Lista 2 - 22/10/2023

João Pedro Cunha - 1910626

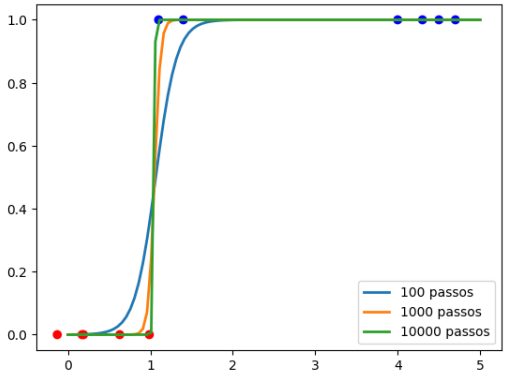
▼ 6.1. Implementing sigmoidal Least Squares cost

Repeat the experiment described in Example 6.3 by coding up the Least Squares cost in Equation (6.9) and the normalized gradient descent algorithm detailed in Section A.3. You need not reproduce the contour plot shown in the right panel of Figure 6.5; however, you can verify that your implementation is working properly by reproducing the final fit shown in the left panel of that figure. Alternatively show that your final result produces zero misclassifications (see Section 6.8.3).

- A partir dos códigos visto em aula, foi realizada a classificação do dataset de 2 dimensões (2d_classification_data_v1_entropy.csv'), com a aplicação do gradiente descendente normalizado.
- Função custo utilizada + gradiente correspondente (visto em aula)

$$g(w) = \frac{1}{P}\sum_{p=1}^{P}(\sigma(\mathring{x}_p^T w))^2$$
$$\nabla g(w) = \frac{2}{P}\sum_{p=1}^{P}\frac{e^{-\mathring{x}_p^T w}}{1+e^{\mathring{x}_p^T w}}\mathring{x}_p$$

- A implementação em Python foi anexada junto com esse documento. A imagem abaixo mostra o resultado do gradiente normalizado com 10000 passos e $\alpha_0 = 10$



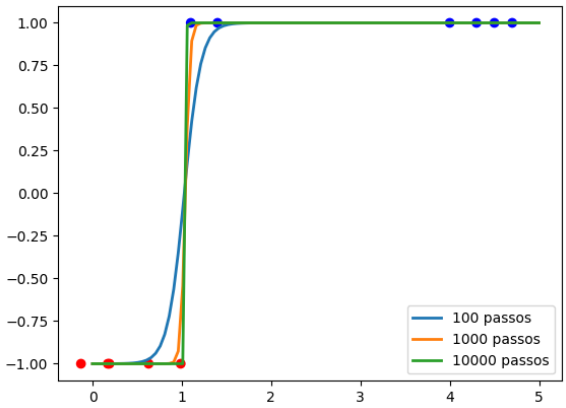
▼ 6.7. Implementing the Softmax cost

Repeat the experiment described in Example 6.5 by coding up the Softmax cost function shown in Equation (6.25). You need not reproduce the contour plot shown in the right panel of Figure 6.9; however, you can verify that your implementation is working properly by reproducing the final fit shown in the left panel of that figure. Alternatively show that your final result produces zero misclassifications (see Section 6.8.3).

- Função custo e gradiente correspondente:

$$g(w) = \frac{1}{P}\sum_{p=1}^{P}\log(1+e^{-y_p\mathring{x}_p^T w})$$
$$\nabla g(w) = -\frac{1}{P}\sum_{p=1}^{P}\frac{e^{-y_p\mathring{x}_p^T w}}{1+e^{-y_p\mathring{x}_p^T w}}y_p\cdot\mathring{x}_p$$

- A implementação em Python foi anexada junto com esse documento. A imagem abaixo mostra o resultado do gradiente normalizado com passo inicial $\alpha_0 = 10$, e 3 valores de passo. Pode-se observar que, à medida que aumentamos o passo, obtemos uma separação sem missclassifications.



▼ 6.10. The Perceptron cost is convex

Show that the Perceptron cost given in Equation (6.33) is convex using the zero-order definition of convexity described in Exercise 5.8.

- Uma função g é convexa se, e somente se,

$$g(\lambda w_1 + (1-\lambda)w_2) \leq \lambda g(w_1) + (1-\lambda)g(w_2), \forall\ w_1, w_2 \in U, \forall\ \lambda \in [0,1]$$

- A função custo do Perceptron é $g(w) = \frac{1}{P}\sum_{p=1}^{P} \max(0, -y_p \cdot \mathring{x}_p^T w)$
- Assim, o lado esquerdo fica

$$g(\lambda w_1 + (1-\lambda)w_2) = \frac{1}{P}\sum_{p=1}^{P} \max(0, -y_p \cdot [\lambda w_1 + (1-\lambda)w_2] \cdot \mathring{x}_p^T)$$
$$= \sum_{p=1}^{P} \max(0, [-\lambda y_p w_1 \mathring{x}_p^T - (1-\lambda) y_p w_2 \mathring{x}_p^T])$$

- E o lado direito:

$$\lambda g(w_1) + (1-\lambda)g(w_2) =$$
$$\frac{1}{P}\sum_{p=1}^{P} \max(0, -\lambda y_p w_1 \mathring{x}_p^T) + \frac{1}{P}\sum_{p=1}^{P} \max(0, -(1-\lambda) y_p w_2 \mathring{x}_p^T) = \sum_{p=1}^{P} [\max(0, -\lambda y_p w_1 \mathring{x}_p^T) + \max(0, -(1-\lambda) y_p w_2 \mathring{x}_t^T$$

- Assim, denotando $a_p = -\lambda y_p w_1 \mathring{x}_p^T$ e $b_p = -(1-\lambda) y_p w_2 \mathring{x}_p^T$, temos que provar que

$$\sum_{p=1}^{P} \max(0, a_p + b_p) \leq \sum_{p=1}^{P} [\max(0, a_p) + \max(0, b_p)]$$

- Tal equação é verdadeira se $\max(0, a_p + b_p) \leq \max(0, a_p) + \max(0, b_p), \forall\ a_p, b_p$. Para provar tal afirmação sem perda de generalidade, podemos assumir que $|a| \geq |b|$
  - Caso 1: $a \geq 0, b \geq 0$

$$\max(0, a+b) \leq \max(0, a) + \max(0, b)$$
$$\therefore (a+b) \leq a + b$$

  - Caso 2: $a \geq 0, b < 0$

$$|a| \geq |b| \rightarrow 0 \leq a + b \leq a$$
$$\max(0, a+b) \leq \max(0, a) + \max(0, b)$$
$$\therefore a + b \leq a$$

- Caso 3: $a < 0, b \geq 0$

$$|a| \geq |b| \rightarrow a + b \leq 0 \leq b$$
$$\max(0, a + b) \leq \max(0, a) + \max(0, b)$$
$$\therefore a + b \leq b$$

- Caso 4: $a < 0, b < 0$

$$\max(0, a + b) \leq \max(0, a) + \max(0, b)$$
$$\therefore 0 + 0 \leq 0$$

Portanto, provamos que a função é convexa.

▼ 6.11. The Softmax cost is convex

Show that the Softmax cost function given in Equation (6.25) is convex by verifying that it satisfies the second-order definition of convexity. Hint: the Hessian, already given in Equation (6.3.2), is a weighted outer-product matrix like the one described in Exercise 4.2.

- A função custo Softmax é $g(w) = \frac{1}{P} \sum_{p=1}^{P} \log(1 + e^{-y_p \cdot \mathring{x}_p^T w})$
- Podemos calcular a matriz Hessiana como

$$M = \nabla^2 g(w) = \frac{1}{P} \sum_{p=1}^{P} \left(\frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \right) \left(1 - \frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \right) \mathring{x}_p \mathring{x}_p^T$$

- A função $g(w)$ é convexa se, e somente se, a matriz Hessiana for positiva semi-definida, ou seja: $v^T M v \geq 0, \forall\ v \in \mathbb{R}^n$
- Podemos chamar o termo escalar da soma de k_p, e verificamos que k é sempre positivo:

$$k_p = \left(\frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \right) \left(1 - \frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \right)$$
$$e^{\theta} \geq 0, \forall\ \theta \in \mathbb{R}$$
$$\therefore 0 \leq \frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \leq 1$$
$$\therefore 1 - \frac{1}{1 + e^{-y_p \mathring{x}_p^T w}} \geq 0$$
$$\therefore k_p \geq 0$$

- Daí, e sabendo que a norma $||v||^2$ de um vetor $v \in \mathbb{R}^n$ é sempre não-negativa, finalizamos a demonstração:

$$v^T M v = v^T \left(\frac{1}{P} \sum_{p=1}^{P} k_p \mathring{x}_p \mathring{x}_p^T \right) v$$
$$= \frac{1}{P} \sum_{p=1}^{P} k_p (v^T \mathring{x}_p)(\mathring{x}_p^T v)$$
$$= \frac{1}{P} \sum_{p=1}^{P} k_p \cdot \|v^T \mathring{x}_p\|^2 \geq 0$$

▼ 6.13. Compare the efficacy of two-class cost functions I

Compare the efficacy of the Softmax and the Perceptron cost functions in terms of the minimal number of misclassifications each can achieve by proper minimization via gradient descent on a breast cancer dataset. This dataset consists of P=699 data points, each point consisting of N = 9 input of attributes of a single individual and output label indicating whether or not the individual does or does not have breast cancer. You should be able to achieve around 20 misclassifications with each method.

- A implementação dos modelos foi bem-sucedida, convergindo para resultados satisfatórios. Em ambos, foi utilizado o gradiente descendente normalizado, $\alpha = 10$ e 10000 passos
- O gradiente descendente com Softmax demorou mais para convergir e obteve piores resultados

```
c = 0.10061943697259336
c = 0.10061943697243891
===== Fim do gradiente =====
w=[ 9.94528658 -0.57754259  0.01153833 -0.56792358 -0.31367156 -0.13055179
 -0.57992909 -0.12319455 -0.60780458], c=0.10061943697243891, 10000 iteracoes
```

- Já o gradiente com Perceptron convergiu muito mais rapidamente, atingindo um patamar de 0.008 bem cedo nos 10000 passos

```
===== Fim do gradiente =====
w=[ 1.39474244 -0.10328418 -0.0139999  -0.04353292 -0.04295274 -0.02627332
 -0.09228729 -0.00902015 -0.005469  ], c=0.008208041420732266, 10000 iteracoes
```