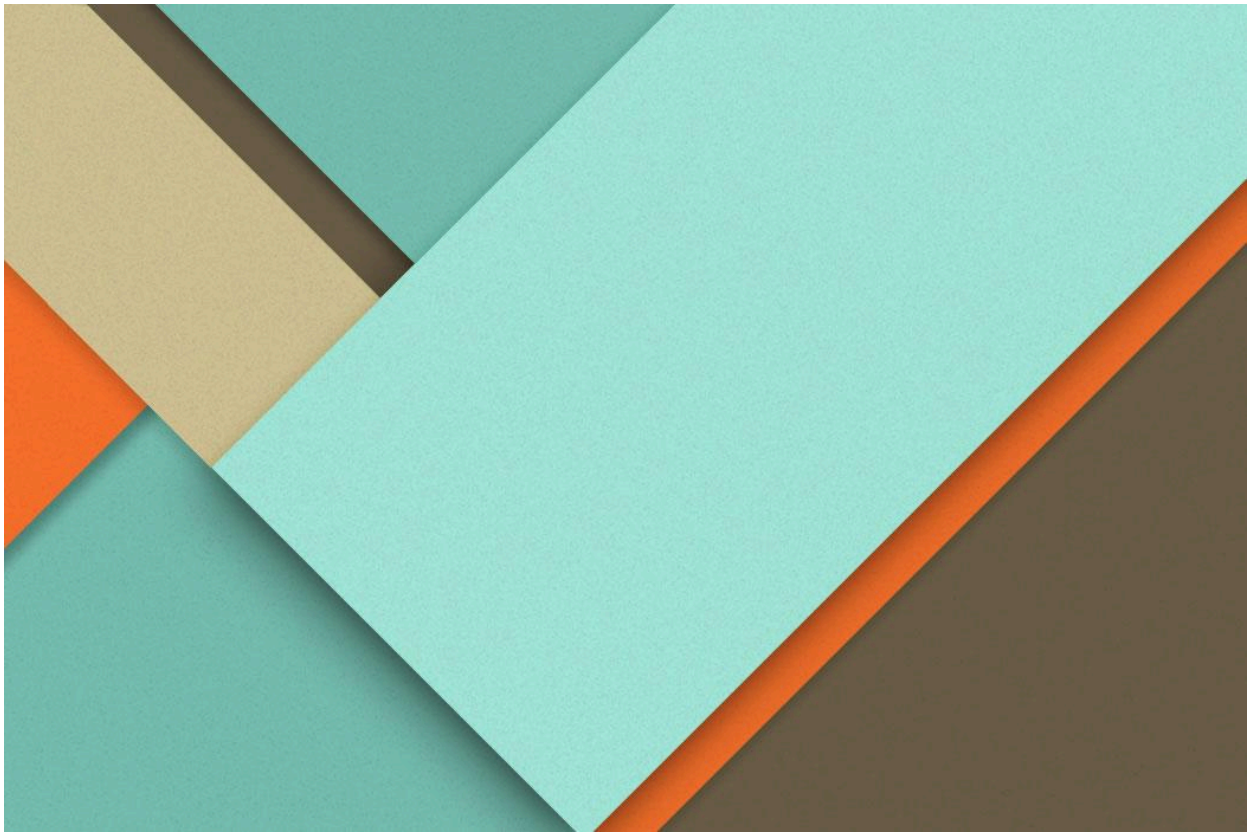

Atividade Playground

Tech Lead - Data Science e IA

—

Rafael Winter
rwt@cesar.school

Turma 2024.02

Introdução

Neste experimento foram testadas diversas configurações no *site playground* playground.tensorflow.org, com o objetivo de se obter uma rede neural otimizada para resolver o *dataset* "Spiral", alterando os seguintes parâmetros:

- Número de camadas
- Número de neurônios por camada
- Função de ativação
- *Learning Rate*
- Fator de regularização

Só não foi permitido alterar a quantidade de entradas, que foi mantida em duas (x1 e x2).

Pesquisa

Inicialmente foi criada uma rede usando os parâmetros padrão do site, que são:

- *Learning Rate*: 0.03
- *Activation*: Tanh
- *Regularization*: None
- *Regularization rate*: 0

A rede padrão era composta por duas camadas com 4 e 2 *neurons*, respectivamente. A rede nunca conseguiu convergir, e o gráfico de relação entre *Test* e *Training loss* indicava uma situação leve de overfitting, pois o *Training loss* era ligeiramente menor do que o *Test loss.*

A primeira tentativa foi adicionar mais camadas e neurônios. Adicionando o número máximo de camadas permitido pelo site, seis, e o número máximo de neurônios por camada, oito, ou seja, 48 neurônios. Com essa mudança, o modelo convergia entre 650 e 1800 épocas.

Apesar de atingir a solução, a quantidade de neurônios da rede é muito alta, o que acarretaria um alto custo computacional numa situação real. Para isso foram feitas otimizações inicialmente diminuindo a quantidade de camadas ocultas da rede.

Reduzir a quantidade de camadas para quatro, com 8 neurônios cada, ocasionou uma redução sutil na quantidade de épocas requeridas para solucionar o problema. Entretanto, reduzir para três camadas acarretou num aumento na quantidade de épocas necessárias. A partir deste ponto foram testadas outras funções de ativação.

As funções *Linear* e *Sigmoid* não conseguiram convergir, e os valores de *Test* e *Training loss* permaneciam estáveis ou não variaram o suficiente após 5000 épocas. Portanto estas funções não permitem solucionar o *dataset* Spiral e foi testada a função *ReLU*.

Refinamento

A primeira mudança para refinar a rede foi ajustar ainda mais a quantidade de neurônios de cada camada. Foram testadas diversas quantidades de neurônios por camada, e o layout de 6/8/6 mostrou uma redução da quantidade de épocas necessárias no treinamento. Em alguns cenários foi visto que camadas com quatro neurônios não eram suficientes para obter a convergência em um tempo de teste satisfatório.

Em seguida foi calibrado o valor de Learning Rate, onde uma taxa maior permite o ajuste de pesos mais rápido, mas com o risco de instabilidade. Durante os testes o valor foi aumentado para 0.1, com bom desempenho e estabilidade.

Por último foi ajustada a regularização, pois a ausência dela pode causar o crescimento de pesos descontrolados no treinamento. Foram testadas as funções L1 e L2, e com a função L2 com taxa 0.001 foi observado o ajuste de pesos menos errático, o que levou a convergência entre 150 e 1100 épocas.

Solução

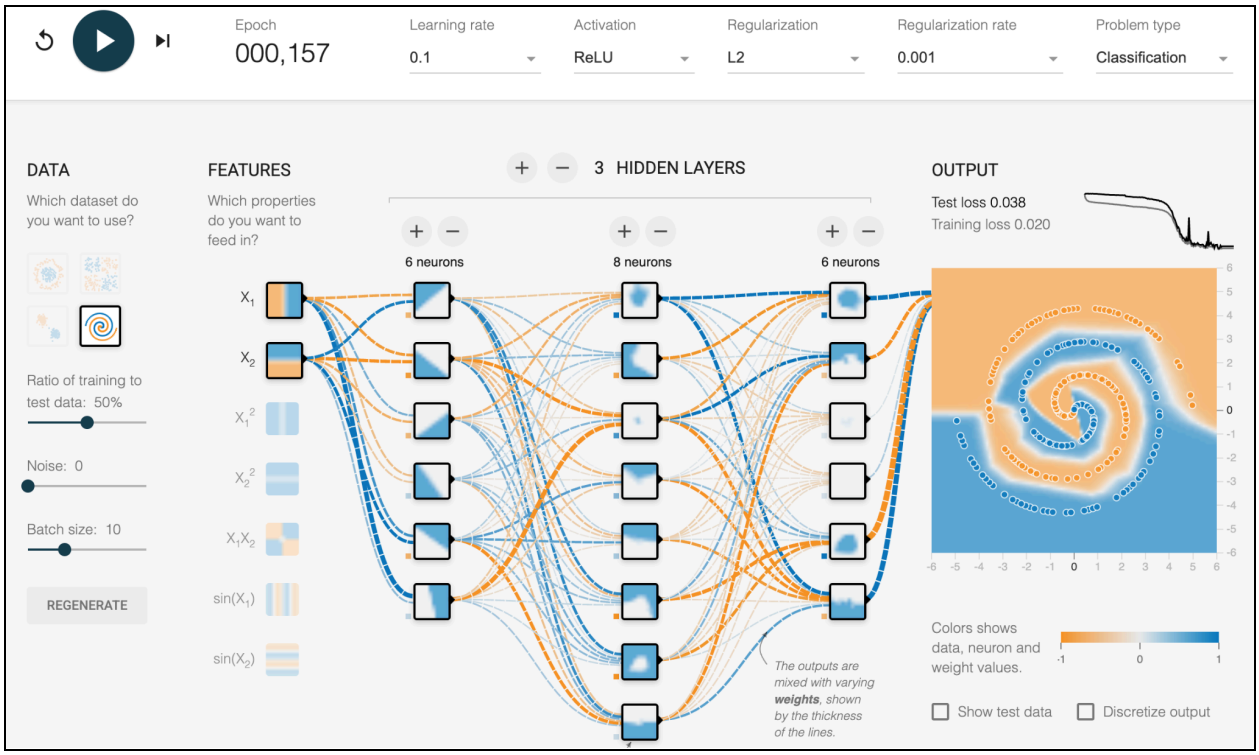

Figura 1 - Capura de tela do site playground.tensorflow.org mostrando a solução para o problema

Conclusão

O site TensorFlow Playground se mostrou um recurso muito valioso para que os principais conceitos de redes neurais fossem compreendidos mais a fundo. No site foi possível ver o impacto real que a mudança de cada parâmetro causa na performance da rede, e com isso pude experimentar diversas configurações até chegar à solução ideal.

A interface do site é prática e permitiu rapidamente testar várias configurações e alternar entre layouts da rede, observando as diferenças entre configurações como 8/6/4, 4/6/8 e 6/8/6. Além disso, o gráfico de perda permitia rapidamente detectar situações de Underfitting e Overfitting, o que ajudou bastante na comparação e calibração dos parâmetros.

Por fim, essa pesquisa mostrou a importância de se usar uma ferramenta visual e de fácil utilização com o TensorFlow Playground, pois com ela a compreensão de tópicos complexos sobre redes neurais foi facilitada. Todos os conceitos aprendidos e postos em prática nesta ferramenta podem ser diretamente aplicados em outras ferramentas de aprendizado de máquina, como Knime, ou diretamente em desenvolvimento com bibliotecas como TensorFlow e PyTorch.