

Parte 1 - - Teórico/Dissertativa.

1. Cite a principal limitação do Perceptron e como ela ficou popularmente conhecida. Qual a estratégia foi proposta para contornar tal limitação?

XOR Problem e Multilayer network

A principal limitação do Perceptron foi a sua falta de capacidade de resolver problemas não-lineares. Ele era capaz de resolver os problemas lineares clássicos, porém, não era capaz de resolver o problema XOR, que é um clássico problema não linearmente separável. A partir disso, foi preciso descobrir outros métodos que pudessem superar essa falha do perceptron. A estratégia proposta para contornar a limitação foi a criação de novas camadas, chamadas de “camadas ocultas”, que iriam envolver funções de ativações não lineares como a função sigmoid e a função tanh (tangente hiperbólica), de modo que esse novo multilayer perceptron fosse capaz de realizar operações não lineares. A adição de camadas não lineares permitiu que os perceptrons pudessem entender correlações mais complexas entre as variáveis, além de possibilitar o uso de diferentes funções de ativações, de modo que cada função pode extrair uma característica de uma feature.

2. Qual a função de ativação do perceptron?

A função de ativação do perceptron é a signal function, ou step function. é uma função que se baseia na seguinte premissa:

se $z > 0$ então $f(z) = 1$; se $z < 0$, então $f(z) = 0$.

dado que $z = w.T * x + b$

O resultado do perceptron vai ser uma classificação em duas classes, de modo que ele vai separar linearmente essas duas classes, onde acima dessa linha será uma classe e abaixo será outra classe. A partir disso, o perceptron só é capaz de ser utilizado em problemas linearmente separáveis.

3. Qual a função de ativação utilizada na regressão logística? Qual a expressão da derivada correspondente?

A função de ativação utilizada na regressão logística é a função sigmoid que é dada por:

$1/(1+e^{-z})$ onde $z = w.T * x + b$

A função sigmoid começa em 0, tem seu centro em 0.5 e termina em 1, com um formato em S. Ela irá transformar a saída linear, que no perceptron normal era binária, pertencente ou não, para uma probabilidade.

A sua derivada é dada por $\text{sig}(x) = \text{sig}(x) * (1 - \text{sig}(x))$

ela é diferenciável e tem uma transição gradual do 0 para o 1

4. Explique com suas palavras quais são as etapas principais do processo de treinamento de um neurônio. Explique cada etapa de forma sucinta

1 - inicializar os parametros

Inicializa os parametros de alguma forma, você pode utilizar inicialização totalmente aleatoria, inicialização com zeros ou algum tipo de inicialização como a "He". A inicialização pode mudar a forma com que seu modelo se comporta, logo é importante definir uma boa.

2 - forward propag para ter os resultados

com os primeiros valores de parametros definidos, deve-se testar esses parametros para que possamos entender como que o modelo se comporta, de modo a calcularmos o loss function e a consequente cost function do nosso modelo, para que assim possamos saber quao distante do resultado esperado nós estamos. Essa etapa é essencial justamente para termos um pontapé inicial de como nosso modelo está se comportando. Diferentes formas de inicialização podem gerar diferentes resultados.

3 - computar o loss function e a consequente cost function

Calcular

4 - backprop para achar por meio do gradiente descendente os novos valores de pesos e etc

Dado que temos os parametros e o cost function, nosso objetivo é minimizar a cost function, diminuindo a diferença entre o output da rede neural e o valor esperado. Para isso, utiliza-se o backpropagation por meio do gradiente descendente, que irá buscar minimizar, por meio de uma função step, que irá atualizar os valores dos parametros até achar um "minimo" na curva do gradiente descendente, de modo que a função de custo seja minimizada. A cada iteração do backpropagation, ele irá atualizar os valores dos parametros. Após uma quantidade definida de iterações ou ele atingir um treshhold, então ele irá parar e utilizar os parametros definidos para predizer

5 - atualiza os parametros

6 -

5 - Diferença entre inicializar pesos aleatoriamente e zeros

São duas formas de você inicializar os paramêtros de uma rede neural. Você pode inicializar todos os valores como zeros, ou pode utilizar gerações aleatorias de números. Escolher a melhor forma de se inicializar os parametros pode auxiliar a rede neural no aprendizado e evitar problemas na descida do gradiente. Ao inicializar com zero, você corre o risco das camadas ocultas aprenderem exatamente as mesmas características, o que não é desejado. Ao se inicializar com pesos aleatorios, você quebra a simetria, fator importante para que as camadas ocultas de uma rede neural possam extrair diferentes características dos dados.

6 - Diferença entre as activations functions

Cada função de ativação tem uma característica e é melhor aplicada a um certo contexto, de modo que extrai informações dos dados de uma forma. Logo, idealmente, uma rede neural irá combinar várias camadas ocultas com diferentes funções de ativação. Algumas das mais famosas são a função identidade, onde o valor é igual $f(x) = x$

Temos a função signal/step, que é utilizada no perceptron single layer: se $x > 0$, logo 1; cc 0

Temos a função sigmoid, que é utilizada na regressão logística, e representa um número real em termos da probabilidade do evento acontecer, de modo que ela tem a fórmula $1/(1+e^{-z})$ e sua derivada é $\text{sig}(x)(1-\text{sig}(x))$

Temos a função tanh, tangente hiperbólica, que é uma versão “melhorada” da sigmoid, de modo que ela varia de -1 a 1 com centro no 0

7 - O que é loss function e cost function

A função de perda é a responsável por calcular a diferença entre o resultado de predição do modelo e o resultado esperado. Temos algumas formas de calcular esse erro como o mean squared $(\hat{y}-y)^2$ e o cross entropy log loss que é $y \log \hat{y} + (1-y)(\log(1-\hat{y}))$

A função de custo é a média das funções de perda de todas as previsões que a rede neural fez. Ela vai representar o custo médio que a rede neural está tendo, de modo que o objetivo de otimizar uma rede neural será de minimizar essa função de custo, de modo que a diferença entre o resultado que nossa rede neural predisse e o resultado esperado seja o menor possível. Ela é dada por $J = 1/m \sum (\text{loss function})$

8 - O que é regularization, high bias e variance

9 - Explique L2 Reg, L1 Reg e Dropout Reg