MULTILAYER PERCEPTRON

LISTA DE EXERCÍCIOS 2

Perceptron

I) Por que uma rede de Perceptrons de Rosenblatt não pode ser treinada via descida de gradiente e *backpropagation*?

II) Prove que um Perceptron com duas entradas $\hat{y} = sinal(\theta_0 + X_1\theta_1 + X_2\theta_2)$ gera uma fronteira de decisão linear.

III) Crie um Perceptron que se comporta como uma função *and*.

IV) Demonstre que uma regularização L2 na função de custo de uma unidade com função de ativação identidade é equivalente a diminuir a magnitude do vetor de pesos antes de realizar a atualização. *Dica: escreva a função de custo com o termo de regularização* $||\theta||^2$ *e use essa função de custo na expressão de atualização dos pesos da descida de gradiente. Colocando o* θ *em evidência ficará claro que os pesos estão sendo multiplicados por um valor na faixa* $(0, 1)$.

Descida de Gradiente e Backpropagation

V) Dado que a descida de gradiente possui na sua formalização original a seguinte regra de atualização $\theta_{t+1} = \theta_t = \eta\nabla_\theta J$. Por que a atualização dos pesos com $-\nabla J_\theta$ diminui o valor da função de custo?

VI) Por que dizemos que a formulação original do *momentum* calcula o gradiente "no local errado"?

VII) Qual o efeito prático de normalizar o gradiente (conforme feito no Adam)?

VIII) Considere um MLP com uma unidade na entrada, uma unidade na saída, K camadas ocultas e D unidades em cada camada. Quantos parâmetros esse MLP possui?

IX) Quais são os quatro passos realizados para fazer o treinamento de um MLP?

X) Qual a diferença de uma iteração para uma época?

Funções de Ativação

XI) Calcule as derivadas das seguintes funções de ativação:

a) ReLU $\varphi(x) = max(0, x)$

b) Sigmoid $\varphi(x) = \frac{1}{(1+e^{-x})}$

c) ☠ Softmax $\varphi(\mathbf{x}) = S_i = \frac{e^{x_i}}{\Sigma e^{x_j}}$, (encontrar a matriz Jacobiana)

XII) Qual a utilidade das funções de ativação nas camadas ocultas de um MLP?

XIII) Qual a utilidade das funções de ativação na camada de saída de um MLP?

Inicialização de Pesos

XIV) Que problemas podemos ter com unidades na camada oculta que usam a função sigmoid?

XV) Que problemas podemos ter com unidades na camada oculta que usam a função relu?

XVI) Qual a motivação das inicializações *He* e *Xavier*.

Funções de Custo

XVII) Calcule as derivadas das seguintes funções de custo:

a) $L_{l2} = \sum \left(y^{(i)} - \hat{y}^{(i)}\right)^2$

b) $L_{BCE} = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$

XVIII) Desenvolva a função de custo do Erro Médio Quadrático a partir da distribuição Gaussiana.

$$\Pr(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

XIX) Desenvolva a função de custo Entropia Binária Cruzada a partir da distribuição de Bernoulli.

$$\Pr(y|\lambda) = (1-\lambda)^{(1-y)}\lambda^y$$

XX) ☠ Suponha que você quer criar uma rede neural para estimar a direção y (em radianos) do vento a partir de medições de pressão $\mathbf{x}$. Uma distribuição que trabalha no domínio circular é a distribuição de von Mises:

$$\Pr(y|\mu,k) = \frac{\exp[k * \cos(y-\mu)]}{2\pi * \mathrm{Bessel}_0[k]}$$

onde μ é a direção média e k é o inverso da variância. O termo $\mathrm{Bessel}_0[k]$ é uma função modificada de Bessel com grau 0. Crie uma função de custo para aprender o parâmetro μ dessa distribuição.

XXI) ☠ Suponha que você quer criar uma rede neural para estimar o número de pedestres $y \in \{0,1,2,...\}$ que passam em uma rua da cidade em algum determinado momento. Você possui um vetor de atributos $\mathbf{x}$ que contém informações sobre a hora do dia, sobre o clima e sobre o dia ser feriado ou não. Uma distribuição de probabilidade adequada para trabalhar com esse problema é a distribuição de Poisson:

$$\Pr(y=k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

onde λ é o único parâmetro, que representa a média da distribuição. Crie uma função de custo assumindo que temos acesso a I instâncias de treinamento compostas por pares $\{x^i, y^i\}$.

Gabarito

I) Pois a derivada da função de ativação sinal é zero ou indefinida em todo domínio da função. Devido a isso, não conseguimos propagar os erros.

II) Para resolver essa questão você deve mostrar que $\theta_0 + X_1\theta_1 + X_2\theta_2 = 0$ é uma reta.

III) $\hat{y} = \mathrm{sinal}(\theta_0 + X_1\theta_1 + X_2\theta_2)$, onde $\theta_0 = -1$, $\theta_1 = 1$, $\theta_2 = 1$ e as entradas falsas são codificadas como -1 e as entradas verdadeiras como $+1$.

IV) $\theta_{(t+1)} = (1 - \eta\beta)\theta_t - \nabla_\theta J$

V) O gradiente nos dá a direção de atualização dos pesos que causaria aumento na função de custo. Ao atualizar os pesos em direção contrária, estamos diminuindo o custo.

VI) Dizemos que o gradiente está atrasado pois, devido ao *momentum*, já existe uma atualização dos pesos conforme o histórico dos gradientes. Essa observação leva a definição do *Nesterov Accelerated Momentum*.

VII) Ao normalizar o gradiente, fica mais fácil definir uma taxa de aprendizado que funcione igualmente bem para todos os parâmetros da rede.

VIII) $3D + 1 + (K-1)D(D+1)$. Repare que essa expressão não é a única resposta possível para o exercício. Dependendo de como você desenvolver a questão poderá chegar em expressões equivalentes como por exemplo: $2D + (K-1)DD + KD + 1$

IX) *Forward Pass*: onde as saídas são calculadas; *Loss Function*: onde a diferença entre a saída e o valor esperado são comparadas; *Backward Pass*: onde os gradientes da função de custo em relação a cada um dos pesos é calculado; *Optimizer*: onde os pesos são atualizados.

X) Uma iteração consiste na apresentação de algumas instâncias para o modelo, sob as quais são realizados *Forward Pass*, cálculo da função de custo, *Backward Pass* e otimização. Uma época ocorre quando foram feitas iterações suficientes para ver todas as instâncias do conjunto de treinamento 1 vez.

XI) a)

$$\varphi'(x) = \begin{cases} 0, & x \leqslant 0, \\ 1, & x > 0 \end{cases}$$

b) $\varphi'(x) = \varphi(x)(1 - \varphi(x))$

c) ☠

$$\varphi'(x) = \begin{cases} S_i(1 - S_i), & i = j, \\ -S_iS_j, & i \neq j \end{cases}$$

XII) Adicionar não linearidades.

XIII) Ajustar a saída da rede neural a imagem da função modelada

XIV) *Vanishing Gradient*

XV) Unidades mortas

XVI) Inicializar os pesos da rede neural de modo que não ocorra explosão nem dissipação de gradiente.

Prof. Lucas Silveira Kupssinskü
PUCRS - Escola Politécnica

XVII) a) $2(\hat{y} - y)$

b) $\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}$

XVIII) $L_{l2} = \sum \left(y^{(i)} - \hat{y}^{(i)}\right)^2$

XIX) $L_{BCE} = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$

XX) ☠ $-\cos(y^{(i)} - f(x^{(1)}, \theta))$

XXI) ☠ $\sum_{i=1}^{I}(f[x^{(i)}, \theta] - y^{(i)}\ln(f[x^{(i)}, \theta]))$