



# Atividade Playground

Bruna Alves

# Playground

## Convergência > Circle

### Configurações

**Learning rate:** 0.03

Um valor relativamente alto, mas aparentemente estável para esse problema simples. Nesse sentido, concluímos que este valor acelerou a convergência.

**Activation:** Tanh

É um modelo de ativação recomendado para problemas com fronteiras de decisão suaves. A saída varia de -1 a 1, o que permitiu capturar não linearidades.

**Regularization:** None

Não há penalização de pesos — o dataset é simples e não há sinal de overfitting (train loss  $\approx$  test loss).

**Regularization rate:** 0

Confirma que não há regularização aplicada.

**Problem type:** Classification

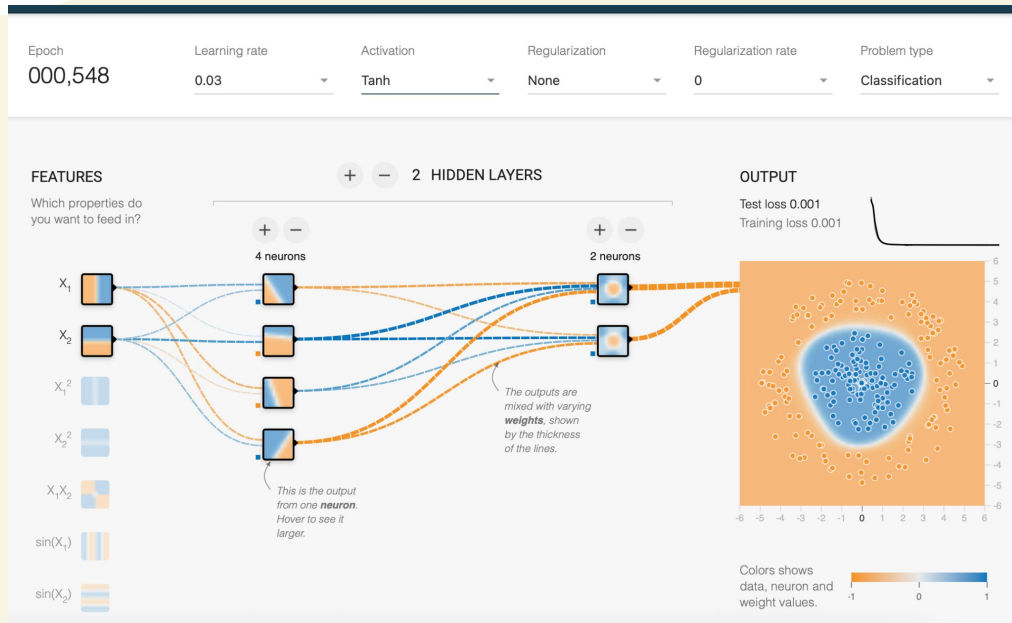
Classificação binária (área azul vs. laranja no gráfico final).

### Arquitetura

- **Entradas:**  $X_1$  e  $X_2$  (sem features polinomiais ou funções trigonométricas extras ativas).
- **2 camadas ocultas:**
  - 1ª camada: **4 neurônios**
  - 2ª camada: **2 neurônios**
- As conexões têm espessura variável, indicando diferentes pesos aprendidos.
- As cores azul/laranja representam valores negativos/positivos das ativações e pesos.

# Playground

## Convergência > Circle



## Resultados

- **Training loss:** 0.001
- **Test loss:** 0.001  
Generalização desejável, sem diferença significativa entre treino e teste.
- **Fronteira de decisão:**  
O modelo aprendeu um limite circular suave, separando claramente a classe central (azul) da externa (laranja).
- **Epoch: 548**  
A rede já passou por várias iterações, suficiente para convergir — o gráfico de perda apresenta-se estável pois alcançou o valor de 0.001

## Interpretação

- O problema parece ser separável por uma fronteira circular — a rede com **tanh** pareceu ter sido uma ótima escolha para capturar este resultado
- Não há sinais de overfitting, então a ausência de regularização não é um problema.
- O learning rate de 0.03 funcionou nesse caso.
- A cor e espessura das linhas tracejadas indicam que o resultado poderia ter sido alcançado com menos neurônios na primeira camada.

# Playground

## Convergência > Circle

Resultado alcançado com menos épocas

### Arquitetura

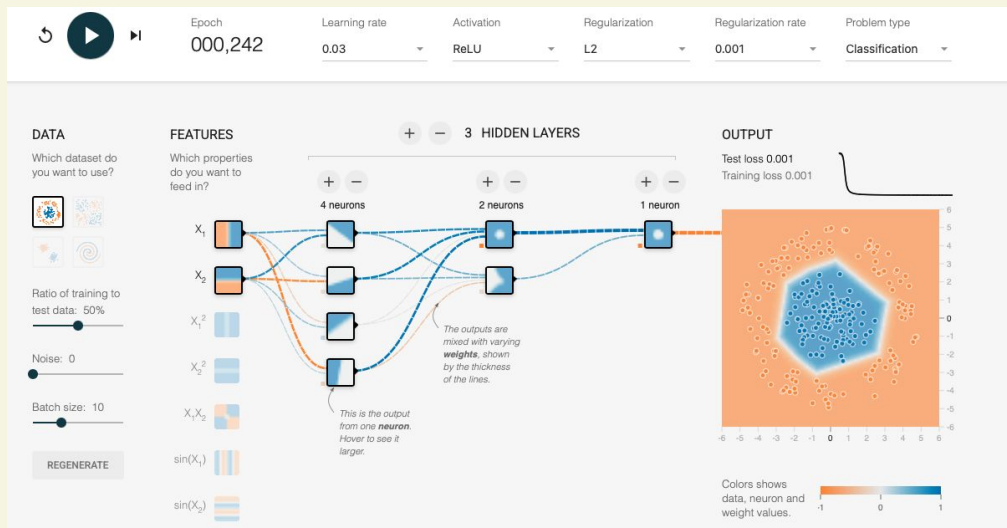
- **Entradas:**  $X_1$  e  $X_2$  (sem features polinomiais ou funções trigonométricas extras ativas).
- **Activation:** ReLU
- **3 camadas ocultas:**
  - 1ª camada: 4 neurônios
  - 2ª camada: 2 neurônios
  - 3ª camada: 1 neurônio

### Resultados

- **Test loss:** 0.001
- **Training loss:** 0.001
- Ambos os valores de loss são muito baixos, indicando que o modelo está treinando bem e não parece estar sofrendo de overfitting (pois o loss de teste acompanha o de treino).
- **Epoch: 242**  
A rede já passou por várias iterações, suficiente para convergir em menos tempo

### Interpretação

- A rede está aprendendo a separar as duas classes muito bem, como indicado pela divisão clara da região azul e laranja no gráfico de saída.
- A regularização L2 está sendo usada para evitar overfitting, e o resultado sugere que isso está dentro do comportamento esperado
- O learning rate está alto, mas o modelo parece estável, talvez porque os dados sejam simples e o batch size pequeno
- O ReLU foi uma boa escolha dado que ajuda a evitar o problema de gradiente desvanecente



# Playground

## Convergência > Exclusive or

### Configurações

**Learning rate:** 0.3

Taxa de aprendizado alta, acelerando a convergência, mas que poderia ter gerado oscilações em problemas mais sensíveis.

**Activation:** Sigmoid

Adequada para classificação binária, embora possa sofrer com gradiente pequeno em redes mais profundas.

**Regularization:** None

Apesar do "Regularization rate" ser 1, isso não afeta o resultado, dado que não defini o L1 ou L2

**Problem type:** Classification

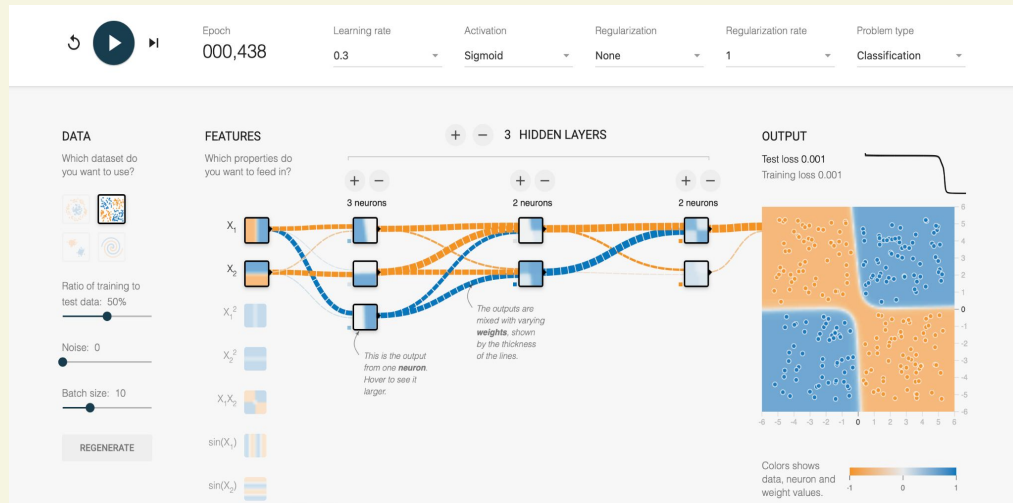
Problema de separação em regiões distintas no plano  $X_1$ – $X_2$ .

### Arquitetura

- **Entradas:**  $X_1$  e  $X_2$  apenas.
- **3 camadas ocultas:**
  - 1ª camada: **3 neurônios**
  - 2ª camada: **2 neurônios**
  - 3ª camada: **2 neurônios**
- Os pesos (espessura das linhas) indicam que algumas conexões têm muito mais influência na saída final que outras.
- As cores (azul = valores negativos, laranja = positivos) mostram que a rede aprendeu a combinar regiões com polaridades alternadas.

# Playground

## Convergência > Exclusive or



## Resultados

- **Training loss: 0.001**
- **Test loss: 0.001**
- Bom desempenho e generalização alcançados
- **Fronteira de decisão:** Padrão de **quatro quadrantes** alternando classes (azul/laranja), com limites suaves, quase lineares, formados pela combinação de neurônios sigmóides.
- **Epoch: 438** Rede treinada por menos tempo, mas já com convergência clara.

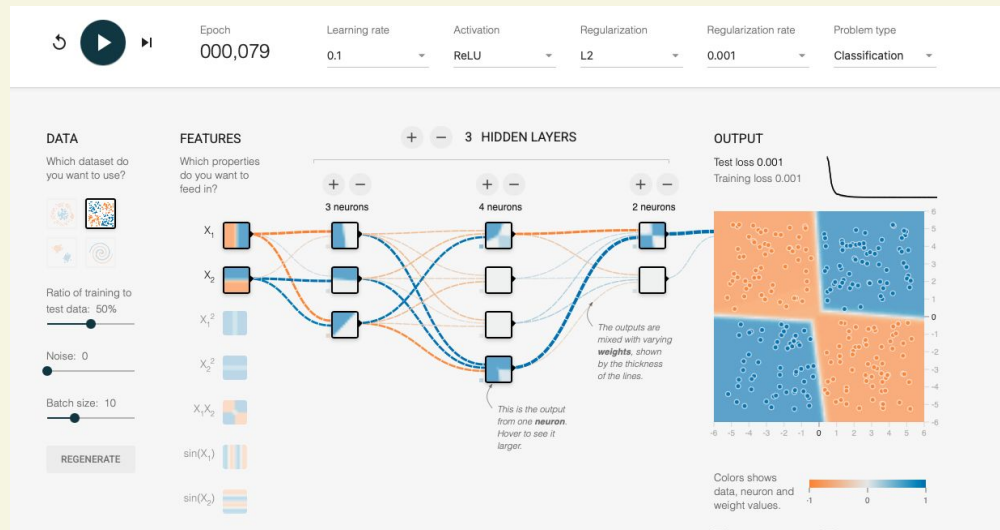
## Interpretação

- O modelo conseguiu capturar rapidamente a estrutura de "XOR contínuo" do dataset.
- A taxa de aprendizado alta (0.3) permitiu atingir um bom resultado em apenas 438 épocas.
- Não há sinais de overfitting: treino e teste com perdas idênticas.

# Playground

## Convergência > Exclusive or

Resultado alcançado com menos épocas



## Arquitetura

- Entradas:  $X_1$  e  $X_2$ .
- Activation: ReLU
- 3 camadas ocultas:
  - 1ª camada: 3 neurônios
  - 2ª camada: 4 neurônios
  - 3ª camada: 2 neurônios

## Resultados

- Training loss: 0.001
- Test loss: 0.001
- Conexões com variação significativa de espessura, mostrando importância desigual dos pesos.
- As ativações estão bem distribuídas entre azul (negativo) e laranja (positivo), típico da ReLU em padrões complexos.
- Epoch: 79  
Rede convergiu rapidamente, apesar de superdimensionada.

## Interpretação

- O uso de ReLU + L2 contribuiu para uma separação mais nítida das classes, mesmo com poucas épocas de treino.
- A regularização ajudou a evitar oscilações ou ajustes exagerados dos pesos.
- A fronteira é mais “angular” e menos suave que a vista com Sigmoid, refletindo a natureza linear por partes da ReLU.
- Pela espessura das linhas apresentadas, 2 neurônios na segunda camada e 1 na terceira teriam sido suficientes para a convergência do resultado

# Playground

## Convergência > Gaussian

### Configurações

**Learning rate:** 0.03

Tava de aprendizado é relativamente baixa, o que significa que o modelo atualiza seus pesos de forma gradual, evitando mudanças bruscas. Contudo, reduz o risco de divergências e oscilações

**Activation:** ReLU

Utilizada para evitar saturação e acelerar o treino, contudo, é menos suave na transição que sigmoide/tanh.

**Regularization:** 0.001

Prevenção de overfitting penalizando pesos grandes, mas de forma leve

**Problem type:** Classification

Problema de separação em regiões distintas no plano  $X_1$ – $X_2$ .

### Arquitetura

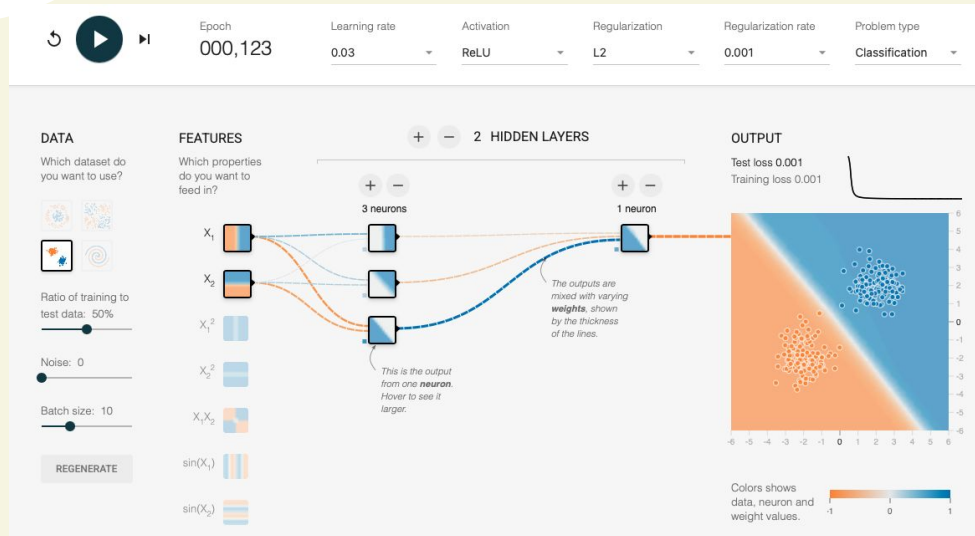
- **Entradas:**  $X_1$  e  $X_2$  apenas.
- **2 camadas ocultas:**
  - 1ª camada: **3 neurônios**
  - 2ª camada: **1 neurônios**

**Saída:** 1 neurônio, compatível com problema binário (classificação).



# Playground

## Convergência > Gaussian



## Resultados

- **Training loss:** 0.001
- **Test loss:** 0.001
- Bom desempenho e generalização alcançados
- **Fronteira de decisão:** Quase perfeitamente linear, separando claramente as duas classes.
- **Epoch:** 123
- Rede treinada por menos tempo, mas já com convergência clara.

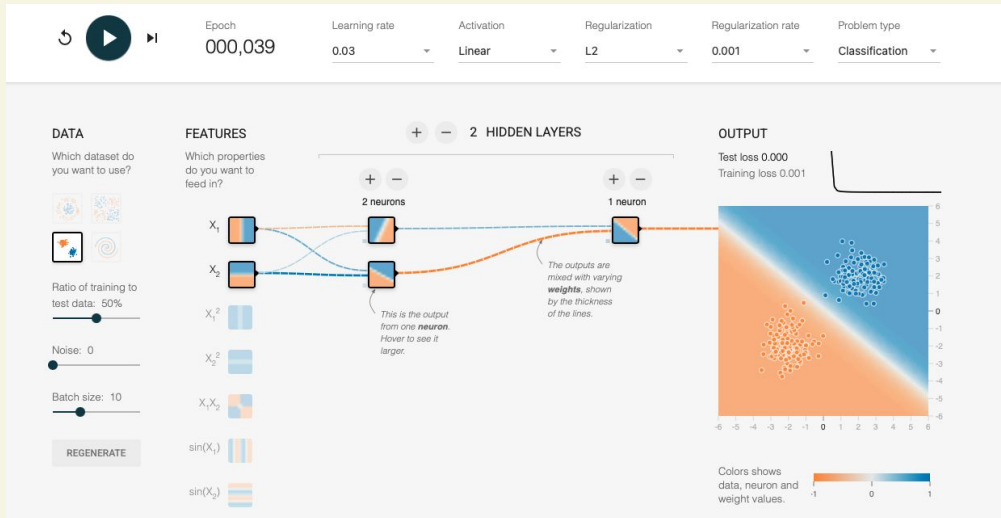
## Interpretação

- A rede está levemente superdimensionada para este problema, que é linearmente separável
- O uso de 2 camadas ocultas é mais do que o necessário, mas não prejudicou o desempenho por conta da regularização e do dataset limpo.
- O batch pequeno ajudou a refinar rapidamente a fronteira de decisão.

# Playground

## Convergência > Gaussian

Resultado alcançado com menos épocas



## Arquitetura

- Entradas:  $X_1$  e  $X_2$ .
- Activation: Linear
- 2 camadas ocultas:
  - 1ª camada: 2 neurônios
  - 2ª camada: 1 neurônio

## Resultados

- Training loss: 0.001
- Test loss: 0.000
- Ativação Linear, o que transforma toda a rede em um modelo linear, mesmo com múltiplas camadas.
- Epoch: 39  
Rede convergiu rapidamente e apresentou por test loss 0.000, o que representa uma excelente generalização

## Interpretação

- Como o problema é linearmente separável, a escolha de ativação Linear não prejudicou o desempenho (apesar de em problemas mais complexos ser preferível usar funções não lineares como ReLU ou Tanh).
- O uso de apenas 3 neurônios ocultos foi suficiente — mais neurônios ou camadas não trariam ganhos relevantes nesse caso.
- O L2 leve ajudou a evitar ajustes exagerados, mantendo a fronteira de decisão simples e robusta.



C.E.S.A.R  
school

# Treinamento Spiral



## Configurações

**Learning rate:** 0.03

Tava de aprendizado é relativamente baixa, o que significa que o modelo atualiza seus pesos de forma gradual, evitando mudanças bruscas. Contudo, reduz o risco de divergências e oscilações

**Activation:** ReLU

Adequada para classificação binária, embora possa sofrer com gradiente pequeno em redes mais profundas.

**Regularization:** L2

Prevenção de overfitting penalizando pesos grandes, mas de forma leve

**Problem type:** Classification

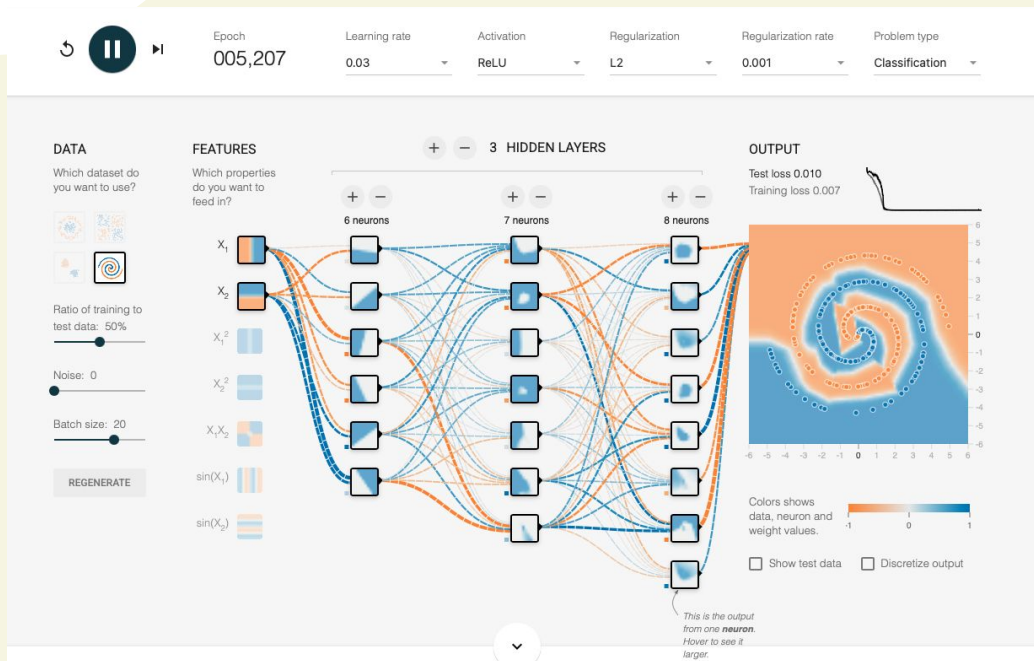
Problema de separação em regiões distintas no plano  $X_1$ – $X_2$ .

## Arquitetura

- **Entradas:**  $X_1$  e  $X_2$  apenas.
- **3 camadas ocultas:**
  - 1ª camada: **6 neurônios**
  - 2ª camada: **7 neurônios**
  - 3ª camada: **8 neurônios**
- **Conexões:** Totalmente conectadas, com pesos positivos (laranja) e negativos (azul).
- **Saída:** 1 camada com classificação binária (espiral laranja vs azul).

# Playground

## Convergência > Spiral



## Resultados

- **Training loss:** 0.007
  - **Test loss:** 0.001
- Bom desempenho e generalização alcançados
- **Epoch:** 5207
- Por apresentar uma maior complexidade, o resultado demorou um pouco mais para ser alcançado. O learning rate de 0.03 pode ter influenciado, dado que o aprendizado se dá de forma gradual para evitar mudanças bruscas

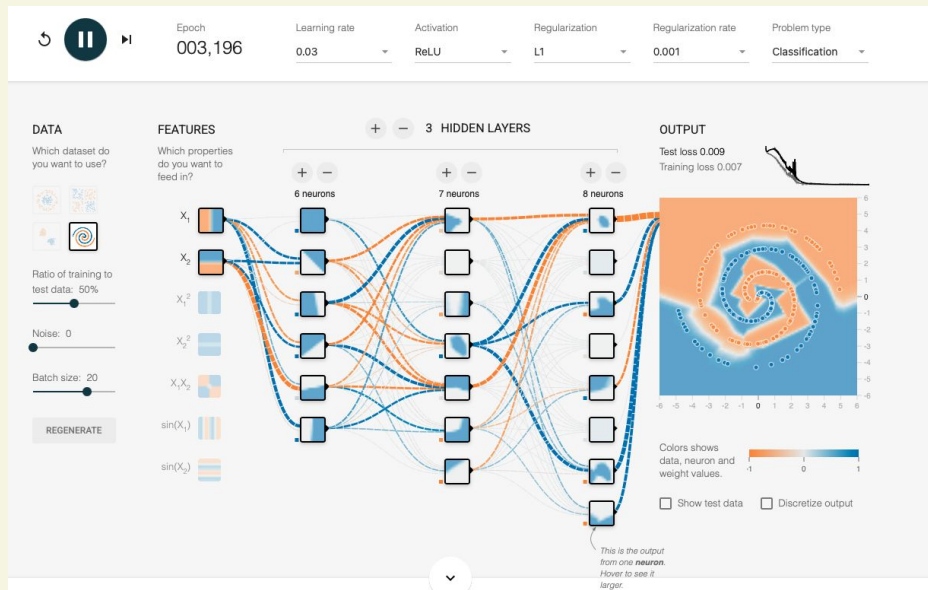
## Interpretação

- A rede foi bem configurada para o problema em espiral
- O modelo está com alta capacidade de generalização, mantendo baixo erro em treino e teste.
- A arquitetura com múltiplas camadas + **ReLU** foi essencial para capturar as relações não lineares dos dados.
- Generalização muito boa, sem overfitting aparente
- A fronteira de decisão (área azul vs laranja) segue com precisão o formato em espiral dos dados.
- A rede conseguiu aprender relações altamente não lineares graças ao uso do **ReLU**, que permite expressividade não linear e ao número maior de neurônios e camadas, que aumenta a capacidade de modelagem.

# Playground

## Convergência > Spiral

## Resultado alcançado com menos épocas



## Arquitetura

- **Entradas:** X<sub>1</sub> e X<sub>2</sub>.
- **Regularization:** L1 com regularization rate de 0.001
- **3 camadas ocultas:**
  - 1ª camada: 6 neurônios
  - 2ª camada: 7 neurônios
  - 3ª camada: 8 neurônios



## Resultados

- **Test loss:** 0.009
- **Training loss:** 0.007
- **Epoch:** 3196
- Mais de 2000 Epoch a menos do que no experimento anterior
- O modelo conseguiu separar muito bem as duas espirais, criando fronteiras suaves e precisas.
- As áreas azul e laranja seguem o formato espiral, respeitando a geometria dos dados.
- Perdas muito próximas, sem overfitting, indicando que a regularização L1 foi eficaz.

## Interpretação

- O uso de **L1** (ao invés de L2) fez o modelo selecionar conexões mais relevantes, simplificando a rede sem perder precisão.
- A performance pode ser considerada excelente uma vez que apresentou: **baixa perda de treino e teste, e fronteiras de decisão bem ajustadas.**
- Comparando com o experimento anterior (que usava L2), esta configuração pode gerar uma rede mais enxuta e interpretável, mantendo a generalização.
- É possível que o mesmo resultado pudesse ter sido alcançado com menos neurônios da segunda e terceira camada, dado que alguns deles não apresentaram conexões relevantes para treinamento do modelo.

# Playground

## Tabela comparativa - Convergência Spiral

Aspecto	Treinamento 1 (L2, $\lambda=0.001$ )	Treinamento 2 (L1, $\lambda=0.001$ )
Regularização	Penaliza quadrado dos pesos	Penaliza valor absoluto dos pesos
Efeito nos Pesos	Pesos pequenos, mas não zero.	Sparsidade: pesos irrelevantes $\rightarrow$ zero.
Número de Épocas*	5.207 (convergência mais lenta).	3.196 (convergência mais rápida).
Learning Rate (0.03)	Estável, mas pode exigir mais épocas.	Mais eficiente em reduzir pesos irrelevantes.
Training Loss	0.007	0.007
Test Loss	0.01	0.009
Fronteira de Decisão	Suave e precisa.	Igualmente precisa, mas com menos pesos ativos.
Overfitting	Controlado (L2 suave).	Melhor controle (L1 elimina ruído).
Capacidade de Interpretação	Menos interpretável (todos os pesos ativos).	Mais interpretável (pesos zerados = features menos importantes)

### \* Número de Épocas

- Treinamento 1 (5.207 épocas):
  - A alta quantidade de épocas garante convergência completa, mas consome mais recursos.
  - L2 beneficia-se de treinamento prolongado para ajustar pesos suavemente.
- Treinamento 2 (3.196 épocas):
  - L1 converge mais rápido porque "mata" pesos irrelevantes cedo.
  - Menor quantidade de épocas pode economizar tempo computacional.

## Pontos fortes de cada configuração

### Treinamento 1 (L2)

- **Estabilidade:** Ideal para problemas onde todas as features são relevantes.
- **Generalização:** Bom equilíbrio entre viés e variância.
- **Suavidade:** Fronteiras de decisão mais contínuas (útil para dados não lineares complexos).

### Treinamento 2 (L1)

- **Simplicidade:** Cria modelos **esparsos** (ótimo para seleção de features).
- **Eficiência:** Convergiu em **menos épocas** que L2 (3.196 vs. 5.207).
- **Robustez:** Remove ruído e conexões irrelevantes automaticamente.

## Pontos de atenção de cada configuração

### Treinamento 1 (L2)

- **Pesos redundantes:** Pode manter pesos pequenos sem utilidade prática.
- **Épocas:** Exige mais tempo de treinamento para convergir.

### Treinamento 2 (L1)

- **Underfitting:** Se  $\lambda$  for muito alto, o modelo pode ficar **muito simples**.
- **Instabilidade:** Em alguns casos, a esparsidade pode eliminar pesos importantes.



## Playground - Considerações finais

### Convergência > **Spiral**

- Ambos os modelos performaram muito bem no dataset complexo (espiral). No entanto, é possível inferir que: **L2** é mais "conservador" e indicado para problemas onde todas as features importam. **L1** é mais "agressivo" e útil para simplificação e interpretabilidade.
- Se o objetivo é saber quais **inputs** são **mais relevantes**, L1 é a melhor escolha. Se o foco é **precisão pura** em dados não lineares, L2 pode ser mais seguro.