

# TP555 - AI/ML

## Lista de Exercícios #2

### Regressão Linear

1. Qual técnica de regressão linear você usaria se tivesse um conjunto de treinamento com milhares de features? Explique por quais razões você utilizaria esta técnica.
2. Suponha que as features (i.e., atributos) do seu conjunto de treinamento tenham escalas muito diferentes. Qual técnica de regressão linear pode sofrer com isso e como? O que pode ser feito para mitigar este problema?
3. Suponha que você use o gradiente descendente em batelada e plote o erro de cada época. Se você perceber que o erro aumenta constantemente, o que provavelmente está acontecendo? Como você pode consertar isso?
4. Entre os algoritmos baseados no gradiente descendente (GD) que discutimos (batch, estocástico e mini-batch), qual deles chega mais rapidamente à vizinhança da solução ótima? Qual deles realmente converge? O que você pode fazer para que os outros também convirjam?
5. Em sala de aula, nós discutimos 3 tipos de algoritmos baseados no gradiente descendente (batch, estocástico e mini-batch), porém, o código do mini-batch foi o único que não foi apresentado. Portanto, neste exercício eu peço que vocês
  - a. Implementem o algoritmo do mini-batch
  - b. Testem sua implementação com  $y = 2 \cdot x_1 + 2 \cdot x_2 + w$ , onde  $x_1$ ,  $x_2$  e  $w$  são  $M = 1000$  valores retirados de uma distribuição aleatória Gaussiana normal padrão (i.e, com média 0 e variância igual a 1) e utilizando a função hipótese  $h = a_1 \cdot x_1 + a_2 \cdot x_2$ ,
  - c. Plotem a superfície de erro, a superfície de contorno com os parâmetros  $a_1$  e  $a_2$  para cada iteração do mini-batch, e o gráfico de iteração versus erro,
  - d. Encontrem o valor ótimo do passo de aprendizagem (**Dica:** utilizem os gráficos da superfície de contorno com os parâmetros  $a_1$  e  $a_2$  para cada iteração do mini-batch e o gráfico de iteração versus erro para saber se aquele passo é o ótimo),
  - e. Comparem os resultados do mini-batch com os resultados obtidos com o GD em batelada (batch) e GD estocástico (**Dica:** para a comparação, usem os códigos que estão nos slides da aula e plotem os gráficos da superfície de contorno com os parâmetros  $a_1$  e  $a_2$  para cada iteração e o gráfico de iteração versus o erro para GD em batelada e estocástico).
  - f. Baseando-se nos gráficos do item anterior, a que conclusões vocês podem chegar quanto ao treinamento dos 3 tipos de gradiente descendente?
6. Dada a seguinte função hipótese e assumindo o erro quadrático médio como função de erro

$$h = a_0 + a_1 \cdot x + a_2 \cdot x^2.$$

Encontre as equações de atualização dos pesos/parâmetros para esta função. Em seguida, utilizando os vetores  $x$  e  $y$  definidos abaixo, encontre os parâmetros  $a_0$ ,  $a_1$  e  $a_2$  através do método da regressão de forma fechada e com gradiente descendente em batelada.

$$y = 3 + 1.5x + 2.3x^2 + w,$$

onde  $x$  é um vetor coluna com  $M = 1000$  valores retirados de uma distribuição aleatória uniformemente distribuída no intervalo de  $-5$  a  $5$  e  $w$  é outro vetor coluna com  $M$  valores retirados de uma distribuição aleatória Gaussiana com média  $0$  e variância igual a  $10$ .

- Plote o gráfico do número de iterações versus o erro.
  - Baseado no gráfico acima, encontre o melhor valor para o passo de aprendizagem.
7. Neste exercício você vai utilizar o arquivo **training.csv** onde a primeira coluna são os valores de  $x$  (feature) e a segunda de  $y$  (label). Baixe o arquivo do endereço: [training.csv](#). Após, leia o conteúdo do arquivo, ou seja, os vetores  $x$  e  $y$ , com os seguintes comandos:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('training.csv', header=None)

x = df[0].to_numpy()
y = df[1].to_numpy()

fig = plt.figure(figsize=(10,10))
plt.plot(x, y, 'b.')
```

Em seguida, utilize o algoritmo do **gradiente descendente em batelada** para encontrar os parâmetros de cada uma das seguintes funções hipóteses.

- $h = a_0 + a_1x$
- $h = a_0 + a_1x + a_2x^2$
- $h = a_0 + a_1x + a_2x^2 + a_3x^3$
- $h = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$

Para cada uma das funções hipótese acima faça o seguinte:

- Encontre os valores ótimos dos parâmetros através do método de forma fechada, i.e., equação normal, ou também conhecida como método dos mínimos quadrados.
- Encontre as equações de atualização dos parâmetros assumindo o erro quadrático médio como função de erro.
- Encontre o valor ótimo do passo de aprendizagem.
- Plote um gráfico que mostre  $x$  vs.  $y$  e  $x$  vs.  $h$ , ou seja, um gráfico comparando os dados originais com a estimativa (i.e., hipótese) da função que gerou  $y$ .
- Plote um gráfico com do número de iterações versus o erro.

Em seguida responda às seguintes perguntas

- A. Qual das funções hipótese acima aproxima melhor a função alvo (target), ou seja, qual produz o menor erro ao final do treinamento?
- B. Dado que você encontrou os parâmetros que otimizam cada uma das funções hipótese acima (ou seja, você agora tem um modelo treinado que pode prever o resultado para novos exemplos), use os dados contidos no arquivo [predicting.csv](#) e calcule o erro quadrático médio para cada um dos modelos (i.e., função hipótese). Qual função hipótese resulta no menor erro quadrático médio? O que você consegue concluir a respeito deste resultado?
8. Neste exercício você irá aplicar escalonamento de feature aos dados de treinamento. Dada a seguinte função objetivo

$$y = x_1 + x_2,$$

onde  $x_1$  é um vetor coluna com  $M = 1000$  amostras retiradas de uma distribuição Gaussiana com média 0 e desvio padrão unitário e  $x_2$  é um vetor coluna com  $M$  amostras retiradas de uma distribuição Gaussiana com média 10 e desvio padrão igual a 10. Utilize o gradiente descendente em batelada com a seguinte função hipótese

$$h = a_1 x_1 + a_2 x_2,$$

com  $a_1$  e  $a_2$  iniciais iguais a -20 e -20, respectivamente. Para todos os casos abaixo, treine os modelos com o mesmo número máximo de iterações, por exemplo, 2000 iterações. Pede-se

- Sem aplicar nenhum escalonamento de features aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros  $a_1$  e  $a_2$  encontrados durante as iterações (ou seja, o histórico de valores que o algoritmo encontra durante o treinamento do modelo) e o gráfico de erro versus iteração. Não se esqueça de encontrar o valor ótimo para o passo de aprendizagem.
- Aplique a normalização min-máx aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros  $a_1$  e  $a_2$  encontrados durante as iterações e o gráfico de erro versus iteração. Não se esqueça de encontrar o valor ótimo para o passo de aprendizagem.
- Aplique a padronização aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros  $a_1$  e  $a_2$  encontrados durante as iterações e o gráfico de erro versus iteração. Não se esqueça de encontrar o valor ótimo para o passo de aprendizagem.
- Baseado nos resultados anteriores o que você pode concluir à respeito do escalonamento de features? (**Dica:** Comente a respeito da forma da superfície de erro, do número de iterações necessárias para se alcançar o ponto ótimo, etc. Quanto mais detalhada sua análise dos resultados, melhor será sua avaliação neste exercício.)