

Professora Inês Dutra

Tópicos Avançados em Inteligência Artificial

Nona Entrega: Redes Neurais

Novembro de 2022

Trabalho realizado por:

Pedro Leite - 201906697

1. Exercícios do Playground

1.1. Uma Primeira Rede Neural

Apenas redes neurais que utilizam uma função de ativação linear, podem aprender não linearidade.

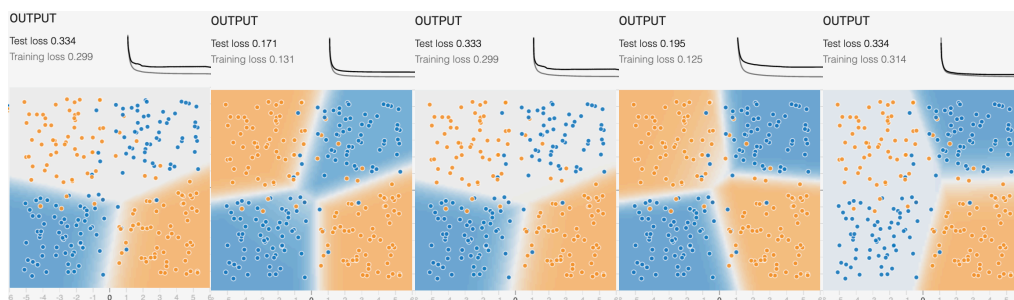
Uma rede com apenas uma camada com um único neurónio, pode apenas completar tarefas mais simples. Para fazer tarefas mais complexas, é necessário adicionar mais neurónios, que se vão interligar, para completar essas tarefas, normalmente utilizando funções de ativação não lineares. Se utilizássemos apenas um neurónio alguma das não linearidade podem ser perdidas, daí a importância de utilizar vários neurónios. Para este modelo, é necessário apenas uma camada com 3 neurónios para chegar a um bom modelo. No entanto, às vezes os testes podem variar em termos de eficácia, sendo por vezes o modelo com apenas 2 neurónios, melhor do que o com 3.

Ao aumentar o número de camadas, numa rede neural, estamos a aumentar o número de parâmetros e a complexidade da rede, e portanto a aumentar a probabilidade de termos um modelo melhor. Mas se aumentarmos demasiado, percebemos que vai ter um desempenho pior que a versão mais simples. E se a primeira camada tiver só um nó, aumentar o número de camadas não altera a eficácia do modelo.

Ao permitir que o modelo considere muitas formas diferentes por vários neurónios ocultos diferentes, o modelo começa a fazer *overfitting*. Em casos extremos, um modelo grande pode aprender uma ilha em torno de um ponto de ruído individual, o que é chamado de memorização dos dados.

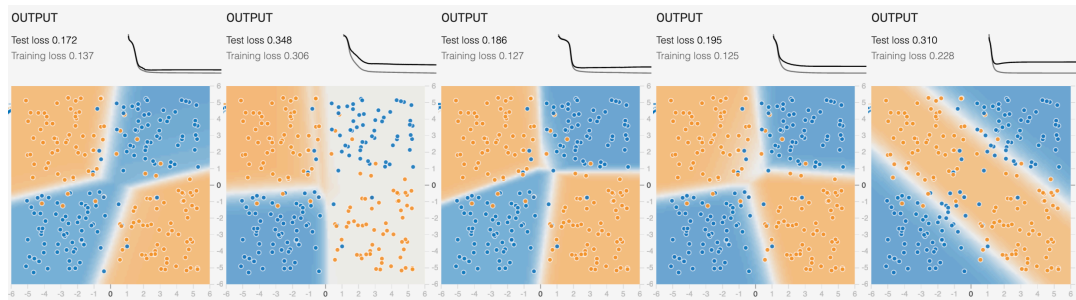
1.2. Inicialização da Rede Neural

5 execuções diferentes (500 passos):



O modelo demonstra sempre uma forma diferente após cada execução.

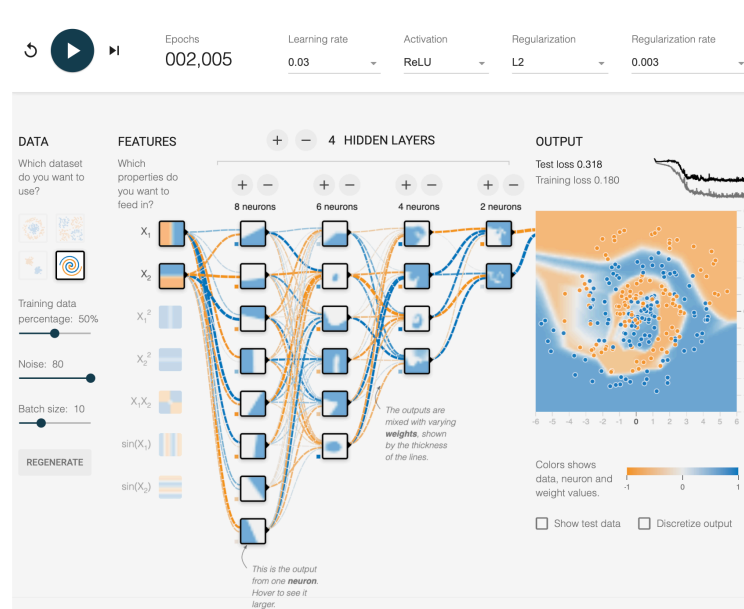
5 execuções diferentes ao adicionar algumas camadas e nós (500 passos):



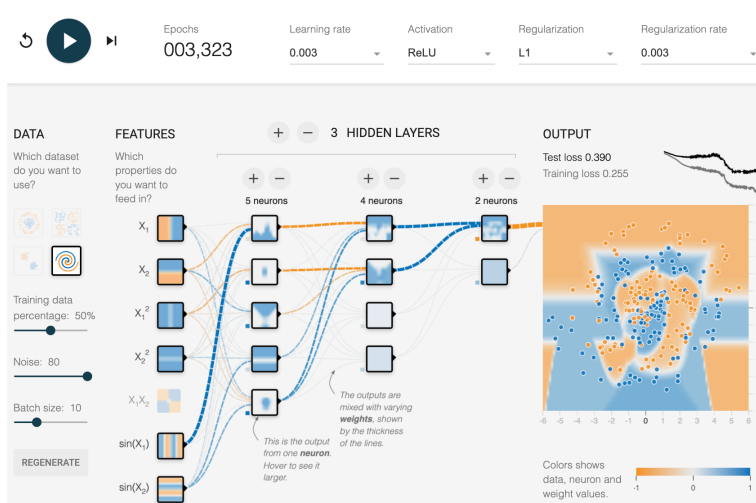
Os resultados melhoraram e tornaram-se mais constantes.

1.3. Espiral da Rede Neural

Resolver o problema utilizando apenas x_1 e x_2 :



Resolver o problema utilizando mais *features*:



Ao aumentar o número de *features*, podemos reduzir o número de neurónios por camada e de camadas, e obter um modelo melhor, tornando a saída do modelo mais suave.

2. Exercícios de Programação

Normalizamos as *features dataset* do conjunto de treino e de teste, ao fazer:
(conjunto - mediana do conjunto)/desvio padrão do conjunto.

Para representar uma camada de *features* definimos 3 *features*: *latitude* x *longitude*, *median_income* e *population*.

regression model

plot function

function to create and train a linear regression model

deep neural net model