

Relatório: Análise Comparativa de Redes Neurais Artificiais Clássicas no Conjunto Iris

1. Introdução e Contextualização

O presente trabalho tem como objetivo consolidar a compreensão prática e teórica de três arquiteturas fundamentais de Redes Neurais Artificiais (RNAs): o **Perceptron** de camada única, o **ADALINE** (*Adaptive Linear Neuron*) e o **Perceptron Multicamadas (MLP)**.

O estudo de caso utiliza o conjunto de dados **Iris**, um *benchmark* clássico no reconhecimento de padrões. O desafio consiste em classificar amostras de plantas em três espécies (*Setosa*, *Versicolor* e *Virginica*) com base em quatro atributos contínuos. A análise foca na capacidade de separação de classes, na estabilidade do treinamento e na comparação entre fronteiras de decisão lineares e não lineares.

2. Fundamentação Teórica

2.1 Perceptron (Camada Única)

O Perceptron, proposto por Frank Rosenblatt, é o modelo mais simples de um classificador neural linear.

- Formulação:** A saída é calculada pela soma ponderada das entradas ($z = \mathbf{w} \cdot \mathbf{x} + b$).
- Ativação e Aprendizado:** Utiliza uma função de ativação do tipo degrau (Heaviside). A regra de atualização dos pesos ocorre apenas quando há erro na classificação final (saída quantizada), ajustando o vetor de pesos na direção do erro.
- Limitações:** Sua principal restrição é a incapacidade de resolver problemas que não sejam linearmente separáveis. Se as classes se sobrepõem, o algoritmo não converge e os pesos oscilam.

2.2 ADALINE (Adaptive Linear Neuron)

O ADALINE, desenvolvido por Widrow e Hoff, representa uma evolução no algoritmo de treinamento.

- Diferença Conceitual:** Ao contrário do Perceptron, o ADALINE minimiza o erro da saída linear (pré-ativação) em relação ao alvo, e não o erro da saída quantizada.
- Função de Custo:** Utiliza o Erro Quadrático Médio (MSE - *Mean Squared Error*).
- Treinamento:** Aplica o algoritmo de **Gradiente Descendente** (*Gradient Descent*), permitindo atualizações contínuas e suaves dos pesos baseadas na derivada da função de custo. Isso tende a oferecer uma convergência mais estável, embora a fronteira de decisão final permaneça linear.

2.3 MLP (Multilayer Perceptron)

O MLP é uma rede neural *feedforward* que supera as limitações dos modelos lineares.

- **Arquitetura:** Composto por uma camada de entrada, pelo menos uma camada oculta e uma camada de saída.
- **Não-linearidade:** A introdução de funções de ativação não lineares (como ReLU ou Sigmoide) nas camadas ocultas permite que a rede aprenda mapeamentos complexos.
- **Backpropagation:** O treinamento utiliza a retropropagação do erro, calculando o gradiente da função de custo em relação a todos os pesos da rede (da saída para a entrada) através da regra da cadeia.

3. Metodologia Experimental

3.1 Dados e Pré-processamento

O conjunto Iris (150 amostras, 4 atributos) foi utilizado integralmente.

- **Pré-processamento:** Foi aplicada a **Padronização (StandardScaler)** para que os atributos tenham média 0 e desvio padrão 1. Esta etapa é obrigatória para o bom funcionamento do Gradiente Descendente no ADALINE e no MLP.

3.2 Procedimento

- **Divisão dos Dados:** 70% para treinamento e 30% para teste (estratificado para manter a proporção das classes).
- **Execuções:** Foram realizadas **10 execuções independentes** para garantir significância estatística.
- **Visualização:** Para os gráficos de fronteira, utilizou-se **PCA (Principal Component Analysis)** reduzindo os dados para 2 componentes principais, permitindo a visualização no plano cartesiano.

4. Resultados Experimentais

A tabela abaixo apresenta o desempenho agregado (média e desvio padrão) das 10 execuções realizadas.

Modelo	Acurácia Média	Desvio Padrão	Análise Preliminar
Perceptron	88.00%	0.0598	Bom desempenho, limitado pela linearidade.
ADALINE	82.00%	0.0491	Desempenho inferior devido à sensibilidade ao erro quadrático em classes sobrepostas.

MLP	94.00%	0.0386	Melhor desempenho e maior estabilidade (menor desvio).
------------	---------------	--------	--

Análise das Matrizes de Confusão (Acumuladas)

As matrizes abaixo somam os erros e acertos de todas as 10 execuções, revelando onde cada modelo falhou:

1. Perceptron:

[[144 6 0] <-- Setosa (quase perfeito)

[8 128 14] <-- Versicolor

[0 26 124]] <-- Virginica

Análise: O modelo erra ao tentar separar *Versicolor* e *Virginica*, distribuindo os erros de forma relativamente equilibrada entre as duas classes.

2. ADALINE:

[[146 4 0]

[0 97 53] <-- ERRO CRÍTICO EM VERSICOLOR

[0 24 126]]

Análise: O ADALINE apresentou o comportamento mais crítico. Classificou erroneamente **53 amostras de Versicolor como Virginica**. Isso indica que, na tentativa de minimizar o Erro Quadrático Médio global, a fronteira de decisão foi empurrada para cima da classe *Versicolor*, sacrificando sua acurácia em favor de um ajuste numérico dos pesos.

3. MLP:

[[148 2 0]

[0 136 14] <-- Redução drástica de erros

[0 11 139]]

Análise: O MLP obteve a melhor separação. Os erros na classe *Versicolor* caíram drasticamente (de 53 no Adaline para apenas 14 aqui). A rede conseguiu criar uma regra de decisão que preserva a identidade de ambas as classes sobrepostas.

5. Visualização e Fronteiras de Decisão

A análise visual dos gráficos gerados via PCA (Figura 1 do anexo) corrobora os dados numéricos:

1. **Perceptron (Linear Rígido):** As fronteiras são retas. O modelo tenta passar uma reta entre as classes azul e vermelha, mas acaba cortando o grupo branco (*Versicolor*) e vermelho (*Virginica*) de forma grosseira.
2. **ADALINE (Deslocamento da Fronteira):** O gráfico central mostra que a fronteira de decisão invadiu a região da classe central. Isso explica a acurácia de 82%: geometricamente, a reta "escolhida" pelo gradiente descendente para minimizar o erro quadrático não foi a melhor reta para maximizar a acurácia de classificação (contagem de acertos).
3. **MLP (Adaptação Não Linear):** O gráfico da direita mostra claramente uma fronteira poligonal/curva. O modelo consegue contornar a massa de dados da classe central, isolando-a de forma muito mais eficiente das outras duas.

6. Discussão Crítica e Conclusão

Os experimentos demonstraram que a complexidade do modelo deve ser adequada à complexidade dos dados.

- **Linearidade vs. Realidade:** O conjunto Iris possui duas classes (*Versicolor* e *Virginica*) que não são linearmente separáveis. Portanto, modelos estritamente lineares como Perceptron e ADALINE possuem um "teto" de desempenho teórico que não podem ultrapassar.
- **O "Fracasso" do ADALINE:** Foi interessante notar que o ADALINE performou pior que o Perceptron. Isso ocorre porque o Perceptron foca apenas em corrigir erros de classificação (discreto), enquanto o ADALINE tenta minimizar a magnitude do erro (contínuo). Em situações de forte sobreposição, o ADALINE pode convergir para uma solução que é matematicamente ótima para o erro quadrático, mas ruim para a acurácia final.
- **Eficácia do MLP:** O MLP, com sua camada oculta e ativação ReLU, provou ser a ferramenta correta para o problema. O custo computacional ligeiramente maior é justificado pelo ganho de 12% em acurácia sobre o ADALINE e pela estabilidade estatística apresentada.