

# Relatório Técnico: Avaliação de Modelos de Classificação em Padrões Não-Lineares

Aluno: Magno José Gonçalves da Silva

Email: mjgs@cesar.school

## 1. Resumo Executivo

O presente relatório detalha uma série de experimentos conduzidos utilizando o TensorFlow Playground para avaliar a capacidade de redes neurais em classificar o conjunto de dados **Espiral**, um problema de classificação não-linear. O estudo focou na influência da arquitetura da rede neural e da função de ativação na eficácia do modelo. A análise dos resultados indica que a combinação de uma função de ativação robusta com uma arquitetura otimizada, em detrimento do aumento indiscriminado de neurônios, resulta no desempenho mais superior.

## 2. Metodologia

Três experimentos de classificação foram realizados com o objetivo de analisar o impacto de parâmetros-chave. Os parâmetros mantidos fixos em todas as iterações foram o conjunto de dados (Espiral), a taxa de divisão entre treinamento e teste (50%), a ausência de ruído (0) e a taxa de aprendizado (0.03). Os parâmetros variáveis foram a arquitetura da rede e a função de ativação.

- Experimento 1:** Arquitetura 8-8-8 (3 camadas ocultas, 8 neurônios por camada). Função de ativação Tanh.
- Experimento 2:** Arquitetura 4-2-2 (3 camadas ocultas, número decrescente de neurônios). Função de ativação ReLU.
- Experimento 3:** Arquitetura 4-4-4 (3 camadas ocultas, 4 neurônios por camada). Função de ativação Tanh.

## 3. Resultados

O desempenho de cada modelo foi avaliado com base na métrica de perda (loss), tanto para o conjunto de dados de treinamento quanto para o de teste. Uma perda menor indica um melhor ajuste do modelo aos dados.

Experimento	Arquitetura	Função de Ativação	Perda de Teste	Perda de Treinamento
1	8-8-8	Tanh	0.009	0.008
2	4-2-2	ReLU	0.005	0.004
3	4-4-4	Tanh	0.010	0.007

Os dados demonstram que o **Experimento 2** obteve a menor perda de teste e de treinamento, indicando que sua configuração foi a mais eficaz na resolução do problema de classificação.

---

## 4. Discussão

### 4.1. Arquitetura da Rede e o Número de Neurônios

O problema da espiral é intrinsecamente não-linear e requer, no mínimo, uma rede com múltiplas camadas ocultas para traçar uma fronteira de decisão complexa. O uso de 3 camadas ocultas em todos os experimentos foi suficiente para essa tarefa. A variação no número de neurônios por camada demonstrou a importância da otimização da capacidade do modelo. O Experimento 2, com uma arquitetura mais enxuta, superou as configurações com maior número de neurônios, sugerindo que uma arquitetura eficiente pode ser mais benéfica para a generalização do que um modelo com maior capacidade computacional.

### 4.2. Função de Ativação

A escolha da função de ativação foi um fator crítico no desempenho. Enquanto a função Tanh demonstrou capacidade de resolver o problema, a ReLU (Experimento 2) foi significativamente superior. A ReLU contribui para uma convergência mais rápida e estável durante o treinamento ao mitigar o problema do "gradiente de saturação", comum em funções como a Tanh para entradas que se aproximam dos extremos.

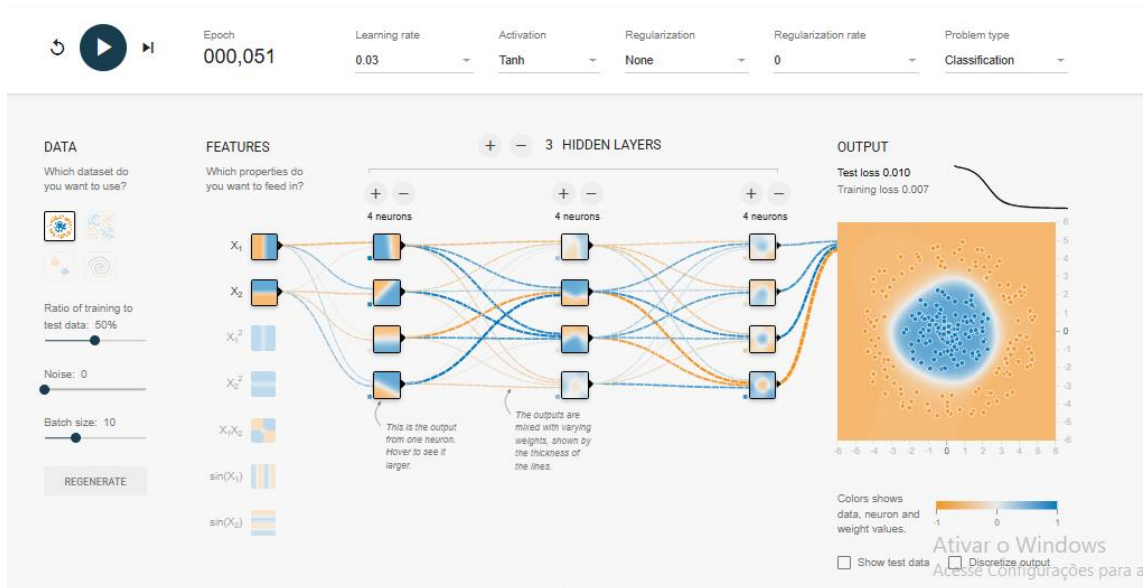
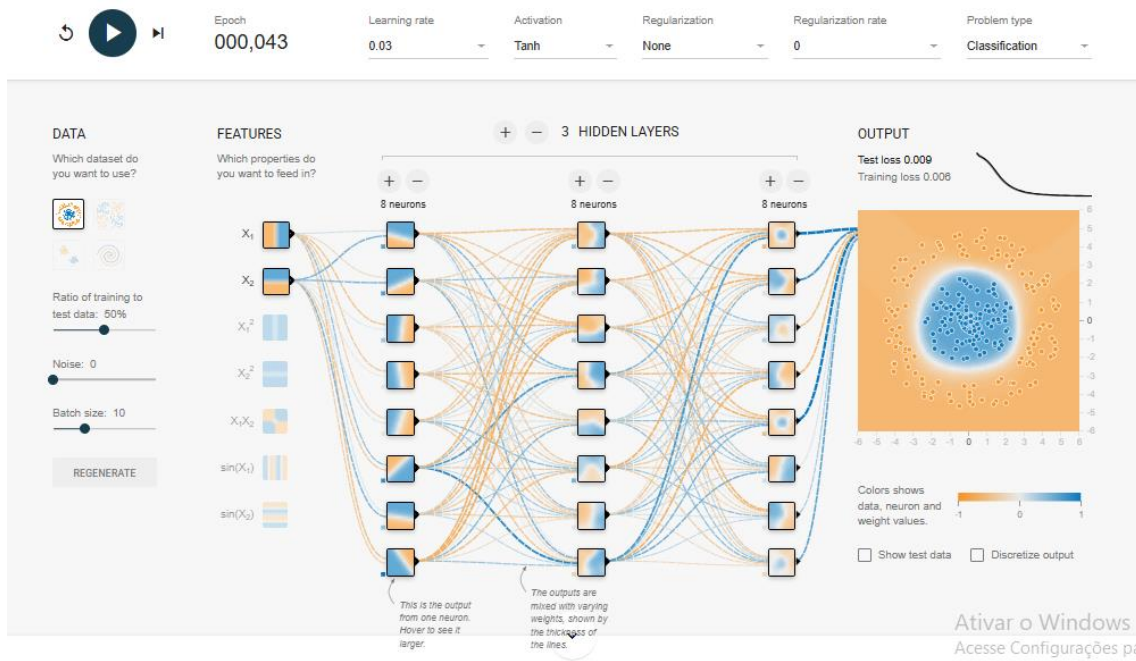
### 4.3. Parâmetros de Treinamento

A taxa de aprendizado (0.03) demonstrou ser um valor adequado para a convergência de todos os modelos, evitando oscilações no processo de otimização. A regularização (None) não foi necessária, uma vez que não foram observados sinais de *overfitting* (a perda de teste permaneceu baixa e em linha com a perda de treinamento).

## 5. Conclusão

A avaliação comparativa dos experimentos demonstrou que a combinação de uma **função de ativação ReLU** com uma **arquitetura de rede otimizada (4-2-2)** resultou na performance superior para a classificação do conjunto de dados Espiral. O estudo conclui que a seleção criteriosa de hiperparâmetros e a otimização da arquitetura são fatores mais decisivos para o sucesso de um modelo do que simplesmente o aumento da sua complexidade.

## 6. Anexos



The screenshot displays the TensorFlow Playground interface. At the top, the configuration is set to Epoch 000,086, Learning rate 0.03, Activation ReLU, Regularization None, Regularization rate 0, and Problem type Classification. The interface is divided into three main sections: DATA, FEATURES, and OUTPUT.

**DATA:** The 'Which dataset do you want to use?' dropdown is set to 'Gaussian'. The 'Ratio of training to test data: 50%' slider is at 50%. The 'Noise: 0' slider is at 0. The 'Batch size: 10' slider is at 10. A 'REGENERATE' button is present.

**FEATURES:** The 'Which properties do you want to feed in?' dropdown is set to '3 HIDDEN LAYERS'. The network structure shows 4 input neurons ( $X_1, X_2, X_1^2, X_2^2$ ), 2 hidden neurons, and 2 output neurons. The output neurons are labeled  $\sin(X_1)$  and  $\sin(X_2)$ . A tooltip indicates: 'The outputs are mixed with varying weights, shown by the thickness of the lines.' Another tooltip points to the output neurons: 'This is the output from one neuron. Hover to see it larger.'

**OUTPUT:** The 'Test loss 0.005' and 'Training loss 0.004' are displayed. A 2D scatter plot shows the data points (orange dots) and the learned decision boundary (blue lines). A color bar at the bottom indicates the values of the data, neuron, and weight outputs.