



Epoch  
000,316

Learning rate  
0.1

Activation  
ReLU

Regularization  
L2

Regularization rate  
0

Problem type  
Classification

#### DATA

Which dataset do you want to use?



Ratio of training to test data: 80%

Noise: 0

Batch size: 15

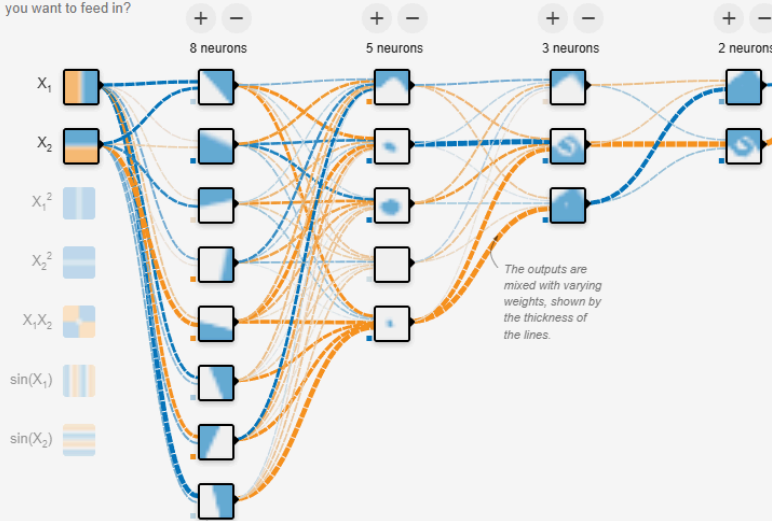
REGENERATE

#### FEATURES

Which properties do you want to feed in?

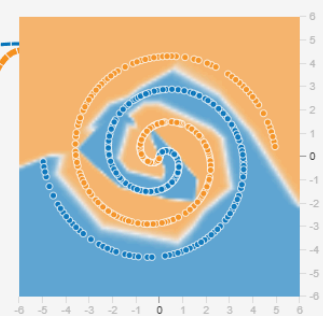
$X_1$   
 $X_2$   
 $X_1^2$   
 $X_2^2$   
 $X_1X_2$   
 $\sin(X_1)$   
 $\sin(X_2)$

#### 4 HIDDEN LAYERS



#### OUTPUT

Test loss 0.004  
Training loss 0.005



Colors shows data, neuron and weight values.

☐ Show test data ☐ Discretize output



Epoch  
000,600

Learning rate  
0.03

Activation  
ReLU

Regularization  
None

Regularization rate  
0

Problem type  
Classification

#### DATA

Which dataset do you want to use?



Ratio of training to test data: 70%

Noise: 0

Batch size: 10

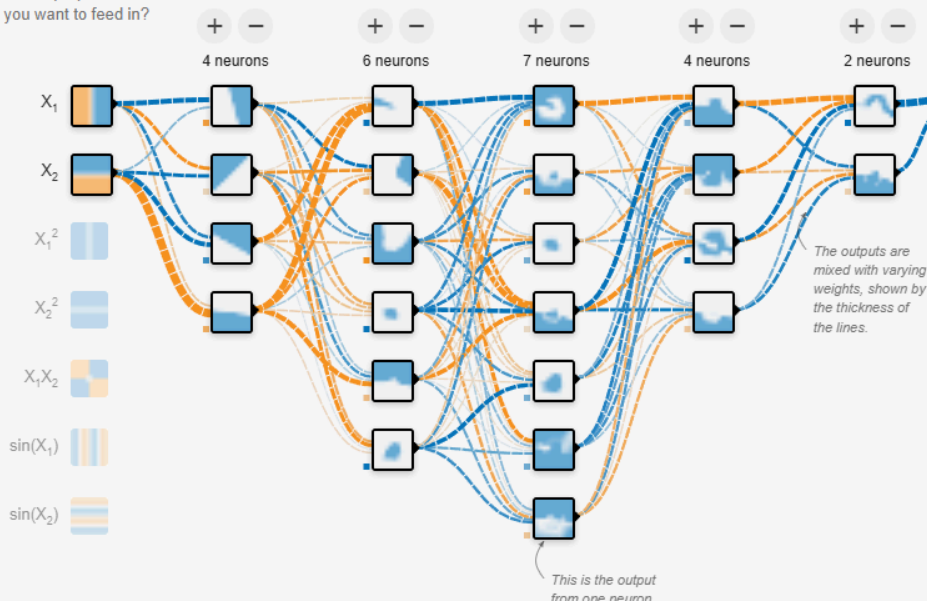
REGENERATE

#### FEATURES

Which properties do you want to feed in?

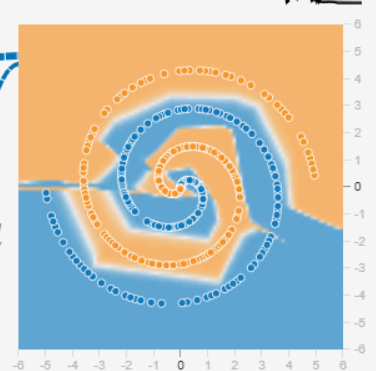
$X_1$   
 $X_2$   
 $X_1^2$   
 $X_2^2$   
 $X_1X_2$   
 $\sin(X_1)$   
 $\sin(X_2)$

#### 5 HIDDEN LAYERS



#### OUTPUT

Test loss 0.001  
Training loss 0.015



Colors shows data, neuron and weight values.

☐ Show test data ☐ Discretize output

# Atividade Playground

## Dataset Spiral Resolution

Angelina Maria

Vamos analisar como a configuração dos parâmetros antes do processo de treinamento influenciou nas características de aprendizagem do modelo.

A análise terá foco nos elementos de arquitetura da rede, função de ativação, taxa de aprendizado e técnicas de regularização, elementos estes que serão utilizados como base de um estudo de caso comparativo entre dois modelos de rede neural (denominados M1 e M2), ambos foram treinados para a mesma tarefa de classificação binária, partindo das mesmas features de entrada, porém com configurações distintas que levaram a resultados e eficiências de treinamento diferentes.

A ideia de analisar dois modelos surgiu depois de concluir o M1 e perceber que o poderia ser feito melhor, já que o exercício foi proposto antes da aula de fundamentação dos elementos de configuração. Com isso, após adquirir mais conhecimento, refiz o exercício resultante no M2 e este mostrou-se mais eficiente.

## Apresentação dos Modelos e Resultados Consolidados

Característica	M1	M2
Learning Rate	0.1	0.03
Activation Function	ReLU	ReLU
Regularization	L2	NaN
Hidden Layers	3 layers (8 → 5 → 3 neurons)	4 layers (4 → 6 → 7 → 4 neurons)
Epoch by Convergence	316	600
Training Loss	0.005	0.015
Test Loss	0.004	0.001

A arquitetura, definida pelo número de *layers* e *neurons*, determina a capacidade de representação do modelo.

- O M1 adotou uma arquitetura em "funil" (8 → 5 → 3), que força uma compressão progressiva da informação. Essa abordagem é eficiente para extrair as características mais relevantes de forma direta.
- O M2 utilizou uma arquitetura mais profunda e complexa, em "diamante" (4 → 6 → 7 → 4). Tal estrutura permite uma expansão da dimensionalidade dos dados nas camadas intermediárias, possibilitando o aprendizado de interações mais complexas entre as features antes da consolidação final.

A *learning rate* controla a magnitude dos ajustes nos pesos do modelo a cada iteração.

- O M1, com uma *learning rate* alta de 0.1, apresentou uma convergência rápida (316 épocas). Essa estratégia, embora eficiente, acarreta o risco de não encontrar o mínimo ótimo exato (*overshooting*).
- O M2, com um valor mais moderado de 0.03, necessitou de mais épocas (600) para convergir, realizando uma busca mais lenta e refinada pelo ponto de menor erro.

A *regularization* é uma técnica para mitigar o *overfitting*, garantindo que o modelo generalize bem para dados não vistos.

- O M1 empregou a regularização L2. O sucesso da técnica é evidenciado pela proximidade entre o *training loss* (0.005) e o de *test loss* (0.004), um indicador clássico de boa generalização.
- O M2, por outro lado, alcançou uma generalização excepcional (*Test Loss* < *Training Loss*) sem o uso de regularização explícita.

Ambos os modelos utilizaram a *Activation Function* ReLU, um padrão na construção de redes neurais modernas devido à sua eficiência computacional e sua capacidade de atenuar o problema do desaparecimento do gradiente. O sucesso consistente em ambas as arquiteturas valida a sua adequação para este problema.

- Sob o critério de precisão, o M2 é superior, atingindo um erro de teste significativamente menor.
- Sob o critério de eficiência (velocidade e custo computacional), o M1 é o vencedor claro, convergindo em quase metade do tempo.

## Conclusão

Este estudo demonstrou, através de um caso prático, como diferentes hiperparâmetros influenciam diretamente a performance, a generalização e a eficiência de redes neurais. Foi observado que uma arquitetura mais complexa pode levar a uma maior acurácia, e que a taxa de aprendizado governa um trade-off direto entre velocidade e precisão. Notavelmente, a análise também revelou que um modelo bem configurado pode alcançar excelente generalização sem o uso de técnicas de regularização explícitas.

Conclui-se que não existe uma configuração universalmente ótima. A seleção de hiperparâmetros é um processo empírico que exige experimentação e uma análise criteriosa dos resultados, sempre considerando os objetivos e as restrições do problema em questão.

## Fun Glossary

- *Learning Rate* é a taxa de cafeína. É a intensidade com a qual um programador vai resolver um bug.
  - Learning Rate ALTO:  
Digamos que um programador consumiu 10 xícaras de café, nesse caso quanto mais cafeína ele toma, mais rápido ele consegue concluir uma atividade, mas não necessariamente concluir da forma certa podendo piorar a situação. O erro pode oscilar ou aumentar porque ele começa a pular passos para ser mais rápido.
  - Learning Rate BAIXO:  
Dessa vez, o programador preferiu tomar chá de camomila, então ele observa o bug, analisa detalhadamente, e toma pequenas decisões para a resolução. Seus passos de resolução são menores e pode demorar muito mais tempo para resolver o problema, mas será mais assertivo.
- *Activation Function* é a personalidade do funcionário. É o quanto um colega de trabalho filtra uma fofoca para decidir se vale passar para a frente ou não.
  - *ReLU* é otimista sincero: Se a notícia for boa ou positiva ele passa para frente, se for ruim, é melhor não perder tempo com isso.
  - *Sigmoid* é dramático binário: Ele pega toda a informação e espreme em uma fofoca em SIM ou NÃO, é bom ou ruim, é tragédia ou comédia... 0 ou 1.
  - *Linear* é o fofoqueiro sem filtro: "Vou te contar tim-tim por tim-tim..." Ele não muda nada na história e só repassa a fofoca. Por se dedicar tanto à fofoca, não consegue executar atividades muito complexas.
  - *Tanh* é sommelier de fofoca, o crítico: Ele avalia a notícia numa escala entre -1 (desperdício de tempo) e 1 (bombástica) considerando o 0 (irrelevante). Ele é muito expressivo e ótimo para discussões internas.
- *Regularization* é da auditoria interna. É a técnica para impedir que o programador resolva bug sempre com gambiarra.
  - L1 é caça às bruxas: Se o programador não segue o regulamento, ele tem seu salário zerado (o peso do neurônio) e é definitivamente demitido.
  - L2 é microgerenciador: Se o programador não segue o regulamento, ele não vai ser demitido, mas terá seu bônus cortado, porém nessa equipe não cabe estrelas então ele também diminui o bônus dos outros

---

funcionários e os força a trabalhar em conjunto sem ninguém se destacar mais que alguém.

- *Regularization Rate* é o nível da bronca que o chefe dá em cima da auditoria interna.
  - *Rating* ALTO significa que a bronca é tão forte que ninguém da equipe tem coragem de fazer mais nada, o medo de errar paralisa todo mundo e o resultado final é simplório demais (*underfitting*).
  - *Rating* BAIXO significa que a bronca foi tão leve que ninguém levou a sério e todos voltarão a fazer gambiarras assim que o chefe virar as costas (*overfitting*).
- *Problem Type* é a descrição das tarefas do estagiário. É dizer para o estagiário o que ele foi contratado para fazer.
  - *Classification* é o Etiquetador: Ele vai identificar as coisas e colocar uma etiqueta em cada uma delas. Identifique e rotule Ex.: “Isso é um gato”, “Isso é uma mão”.
  - *Regression* é o Palpiteiro: Ele olha para as coisas e sugere um número. Identifique e preveja Ex.: “Amanhã fará 25 graus”, “Essa casa custa 500k”.

