

AI VIET NAM – COURSE 2022

Calculus - Exercise 4

Numpy và Vectorization

Ngày 15 tháng 8 năm 2022

1. **NumPy Exercises:** Các bạn hoàn thành 29 câu bài tập thực hành với thư viện numpy theo [link notebook](#)

2. **Matrix Properties:** Các bạn chứng minh các tính chất của ma trận sau:

- (a) $\mathbf{AB} \neq \mathbf{BA}$, ($\mathbf{A} \in \mathbf{R}^{m \times m}$ and $\mathbf{B} \in \mathbf{R}^{m \times m}$) (\mathbf{A} và \mathbf{B} là ma trận vuông)
- (b) $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$, ($\mathbf{A} \in \mathbf{R}^{m \times n}$ and $\mathbf{B} \in \mathbf{R}^{m \times n}$)
- (c) $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$, ($\mathbf{A} \in \mathbf{R}^{m \times n}$ and $\mathbf{B} \in \mathbf{R}^{n \times k}$)

Example:

- $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$, ($\mathbf{A} \in \mathbf{R}^{m \times n}$ and $\mathbf{B} \in \mathbf{R}^{m \times n}$)

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots \\ b_{m1} & \dots & b_{mn} \end{bmatrix}$$

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} (a_{11} + b_{11}) & \dots & (a_{1n} + b_{1n}) \\ \dots & \dots & \dots \\ (a_{m1} + b_{m1}) & \dots & (a_{mn} + b_{mn}) \end{bmatrix} = \begin{bmatrix} (b_{11} + a_{11}) & \dots & (b_{1n} + a_{1n}) \\ \dots & \dots & \dots \\ (b_{m1} + a_{m1}) & \dots & (b_{mn} + a_{mn}) \end{bmatrix} = \mathbf{B} + \mathbf{A}$$

3. **Linear Regression:** Các bạn thực hiện train linear regression model trên tập data advertising.csv theo các yêu cầu sau. Các bạn sẽ dựa trên 3 thông tin đầu vào là TV, Radio, Newspaper để dự đoán Sale.

Giới thiệu về tập data: Data có 200 samples (rows), gồm 4 cột thông tin Tv, Radio, Newspaper, và Sales. Đề bài yêu cầu dùng thông tin ở 3 cột đầu tiên (Tv, Radio, Newspaper) để dự đoán được cột cuối cùng (Sale) dùng linear regression model.

(a) Stochastic Gradient Descent

- **input:** (4 inputs) X_b, y, n_epochs, learning_rate
- **output:** thetas_path, losses

(b) Mini Batch Gradient Descent

- **input:** (5 inputs) X_b, y, n_epochs, minibatch_size, learning_rate
- **output:** thetas_path, losses

(c) Batch Gradient Descent

- **input:** (4 inputs) X_b, y, n_epochs, learning_rate
- **output:** thetas_path, losses

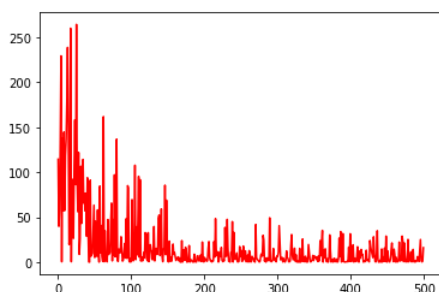
NOTE: YÊU CẦU CỦA ĐỀ BÀI LÀ PHẢI THỰC HIỆN THEO VECTORIZATION

- Các bạn thực hiện theo template ([link notebook](#) phần Linear Regression (Vectorization)) để thu được data sau khi normalize (X_b và y)
- X_b : là thông tin Tv, Radio, Newspaper (thông tin model nhận vào và sử dụng để predict Sale) (đã được normalize)
- y : là thông tin Sale (thông tin mong muốn model dự đoán đúng)
- n_epochs : Số lần train toàn bộ sample trong data
- $minibatch_size$: Số lượng sample sẽ được train trong 1 step (Chỉ sử dụng ở câu b)
- $learning_rate$: Tốc độ học
- $thetas_path$: List weights của model từ lúc khởi tạo cho đến sau mỗi lần cập nhật weights
- $losses$: List loss của mỗi step sau khi cập nhật
- Các bạn có thể thực hiện code mẫu bên dưới để biểu diễn loss

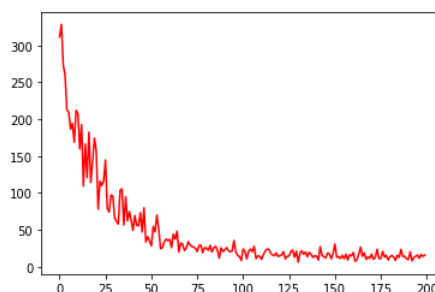
```

1 #Stochastic Gradient Descent
2 bgd_thetas, losses = stochastic_gradient_descent(X_b=X_b, y=y, n_epochs=50,
3         learning_rate=0.01)
4 x_axis = list(range(500))
5 plt.plot(x_axis, losses[:500], color="r")
6 plt.show()
7
8 #Mini Batch Gradient Descent
9 mbgd_thetas, losses = mini_batch_gradient_descent(X_b=X_b, y=y, n_epochs=50,
10         minibatch_size=20, learning_rate=0.01)
11 x_axis = list(range(200))
12 plt.plot(x_axis, losses[:200], color="r")
13 plt.show()
14
15 #Batch Gradient Descent
16 bgd_thetas, losses = batch_gradient_descent(X_b=X_b, y=y, n_epochs=100,
17         learning_rate=0.01)
18 x_axis = list(range(100))
19 plt.plot(x_axis, losses[:100], color="r")
20 plt.show()

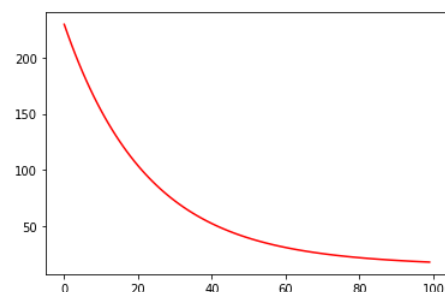
```



Stochastic Gradient Descent



Mini Batch Gradient Descent



Batch Gradient Descent

Hình 1: Stochastic Gradient Descent, Mini Batch Gradient Descent and Batch Gradient Descent Loss