

<b>Iniciado em</b>	
<b>Estado</b>	Finalizada
<b>Concluída em</b>	
<b>Tempo empregado</b>	3 horas 48 minutos
<b>Notas</b>	7,00/8,00
<b>Avaliar</b>	8,75 de um máximo de 10,00(88%)

### Questão 1

Completo

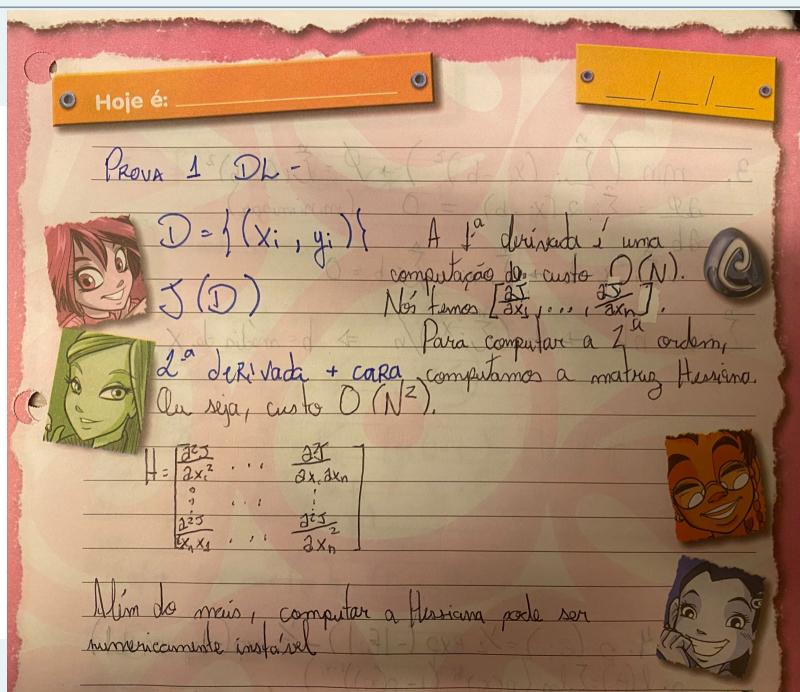
Atingiu 1,00 de 1,00

Assumindo um conjunto de dados,  $\mathcal{D} = \{(x_i, y_i)\}$ , e uma função de perda  $J(\mathcal{D})$ . Por que a segunda derivada é muito mais cara de calcular do que a primeira derivada?

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos: `\ ( x =2 \ )` ou `\ ( \frac{dx}{dy} \ )`

Remova o espaço entre `\` e `(` nos exemplos acima.



[1.jpeg](#)

Comentário:

**Questão 2**

Completo

Atingiu 1,00 de 1,00

Qual é a derivada da função:  $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$ ? Aqui,  $\mathbf{x}$  é um vetor de dimensão  $m$ .

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos: `\( x =2 \)` ou `\( (\frac{dx}{dy}) \)`

Remova o espaço entre `\` e `(` nos exemplos acima.

The image shows handwritten mathematical work on lined paper. At the top, there is a red stamp or drawing of a triangle with a circle inside. The handwriting is in black ink. It starts with the definition of the function  $f(x) = \|\mathbf{x}\|_2^2$ , followed by the application of the chain rule to find its derivative. The derivative is shown as  $\frac{d}{dx} \left[ \left( \sum_i |x_i|^2 \right)^{\frac{1}{2}} \right]^2 = \sum_i |x_i|^2 = \sum_i 2|x_i|$ . Below this, the final result is given as  $f'(x) = \sum_{i=1}^n 2|x_i|$ .

[2.jpeg](#)

Comentário:

**Questão 3**

Completo

Atingiu 1,00 de 1,00

Suponha que temos alguns dados  $x_1, \dots, x_n \in R$ . Nossa objetivo é encontrar uma constante  $b$  tal que  $\sum_i (x_i - b)^2$  é minimizado. Encontre uma solução analítica para o valor ideal de  $b$ . Como esse problema e sua solução se relacionam com a distribuição normal?

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos: `\( x = 2 \)` ou `\( (\frac{dx}{dy}) \)`

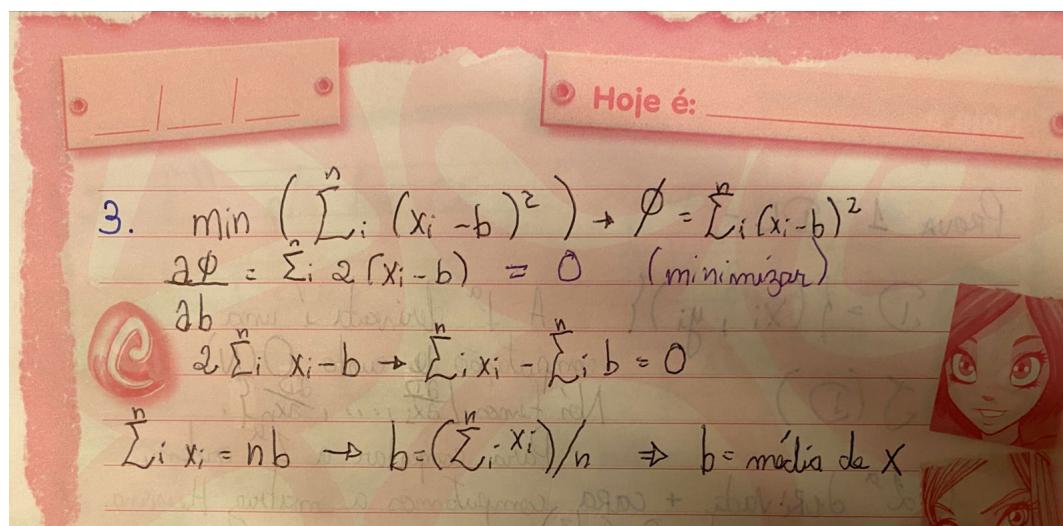
Remova o espaço entre `\` e `(` nos exemplos acima.

Temos que o valor ideal de  $b = \text{média}(x)$ . Isso quer dizer que a soma será o menor possível o quanto mais próximo  $x_i$  está de  $b$ . Contudo, se tivermos dados muito discrepantes, com muito outliers podemos ter uma média muito distante da maior parte dos números, ou seja, a soma será grande. Para evitar isso, ou seja, para evitar que a diferença entre  $x_i$  e  $b$  seja muito grande (e influencie negativamente o tamanho do step que tomaremos) nós normalizamos os dados. Ou seja, transformamos eles em uma distribuição normal usando sua média (o valor ideal de  $b$ )

OBS: primeira parte está na foto

 [3.jpeg](#)

Comentário:



3.  $\min \left( \sum_{i=1}^n (x_i - b)^2 \right) \rightarrow \phi = \sum_{i=1}^n (x_i - b)^2$   
 $\frac{\partial \phi}{\partial b} = \sum_{i=1}^n 2(x_i - b) = 0 \quad (\text{minimizar})$   
 $2b - 2 \sum_{i=1}^n x_i = 0 \rightarrow \sum_{i=1}^n x_i - \sum_{i=1}^n b = 0$   
 $\sum_{i=1}^n x_i = nb \rightarrow b = \frac{\sum_{i=1}^n x_i}{n} \Rightarrow b = \text{média de } X$

**Questão 4**

Completo

Atingiu 0,00 de 1,00

Suponha um modelo de regressão linear onde:

$y_i = w \cdot x_i + b + \epsilon_i$ . Aqui,  $\epsilon_i$  é o erro do  $i$ -ésimo exemplo. Ou seja:

$$\epsilon_i = y_i - w \cdot x_i - b$$

Agora, assuma um modelo que governa o erro  $\epsilon$  é a distribuição exponencial. Isso é,  $p(\epsilon_i) = \frac{1}{2} \exp(-|\epsilon_i|)$ . Sabendo disto:

1. Escreva a log-verossimilhança dos dados no modelo.
2. É possível encontrar uma solução fechada para os parâmetros livres? Para responder tal pergunta, escreva a primeira derivada dos parâmetros.
3. Por fim, indique se um algoritmo de gradiente ascendente teria problemas com tal modelo.

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos: `\ ( x =2 \ )` ou `\ ( \frac{dx}{dy} \ )`

Remova o espaço entre `\` e `(` nos exemplos acima.

Não é possível encontrar uma solução fechada para os parâmetros livres. A derivada em relação a:

$$W = \text{Sum}[ -(2x(1-y) e^{\text{abs}(-b - w x + y)} \text{Abs}'(-b - w x + y)) / (2 e^{\text{abs}(-b - w x + y)} - 1) - \text{abs}(-b - w x + y)]$$

$$b = \text{Sum}[ -(2x(1-y) e^{\text{abs}(-b - w x + y)} \text{Abs}'(-b - w x + y)) / (2 e^{\text{abs}(-b - w x + y)} - 1) - \text{abs}(-b - w x + y)]$$

3. O algoritmo teria problemas, pois não conseguiria uma solução fechada para os parâmetros livres e, assim, não conseguia atualizá-los

[4.jpeg](#)

Comentário:

$$\begin{aligned}
 4. p(\epsilon_i) &= \frac{1}{2} \exp(-|\epsilon_i|) \rightarrow \frac{1}{2} \exp(-|y_i - wx_i - b|) \\
 a. l(\theta) &= \sum_i \log(p(x_i))^{y_i} (1-p(x_i))^{1-y_i} \quad |\epsilon_i| = K \\
 &= \sum_i y_i \log(p(x_i)) + \sum_i (1-y_i) \log(1-p(x_i)) \\
 &= \sum_i y_i \log\left(\frac{1}{2} e^{-K}\right) + \sum_i (1-y_i) \log\left(\frac{(2e^K-1)}{2e^K}\right) \\
 &= -\sum_i y_i \log(2e^K) + \sum_i (1-y_i) \log(2e^K-1) - \sum_i (1-y_i) \log(2e^K) \\
 &= \sum_i (1-y_i) \log(2e^K-1) - \sum_i K \epsilon_i \log \\
 &= \sum_i (1-y_i) \log(2 e^{y_i + wx_i + b}) - \sum_i \log 2 + |y_i - wx_i - b|
 \end{aligned}$$

## Questão 5

Completo

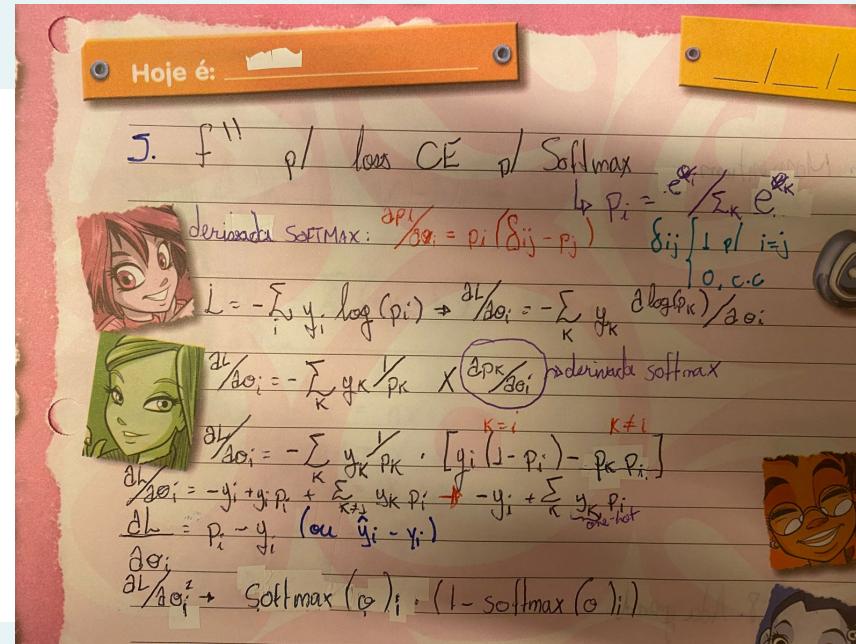
Atingiu 1,00 de 1,00

Compute a segunda derivada da função de perda de entropia cruzada (negativo da verossimilhança) para um classificador Softmax. Assuma que os dados são  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ , onde  $|\mathcal{D}| = N$ . Além do mais,  $P_k(y_i | \mathbf{x}_i)$  é a probabilidade para uma classe  $k \in [1, K]$ . A forma de  $P_k(y_i | \mathbf{x}_i)$  pode ser encontrada nos slides.

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos:  $\backslash ( x =2 \backslash )$  ou  $\backslash ( \frac{dx}{dy} \backslash )$

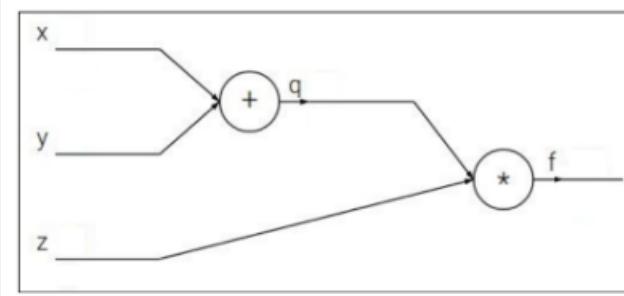
Remova o espaço entre  $\backslash$  e  $($  nos exemplos acima.



5.jpeg

Comentário:

Seja a função  $f(x, y, z) = (x + y)z$ , dada pelo diagrama da figura abaixo.



a) Calcule as seguintes derivadas parciais

$$\frac{\partial f}{\partial x}$$

$$\frac{\partial f}{\partial y}$$

$$\frac{\partial f}{\partial z}$$

$$\frac{\partial f}{\partial q}$$

- 4 -

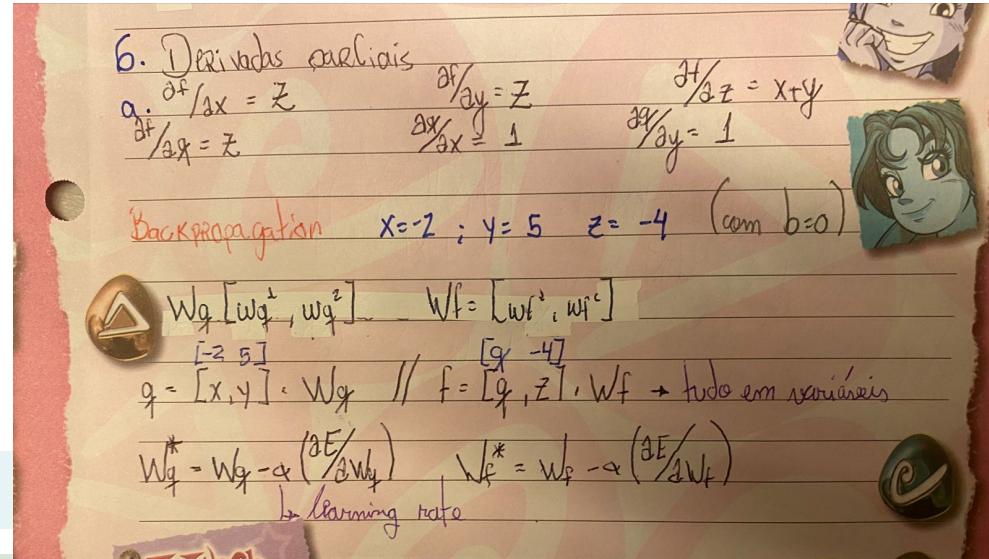
$$\frac{\partial q}{\partial x}$$

$$\frac{\partial q}{\partial u}$$

89

b) Mostre passo a passo o cálculo do Backpropagation, considerando as entradas  $x = -2, y = 5, z = -4$

Como eu não tinha valores de learning rate, inicialização dos pesos ou função de erro eu fiz tudo usando variáveis. Também assumi um bias zero, apenas porque fazia sentido dada a função  $f(x)$ .



### Comentário:

**Questão 7**

Completo

Atingiu 1,00 de 1,00

**Momentum**

Em sala de aula, vimos o gradiente descendente, gradiente descendente estocástico e o gradiente descendente em minibatch. Agora, assuma que temos uma implementação dos três algoritmos em sua forma base. Além disso, assuma que cada uma dessas versões foi subsequentemente alterada para fazer uso de Momentum. Se é que existem, quais você acha que seriam os problemas do Momentum na implementação estocástica e minibatch? Compare com a versão "normal", ou seja, Momentum com gradiente descendente clássico.

Dica: Assuma que os atributos não foram normalizados. Compare as primeiras iterações com iterações mais a frente.

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos:  $\backslash ( x =2 \backslash )$  ou  $\backslash ( \frac{dx}{dy} \backslash )$

Remova o espaço entre  $\backslash$  e  $($  nos exemplos acima.

Estocástica: Momentum evita a paralisação do processo de otimização que é muito mais provável de ocorrer aqui.

Mini-batch: Em termos de melhoria, nos permite aumentar nossa "velocidade" em uma determinada direção. Quando as direções mudam, ficamos um pouco mais lentos porque nosso momentum "quebrou".

Um problema em ambos os casos é que, quando a learning rate é baixa, o termo do momentum e o gradiente atual por si só podem causar oscilações. Momentum leva em consideração os gradientes anteriores para "smooth out". Isso quer dizer que, no início nós temos poucos gradientes e nossa suavização pode ser ruim. Isso deve melhorar com mais iterações do algoritmo, mas o SGD tem menos steps, logo ele terá menos gradientes a seu dispõr. Com dados não normalizados a variação tende a ser maior, podemos acabar por suavizar demais.

Comentário:

**Questão 8**

Completo

Atingiu 1,00 de 1,00

**AdaGrad**

Em sala de aula, vimos o gradiente descendente, gradiente descendente estocástico e o gradiente descendente em minibatch. Agora, assuma que temos uma implementação dos três algoritmos em sua forma base. Além disso, assuma que cada uma dessas versões foi subsequentemente alterada para fazer uso de AdaGrad. Se é que existem, quais você acha que seriam os problemas do AdaGrad na implementação estocástica e minibatch? Compare com a versão "normal", ou seja, AdaGrad com gradiente descendente clássico.

Dica: Assuma que os atributos não foram normalizados. Compare as primeiras iterações com iterações mais a frente.

Faça upload da solução ou escreva em Latex usando notação de parênteses.

Exemplos:  $\backslash ( x =2 \backslash )$  ou  $\backslash ( \frac{dx}{dy} \backslash )$

Remova o espaço entre  $\backslash$  e  $($  nos exemplos acima.

Mini-batch: O algoritmo AdaGrad incorpora dinamicamente o conhecimento da geometria dos dados observados em iterações anteriores para realizar um aprendizado mais informativo com base em gradiente. O que é bom para o mini-batch.

O problema principal do AdaGrad é o acúmulo de gradientes quadrados no denominador: como cada termo adicionado é positivo, a soma acumulada continua crescendo durante o treinamento. Isso, por sua vez, faz com que a taxa de aprendizado diminua e eventualmente se torne infinitesimalmente pequena, ponto em que o algoritmo não é mais capaz de adquirir conhecimento adicional. Isso quer dizer que ele tem resultado melhor no SGD que no GD (pois o primeiro tem menor número de steps).

O AdaGrad vai funcionar bem nas primeiras iterações, contudo, ele tenderá a zero. Ou seja, o aprendizado irá parar.

Comentário:

[◀ 2021\\_2 - Plano de Ensino](#)

Seguir para...



[Prova 2 ►](#)