



## Relatório – Atividade Aula 4.3 (TensorFlow Playground – Dataset Spiral)

Neste exercício utilizei o ambiente interativo TensorFlow Playground para explorar o impacto dos principais hiperparâmetros de uma rede neural treinada para classificar o dataset Spiral usando somente duas entradas ( $x_1$  e  $x_2$ ). O dataset Spiral é particularmente desafiador, pois apresenta forte não-linearidade e exige modelos mais complexos para a separação das classes.

### 1. Número de camadas

Ao testar diferentes profundidades, observei que:

Redes com 1 ou 2 camadas não conseguiam aprender o padrão em espiral e apresentavam Test Loss acima de 0.3.

A partir de 3 camadas ocultas, o modelo passou a ser capaz de capturar as curvas do padrão, reduzindo significativamente o erro.

Conclusão: Mais camadas aumentam a capacidade de representar funções altamente não lineares, o que é essencial para o dataset Spiral.

## 2. Número de neurônios por camada

Testei entre 2 e 12 neurônios por camada:

Poucos neurônios ( $< 6$ ) não davam capacidade suficiente ao modelo.

Muitos neurônios ( $> 12$ ) causavam sobreajuste e instabilidade.

Com 8 neurônios por camada, obtive o melhor equilíbrio entre capacidade e generalização.

Conclusão: Mais neurônios aumentam a expressividade, mas acima de certo ponto o ganho desaparece e surge overfitting.

## 3. Função de ativação

Comparei relu, sigmoid e tanh:

ReLU não performou bem no Spiral, pois não modela curvas com a mesma suavidade.

Sigmoid funcionou, porém treinou lentamente.

Tanh obteve os melhores resultados, ajustando naturalmente regiões curvas.

Conclusão: A escolha da ativação influencia diretamente a habilidade do modelo em aprender padrões curvos. tanh foi a mais adequada.

## 4. Learning rate

Testei valores de 0.001 a 0.3:

Valores baixos (0.001) treinavam muito lentamente e ficavam presos em mínimos locais.

Valores altos ( $> 0.1$ ) causavam instabilidade.

O melhor equilíbrio ocorreu em 0.03, com convergência rápida e suave.

Conclusão: O learning rate é determinante para a velocidade e qualidade da convergência.

## 5. Regularização

Sem regularização, o modelo memorizava parte dos dados e apresentava flutuação no Test Loss.

Com  $L2 = 0.001$ , o treinamento tornou-se mais estável e a generalização melhorou.

Conclusão: A regularização L2 ajuda a evitar overfitting e estabiliza o comportamento da rede.

Resultado final

Usando:

3 camadas ocultas

8 neurônios por camada

Ativação: tanh

Learning rate: 0.03

Regularização L2: 0.001

Obtive:

Test Loss < 0.1 (0.034)