilidade (opcional): avaliar se a inicialização de pesos e a combinação de hiperparâmetros resultam em treinamento consistente ao repetir o processo com diferentes seeds.

Relato no relatório final

Ao final da Questão 2, o estudante deve apresentar um relato técnico comparativo sobre o impacto da variação dos hiperparâmetros na eficiência e estabilidade do treinamento. O relatório deve conter os seguintes elementos:

a. Descrição do experimento:

- Intervalos e valores testados para taxa de aprendizado e termo momento;
- Função de ativação e função de perda utilizadas;
- Critério de parada adotado (perda ≤ 0.001);
- Quantidade de épocas e método de otimização.

b. Resultados obtidos:

- Tabelas comparando os resultados de perda final e número de épocas até convergência para cada combinação de hiperparâmetros;
- Gráficos da perda (loss) ao longo das épocas para diferentes configurações;
- Indicação da combinação que apresentou o melhor equilíbrio entre velocidade e estabilidade.

c. Análise e discussão:

- Identificação das tendências observadas (e.g., taxas de aprendizado muito altas geram instabilidade; muito baixas geram lentidão);
- Comparação do comportamento das curvas de perda entre diferentes momentos e taxas;
- Discussão sobre qual configuração promoveu o aprendizado mais eficiente e estável.

d. Conclusão da etapa:

- Síntese dos principais achados e justificativa da escolha da melhor combinação de hiperparâmetros;
- Observações sobre o impacto do termo momento e da taxa de aprendizado na convergência da rede.

Os resultados desta etapa servirão de base para a próxima fase da avaliação, dedicada à exploração da topologia da rede e à análise do impacto do número de camadas e neurônios sobre o desempenho.

3. Topologia da Rede – Influência do Número de Camadas e Neurônios por Camada

Explore a influência do número de camadas ocultas e do número de neurônios por camada sobre o desempenho do modelo **nos conjuntos de treinamento e validação**. Esta etapa é fundamental para equilibrar desempenho e tempo de treinamento, ajustando a complexidade da rede à dificuldade do problema. Teste sistematicamente variações no número de camadas e neurônios (por exemplo, de 1 a 3 camadas ocultas e de 10 a 100 neurônios por camada, em incrementos de 10). Utilize o conjunto de treinamento para calcular a função de perda (loss) e monitorar o comportamento do modelo durante o aprendizado. Use o conjunto de validação para acompanhar a generalização e detectar indícios de overfitting ou underfitting. O conjunto de teste não deve ser utilizado nesta etapa, sendo reservado apenas para a validação final, após a definição das melhores configurações topológicas.

Métricas a Serem Avaliadas

- Função de Perda (Loss): calcular a perda tanto para o conjunto de treinamento quanto para o conjunto de validação em cada configuração testada. Acompanhar a evolução da perda ao longo das épocas permite identificar padrões de aprendizado e estabilidade.
- Underfitting: quando a perda permanece alta tanto no treinamento quanto na validação.
- Overfitting: quando a perda no treinamento é baixa, mas aumenta no conjunto de validação.
- Curvas de Convergência: analisar gráficos da perda (loss) versus épocas para os conjuntos de treinamento e validação. Essas curvas ajudam a compreender o comportamento da rede para diferentes combinações de camadas e neurônios e a capacidade de generalização do modelo.
- Tempo de Treinamento: registrar o tempo necessário para cada configuração alcançar o critério de parada (por exemplo, perda ≤ 0.001 ou um número máximo de épocas). Relacionar a complexidade da rede (mais camadas ou neurônios) ao tempo computacional.
- Generalização (Medida-F): para as melhores topologias (com base na perda e na convergência), avaliar a Medida-F no conjunto de validação. A Medida-F combina precisão e revocação, oferecendo um indicador único de desempenho global.
- Precisão: proporção de previsões corretas entre todas as previsões positivas.
- Revocação: proporção de previsões corretas entre todos os rótulos positivos reais.

Relato no relatório final

Ao final da Questão 3, o estudante deve apresentar um relato técnico completo, descrevendo o impacto da variação da topologia (número de camadas e neurônios) no aprendizado e na generalização da rede. O relatório deve conter os seguintes elementos:

a. Descrição da configuração experimental:

- Faixas de valores testadas (número de camadas e de neurônios por camada);
- Funções de ativação e função de perda utilizadas;
- Critério de parada (perda ≤ 0.001 ou limite de épocas);
- Otimizador e parâmetros mantidos fixos durante o experimento.

b. Resultados obtidos:

- Tabela com as configurações testadas, incluindo perda final de treinamento, perda de validação, tempo de convergência e Medida-F (quando aplicável);
- Gráficos da perda (loss) para os conjuntos de treinamento e validação, mostrando as diferenças entre topologias.

c. Análise e discussão:

- Identificação de tendências entre o aumento da complexidade da rede e o desempenho obtido;
- Análise dos casos de underfitting e overfitting observados;
- Discussão sobre o ponto de equilíbrio entre capacidade de generalização e tempo de treinamento.

d. Seleção das melhores topologias:

- Selecionar e registrar as quatro melhores configurações (ou treinamentos) com base nas métricas de desempenho e estabilidade;

- Justificar a escolha de cada uma das quatro topologias selecionadas, que serão utilizadas na etapa seguinte (validação inicial com o conjunto de teste).

e. Conclusão da etapa:

- Síntese das observações gerais sobre a influência da arquitetura na performance da rede;
- Comentários sobre como o número de camadas e neurônios impactou o aprendizado e a generalização.

Esses resultados servirão como base para a Questão 4 (Validação Inicial), na qual as quatro melhores configurações serão comparadas no conjunto de teste para determinar o modelo mais robusto.

4. Influência dos Dados de Treinamento – Qualidade e Quantidade de Exemplos

Explore a influência do número de exemplos e da qualidade dos exemplos escolhidos sobre a capacidade de generalização da melhor rede obtida nas etapas anteriores. Esta análise busca compreender como o volume e a representatividade dos dados impactam o desempenho e a estabilidade do modelo. Para isso, varie sistematicamente a quantidade de amostras utilizadas no treinamento, garantindo que as divisões sejam estratificadas de modo a manter a proporcionalidade entre as classes. Essa abordagem assegura que o conjunto de treinamento seja bem distribuído e representativo do problema de classificação. O objetivo é observar como diferentes tamanhos de conjuntos de treinamento afetam a perda (loss), a acurácia e a generalização do modelo.

Métricas a Serem Avaliadas

- Função de Perda (Loss): calcular a perda para os conjuntos de treinamento e validação, observando como ela varia com o aumento ou redução do número de exemplos utilizados. Acompanhar a perda ao longo das épocas permite identificar se há saturação no aprendizado conforme o conjunto cresce.
- Acurácia (Accuracy): avaliar a proporção de acertos no conjunto de validação para cada tamanho de conjunto de treinamento. Essa métrica indica se o aumento de dados está, de fato, melhorando a capacidade de generalização.
- Medida-F (F1-Score): calcular a Medida-F para o conjunto de validação e, posteriormente, para o conjunto de teste, sintetizando precisão e revocação em um único indicador de desempenho global.
- Curvas de Generalização: construir gráficos relacionando o tamanho do conjunto de treinamento ao desempenho (loss, acurácia ou Medida-F) nos conjuntos de validação e teste. Essas curvas permitem identificar pontos de saturação ou de melhora marginal com o aumento de dados.
- Tempo de Treinamento: monitorar o impacto do tamanho do conjunto de treinamento no tempo de convergência da rede, avaliando o custo computacional associado ao aumento da base de dados.

Relato no relatório final

Ao final da Questão 4, o estudante deve apresentar um relato comparativo e interpretativo sobre o efeito da quantidade e da qualidade dos dados de treinamento no desempenho da rede. O relatório deve conter os seguintes elementos:

a. Descrição do experimento:

- Faixas de tamanhos de conjuntos de treinamento testadas (por exemplo, 20%, 40%, 60%, 80% e 100% dos dados disponíveis);
- Estratégia de amostragem utilizada (estratificada, aleatória ou outra);

- Funções de ativação, função de perda e otimizador mantidos constantes em relação ao melhor modelo obtido na etapa anterior.

b. Resultados obtidos:

- Tabelas apresentando, para cada tamanho de conjunto de treinamento, os valores finais de perda, acurácia, Medida-F e tempo de treinamento;

- Gráficos das curvas de generalização, mostrando o comportamento do modelo nos conjuntos de treinamento, validação e teste.

c. Análise e discussão:

- Interpretação das curvas de generalização, identificando se o modelo apresenta melhora contínua ou saturação com o aumento de dados;

- Discussão sobre a influência da qualidade e da representatividade dos exemplos na capacidade de generalização;

- Avaliação de eventuais variações no tempo de convergência com diferentes quantidades de dados.

d. Conclusão da etapa:

- Síntese das observações gerais sobre o impacto da quantidade e da qualidade dos exemplos no desempenho do modelo;

- Indicação do ponto de equilíbrio entre desempenho, tempo de treinamento e disponibilidade de dados.

Os resultados desta etapa consolidam a compreensão da influência dos dados sobre o aprendizado e servirão de base para a etapa seguinte, de Validação Inicial, onde as melhores configurações serão comparadas empregando o conjunto de teste.

5. Validação Inicial – Conjuntos de Treinamento e Teste

Após escolher as quatro melhores configurações obtidas nas etapas anteriores (diferentes topologias e hiperparâmetros), realize a validação inicial aplicando o conjunto de teste a cada uma delas. O objetivo é comparar o desempenho e a capacidade de generalização de cada rede, analisando métricas de erro, estabilidade e precisão. O conjunto de treinamento servirá apenas como referência comparativa, enquanto o conjunto de teste será usado exclusivamente para avaliar a generalização das redes e identificar a configuração mais robusta.

Durante esta etapa, o estudante deverá monitorar as métricas de desempenho, registrar o comportamento das curvas de convergência e justificar os ajustes realizados para otimizar o desempenho antes da aplicação da validação cruzada.

Métricas a Serem Avaliadas

- Função de Perda (Loss): calcular a perda (entropia cruzada categórica) para os conjuntos de treinamento e teste, monitorando a evolução ao longo das épocas e comparando a estabilidade entre as configurações.

- Acurácia (Accuracy): avaliar a proporção de acertos nos conjuntos de treinamento e teste, medindo o grau de aprendizado e generalização do modelo.

- Curvas de Validação: traçar gráficos de perda e acurácia ao longo das épocas para os conjuntos de treinamento e teste, observando padrões de underfitting ou overfitting.

- Medida-F (F1-Score): avaliar a qualidade das previsões no conjunto de teste, combinando precisão e revocação em um único indicador de desempenho global.

- Precisão: proporção de previsões positivas corretas em relação ao total de previsões positivas.
- Revocação: proporção de previsões positivas corretas em relação ao total de rótulos positivos reais.

Relato no relatório final

Ao final da Questão 5, o estudante deve apresentar um relato técnico comparativo entre as quatro melhores configurações testadas, contendo:

a. Descrição do experimento:

- Arquitetura e parâmetros das quatro redes selecionadas (número de camadas, neurônios, taxa de aprendizado, termo momento etc.);
- Critérios de seleção dessas topologias a partir das etapas anteriores;
- Conjuntos utilizados (treinamento e teste) e função de perda empregada.

b. Resultados obtidos:

- Tabela comparando, para cada rede, as métricas de perda, acurácia e Medida-F nos conjuntos de treinamento e teste;
- Gráficos das curvas de validação (perda e acurácia ao longo das épocas) para cada rede;
- Indicação da rede com melhor desempenho e maior estabilidade.

c. Análise e discussão:

- Comparação do comportamento entre as quatro configurações;
- Identificação de padrões de overfitting e underfitting;
- Discussão sobre o equilíbrio entre desempenho no treinamento e generalização no teste;
- Justificativa dos ajustes realizados (hiperparâmetros, topologia, regularização, etc.).

d. Conclusão da etapa:

- Identificação da melhor configuração final entre as quatro redes;
- Justificativa técnica da escolha;
- Preparação para a próxima etapa (Validação Cruzada), destacando por que essa configuração será usada como base.

A configuração com melhor desempenho nesta etapa será a única utilizada na **Questão 6** para validação cruzada (k -fold).

6. Validação Cruzada e Comparativa de Configurações Finais – Avaliação Consolidada

Após a seleção da melhor configuração na Questão 5 (a melhor configuração obtida nas etapas anteriores: topologia, hiperparâmetros e dados de treinamento), realize validação cruzada k -fold dessa configuração vencedora para avaliar robustez e consistência em diferentes partições (subconjuntos) dos dados. Essa técnica permite verificar se o desempenho observado não depende apenas de uma divisão específica entre treinamento e teste, garantindo uma avaliação mais confiável da capacidade de generalização do modelo.

Divida o conjunto de dados em k partições (por exemplo, $k = 5$ ou $k = 10$). Cada partição deve ser usada uma vez como conjunto de teste, enquanto as demais servem como conjunto de treinamento. O processo é repetido k vezes, e ao final calcula-se a média e a variância das métricas obtidas. Essa

abordagem oferece uma estimativa mais estável e estatisticamente relevante do desempenho final da rede MLP.

Métricas a Serem Avaliadas

- Função de Perda (Loss): calcular a perda média e o desvio-padrão da perda entre as diferentes partições, avaliando a estabilidade do aprendizado.
- Acurácia (Accuracy): medir a proporção média de acertos entre as partições e observar a variação do desempenho em diferentes subconjuntos dos dados.
- Medida-F (F1-Score): avaliar a qualidade global das previsões em cada partição e calcular a média e a variância das Medidas-F.
- Curvas de Validação: traçar as curvas de perda e acurácia (ou Medida-F) ao longo das épocas para cada partição, permitindo identificar eventuais sinais de overfitting ou underfitting.
- Variância dos Resultados: analisar a dispersão das métricas (perda, acurácia e Medida-F) entre as diferentes partições, o que indica o grau de consistência do modelo.

Relato no relatório final

Ao final da Questão 6, o estudante deve apresentar um relato consolidado da validação cruzada, evidenciando a consistência e a generalização da rede final. O relatório deve conter os seguintes elementos:

1. Descrição do experimento:

- Valor de k utilizado na validação cruzada e justificativa da escolha;
- Configuração final do modelo (número de camadas, neurônios, taxa de aprendizado, termo momento, função de ativação, função de perda e otimizador);
- Procedimentos adotados para a divisão dos dados e execução dos k treinamentos.

b. Resultados obtidos:

- Tabelas com as métricas de perda, acurácia e Medida-F obtidas em cada partição;
- Cálculo da média e da variância das métricas entre as partições;
- Gráficos das curvas de validação (perda e acurácia) para algumas partições representativas.

c. Análise e discussão:

- Interpretação dos resultados médios e da variabilidade entre partições;
- Avaliação da robustez do modelo frente a diferentes amostragens;
- Identificação de eventuais flutuações e discussão de suas causas (ex.: dados não balanceados, complexidade da rede, etc.).

d. Conclusão da etapa:

- Síntese dos resultados gerais da validação cruzada;
- Conclusões sobre a consistência e a capacidade de generalização do modelo;
- Observações sobre possíveis melhorias ou ajustes futuros na rede MLP.

Os resultados desta etapa consolidam a avaliação final do modelo e devem ser incluídos na discussão geral do relatório, demonstrando a robustez da solução desenvolvida e a validade das escolhas feitas ao longo de todas as etapas do experimento.