MLP implementation

This file recapitulate the notations and the dimensions of the matrix used in matlab, and explain the derivation of the forward and backward pass. To obtain the matlab name : => a2L.

## Forward pass

|  |  |  |
| --- | --- | --- |
| Description | **Notation** | **Dimension** |
| Dimension of the input | M (=576 by default) | Scalar |
| First layer dimension (L & R) | H1 | Scalar |
| 2nd layer dimension (L & LR &R) | H2 | Scalar |
| Left input vector | XL | Mx1 |
| Right input vector | XR | Mx1 |
| Weights for layer 1 | W1L, W1R | H1xM |
| Bias layer 1 | B1L, B1R | H1x1 |
| First layer activation | A1L, A1R | H1x1 |
| Non linear layer 1 | Z1L, Z1R | H1x1 |
| Non linear function 1 | g1 | function |

A1L=W1L\*XL+B1L A1R=W1R\*XR+B1R

Z1L=g1(A1L) Z1R=g1(A1R)

Note that for vectorization, I used the notation of the book.

|  |  |  |
| --- | --- | --- |
| Description | **Notation** | **Dimension** |
| Weights for layer 2 (1) | W2L, W2R | H2xH1 |
| Weights for layer 2 (2) | W2LR | H2x(2ˑH1) |
| Bias layer 2 | B2L, B2LR B2R | H2x1 |
| 2nd layer activation | A2L, A2LR, A2R | H2x1 |
| Non linear layer 2 | Z2 | H2x1 |
| Non linear function 2 | g2 | function |

A2L=W2L\*Z1L+B2L A2R=W2R\*Z1R+B2R

A2LR=W2LR\*[Z1L;Z2L]+B2LR

Z2=g2(A2L,A2R,A2LR)

|  |  |  |
| --- | --- | --- |
| Description | **Notation** | **Dimension** |
| Weights for layer 3 | W3 | 1xH2 |
| Bias layer 3 | B3 | Scalar |
| Output | A3 | Scalar |

A3=W3\*Z2+B3

## Backward pass

1. Definition of variables

Now let’s calculate the backward pass. For this let’s define the matlab variable we are looking for :

|  |  |  |
| --- | --- | --- |
| Description | **Notation** | **Dimension** |
| Error variable for layer 3 | r3 | Scalar |
| Error variable for layer 2 | r2L, r2LR, r2R | H2x1 |
| Error variable for layer 1 | r1L, r1R | H1x1 |
| Gradient along bias layer 3 | grad\_B3 | Scalar |
| Gradient along weights layer 3 | grad\_W3 | 1xH2 |
| Gradient along bias layer 2 | grad\_B2L, grad\_B2LR, grad\_B2R | H2x1 |
| Gradient along weights layer 2 (1) | grad\_W2L, grad\_W2R | H2xH1 |
| Gradient along weights layer 2 (2) | grad\_W2LR | H2x(2H1) |
| Gradient along bias layer 1 | grad\_B1L, grad\_B1LR | H1x1 |
| Gradient along weights layer 1 | grad\_W1L, grad\_W1R | H1xM |

1. Derivation
2. Third layer

This is the tough part. Let’s start with the third layer :

So now we have r3, that is a start. From there, using the formula in the course, we obtain easily :

and

So in matlab :

grad\_W3=r3\*Z2‘ and grad\_B3=r3

1. Second layer

At first, we are interested in

Let us remember the derivative of the sigmoid :

Then

Now, let us remember that :

So :

The same result is obtained for and . We can conclude with the following matlab code:

r2LR =r3.\*W3’.\*sigma(A2L).\*sigma(A2R)

r2R= r3.\*W3’.\*Z2.\*(1-sigma(A2R))

r2L= r3.\*W3’.\*Z2.\*(1-sigma(A2L))

Do not forget that r2LR, r2R, r2L are vectors….

Let’s now consider the gradient along the weights and bias of the second layer. At first notice that influences only via :

It comes :

It follows :

(which is, once again, a matrix).

Same thing can be obtained for R and LR :

Where

This means the following matlab equations :

grad\_W2L=r2L\*(Z1L)’

grad\_W2R=r2R\*(Z2R)‘

grad\_W2LR=r2LR\*([Z1L;Z1R]‘)

The result for the bias is obtained in the same way as for the third layer :

Same thing for R and LR. In matlab :

grad\_B2L=r2L

grad\_B2R=r2R

grad\_B2LR=r2LR

1. First layer

The computation of the errors for the first layer is harder.

influence through and also through .

Which means that for a given j :

Remember that for a given m:

Then

This leads to :

And we can conclude :

Where refers to the multiplication coordinate by coordinate. Notice that we take only half of the matrix .

This leads to the following matlab code :

gp1L=1-z1L.^2

gp1R=1-z1R.^2

r1L=gp1L.\*((r2L’\*w2L)’+(r2LR’\*w2LR(:,1:H1))’)

r1R=gp1R.\*((r2R’\*w2R)’+(r2LR’\*w2LR(:,H1+1:end))’)

Ouch, we have r1L and r1R. What remains is simple, since we can apply the equations of the second layer :

grad\_W1L=r1L\*(XL)’

grad\_W1R=r1R\*(XR)‘

and

grad\_B1L=r1L

grad\_B1R=r1R