FA17 10-701 Homework 2 Recitation 2

Logan Brooks,Matthew Oresky,Guoquan Zhao

October 2, 2017

Outline

Outline

How to Vectorize Code

Linear Regression Example

- The goal in vectorization is to write code that avoids loops and uses whole-array operations. Why?

How to Vectorize Code

Linear Regression Example

- The goal in vectorization is to write code that avoids loops and uses whole-array operations. Why?
- Efficient!
- Advanced algorithms to do computation on matrices and vectors.

Outline

Linear Regression Example

- Objective Function : L

Linear Regression Example

- Objective Function : L
- $\sum_{i=1}^{n}(X^{(i)^T}w - y_i)^2$
- $X^{(i)^T}$ is ith row of data. 1 x p
- w is a p x 1 column vector.
- y is a n x 1 column vector.
- $\frac{dL}{dw_j} =?$

Linear Regression Example

- Objective Function : L
- $\sum_{i=1}^{n}(X^{(i)^T}w - y_i)^2$
- $X^{(i)^T}$ is ith row of data. 1 x p
- w is a p x 1 column vector.
- y is a n x 1 column vector.
- $\frac{dL}{dw_j} = ?$
- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i)$

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T} w - y_i)$
- Octave Code:

```
1    dldw = zeros(p,1)
2    for j = 1:p
3        sum = 0
4        for i = 1:n
5            xw = 0
6            for k = 1:p
7                xw = xw + X(i,k) .* w(k)
8            sum = sum + 2 .* X(i,j) .* (xw - y(i))
9        dldw(j) = sum
```

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T} w - y_i)$
- Octave Code:

```
1    dldw = zeros(p,1)
2    for j = 1:p
3        sum = 0
4        for i = 1:n
5            xw = 0
6            for k = 1:p
7                xw = xw + X(i,k) .* w(k)
8            sum = sum + 2 .* X(i,j) .* (xw - y(i))
9        dldw(j) = sum
```

- How can we vectorize it?

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T} w - y_i)$
- Octave Code:

```
1    dldw = zeros(p,1)
2    for j = 1:p
3        sum = 0
4        for i = 1:n
5            xw = 0
6            for k = 1:p
7                xw = xw + X(i,k) .* w(k)
8            sum = sum + 2 .* X(i,j) .* (xw - y(i))
9        dldw(j) = sum
```

- How can we vectorize it?
- One way: Directly rewrite the formula. Remove the summation. We can easily remove one of the for loop to calculate xw because xw can be changed to $X(i,:) * w$.

Linear Regression Example

```
1    dldw = zeros(p,1)
2    for j = 1:p
3        sum = 0
4        for i = 1:n
5            sum = sum + 2 .* X(i,j) .* (X(i,:)*w - y(i))
6        dldw(j) = sum
```

- Sometimes it's not very intuitive.

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T} w - y_i)$

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i)$
- Another way: Build the matrix from the formula. We already know that if the formula can be rewritten in a matrix form, each entry can be calculated as the summation of some multiplication among a row and a column.

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T} w - y_i)$
- Another way: Build the matrix from the formula. We already know that if the formula can be rewritten in a matrix form, each entry can be calculated as the summation of some multiplication among a row and a column.
- Pretend that you know the result is given by matrix multiplication. Construct the matrix by figuring out which element should be put into which slot. Draw two empty matrices and fill in the blanks.

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i)$
- According to the formula, jth row of dL/dw can be written as the sum of multiplication of a row and a column. The jth row could be $[X_j^{(1)}, X_j^{(2)}, ..., X_j^{(n)}]$. And the column could be $[(X^{1^T}w - y1), (X^{2^T}w - y2), ..., (X^{n^T}w - yn)]^T$

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i)$
- According to the formula, jth row of dL/dw can be written as the sum of multiplication of a row and a column. The jth row could be $[X_j^{(1)}, X_j^{(2)}, ..., X_j^{(n)}]$. And the column could be $[(X^{1^T}w - y1), (X^{2^T}w - y2), ..., (X^{n^T}w - yn)]^T$
- If we stack each data point row by row,

$$X = \begin{bmatrix} X^{(1)^T} \\ X^{(2)^T} \\ \vdots \\ X^{(n)^T} \end{bmatrix} = \begin{bmatrix} X_1^{(1)}, X_2^{(1)}, \ldots, X_p^{(1)} \\ X_1^{(2)}, X_2^{(2)}, \ldots, X_p^{(2)} \\ \vdots \\ X_1^{(n)}, X_2^{(n)}, \ldots, X_p^{(n)} \end{bmatrix} \quad (1)$$

  We can see that jth row of X^T (or the jth column of X) is the same as the row that we construct above! So the first matrix should be X^T And the column could be calculated by $Xw - y$
- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i) = 2X^T(Xw - y)$

Linear Regression Example

- $\frac{dL}{dw_j} = \sum_{i=1}^{n} 2X_j^{(i)}(X^{(i)^T}w - y_i) = 2X^T(Xw - y)$

```
dldw = 2 * X' * (X*w -y)
```

- Use same method for logistic regression if you cannot pass the Autolab's autograder.

Question 1

Which of the following is true about regularization?

a) One of the goals of regularization is to combat underfitting.

b) The L0 norm is rarely used in practice because it is non-differentiable and non-convex.

c) A model with regularization fits the training data better than a model without regularization.

d) One way to understand regularization is that it makes the learning algorithm prefer "more complex" solutions.

Question 1

Which of the following is true about regularization?

a) One of the goals of regularization is to combat underfitting.

b) The L0 norm is rarely used in practice because it is non-differentiable and non-convex.

c) A model with regularization fits the training data better than a model without regularization.

d) One way to understand regularization is that it makes the learning algorithm prefer "more complex" solutions.

Answer

b. The goal of regularization is to combat overfitting, so a model without regularization will better fit the training data. Regularization makes the solution "simpler" by penalizing higher or non-zero weights.

Question 2

For a typical minimization problem with L^1 regularization:

$$min_\beta f(\beta) + \lambda||\beta||_1$$

where f is the objective function, which statement is true about λ?

a) Larger λ leads to a sparser solution of β and a possible underfit of the data.

b) Larger λ leads to a sparser solution of β and a possible overfit of the data.

c) Larger λ leads to a denser solution of β and a possible underfit of the data.

d) Larger λ leads to a denser solution of β and a possible overfit of the data.

Question 2

For a typical minimization problem with L^1 regularization:

$$min_\beta f(\beta) + \lambda||\beta||_1$$

where f is the objective function, which statement is true about λ?

a) Larger λ leads to a sparser solution of β and a possible underfit of the data.

b) Larger λ leads to a sparser solution of β and a possible overfit of the data.

c) Larger λ leads to a denser solution of β and a possible underfit of the data.

d) Larger λ leads to a denser solution of β and a possible overfit of the data.

Answer

a) Larger λ leads to a sparser solution of β and a possible underfit of the data.

Question 3

Consider a binary classification problem. Suppose I have trained a model on a linearly separable training set, and then I get a new labeled data point which is correctly classified by the model, and far away from the decision boundary. If I now append this new point to my training set and re-train, would the learned decision boundary change for the following models?

a) Perceptron

b) Logistic Regression (using gradient descent)

Question 3

Consider a binary classification problem. Suppose I have trained a model on a linearly separable training set, and then I get a new labeled data point which is correctly classified by the model, and far away from the decision boundary. If I now append this new point to my training set and re-train, would the learned decision boundary change for the following models?

a) Perceptron

b) Logistic Regression (using gradient descent)

Answer

a) No, if Perceptron has converged already, a correctly classified point will have no effect.

b) Yes, Logistic regression will consider this data point's gradient in its gradient calculations and this will effect the learned weight vector. (The weight vector likely won't change by a meaningful amount however.)

For Further Reading I

 https://www.gnu.org/software/octave/doc/v4.2.1/Basic-Vectorization.html
Octave Doc.