20231010-ML

𝜃1 Peso da caracteristica x

𝜃2 Bias

L = loss

Derivadas parciais

Gradiente a partir de um ponto chegar ao local ideal

Tendencia para cair em minimos locais

Minimo local é todos os sitos são todos piores

Um modelo é um vetor

Regressão Logistica

Acrescenta uma função transferencia, algo tipo sim ou não

Acrescentamos na regressão linear uma função de transferencia

O mais interessante é o que compõe um neurônio de uma rede

**É uma regressão linar com um “IF” a frente**

A primeira linha da regressão linear + uma função de transferencia

As variáveis precisam ser Diferenciáveis

Diferenciáveis = sermos capazes de encontrar a derivada, por causa do gradiente

Produto + soma + if

Mantermos diferenciaveis, para que encontre os parametros de forma automatica

Encontrar os parametros ótimos é o que a ML faz com derivadas

A derivada da função azul não é diferenciada, não tem derivada

FT =

Derivada de Classificação

É uma função gaunesiana

Função de transferencia: Uma função aproxima-se assintoticamente de valor 1 ou 0 sem nunca tocar

=====================

Função de Custo: diferenca entre o que dava e o que deveria dar

Função de Custo Logistico:

Preocupa-te mais aos casos que estão mais longes, custos mais altos

Quando tu falhares por pouco não é grave, mas quando falhes muito preocupa-te

Entropia cruzada = cross entropy

Penaliza muito quando falhas muito e penalizas nada quando falhas pouco

Paradigma= one against all

20 ou 30 sistemas para detectar objetos = linhas

Colocar os diferentes neuronios para trabalhar em casos especialistas

O tensorflow = matrix + fluxos

Dá me bibliotecas de diferenciação automatica

Gradientes para otimizar

TensorFlow: otimiza diferenciação automatica

Hoje é algo como grafo desenvolver um algoritmo

Muito uso de keras e pytorch

Fazem modelos com imensas camadas

VGG 19 tem camadas

Os ultimos neuronios de uma rede mutias vezes é regressão linear

Convulsão, mistura de sinais

Estudar em base o pytorch