# 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 : $a_L^{(2)}$ => 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)

$$g1(a) = \tanh(a)$$

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(2·H1) |
| 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)

$$g2(a_{LR}, a_R, a_R) = \frac{a_{LR}}{(1+e^{-a_L})(1+e^{-a_R})} = a_{LR}\sigma(a_L)\sigma(a_R)$$

| 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 |

2) Derivation

a. Third layer

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

$$E_i = \log(1 + e^{-t_i a^{(3)}})$$

$$\frac{\partial E_i}{\partial a^{(3)}} = -t_i e^{-t_i a^{(3)}} \sigma\left(t_i a^{(3)}\right) = r^{(3)}$$

$$where\ \sigma(x) = \frac{1}{1+e^{-x}}\ (the\ sigmoid)$$

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

$$\frac{\partial E_i}{\partial W^{(3)}} = r^{(3)} \cdot \left(z^{(2)}\right)^T \text{ and } \frac{\partial E_i}{\partial b^{(3)}} = r^{(3)}$$

So in matlab :

grad_W3=r3*Z2' and grad_B3=r3

b. Second layer

At first, we are interested in

$$g2(a_{LR}, a_R, a_R) = \frac{a_{LR}}{(1+e^{-a_L})(1+e^{-a_R})} = a_{LR}\sigma(a_L)\sigma(a_R)$$

Let us remember the derivative of the sigmoid :

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1-\sigma(x))$$

Then

$$\begin{cases} \dfrac{\partial g_2}{\partial a_{LR}} = \sigma(a_L)\sigma(a_R) \\ \dfrac{\partial g_2}{\partial a_L} = g2(a_{LR}, a_R, a_R) \cdot \left(1-\sigma(a_L)\right) \\ \dfrac{\partial g_2}{\partial a_R} = g2(a_{LR}, a_R, a_R) \cdot \left(1-\sigma(a_R)\right) \end{cases}$$

Now, let us remember that :

$$a^{(3)} = \sum_j W^{(3)}(j) g_2(a_{L,j}^{(2)}, a_{R,j}^{(2)}, a_{LR,j}^{(2)})$$

So :

$$r_{L,j}^{(2)} = \frac{\partial E_i}{\partial a_{L,j}^{(2)}} = \frac{\partial E_i}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial a_{L,j}^{(2)}} = r^{(3)} \frac{\partial a^{(3)}}{\partial a_{L,j}^{(2)}} = r^{(3)} W^{(3)}(j) z_j^{(2)} (1 - \sigma(a_{L,j}))$$

The same result is obtained for $a_R^{(2)}$ and $a_{LR}^{(2)}$ . 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 $W_L^{(2)}(j,k)$ influences $a^{(3)}$ only via $a_{L,j}^{(2)}$:

$$a_{L,j}^{(2)} = \sum_{k=1}^{H_1} W_L^{(2)}(j,k) z_{L,k}^{(1)} + b_{L,j}^{(2)}$$

It comes :

$$\frac{\partial E_i}{\partial W_L^{(2)}(j,k)} = \frac{\partial E_i}{\partial a_{L,j}^{(2)}} \frac{\partial a_{L,j}^{(2)}}{\partial W_L^{(2)}(j,k)} = r_{L,j}^{(2)} z_{L,k}^{(1)}$$

It follows :

$$\frac{\partial E_i}{\partial W_L^{(2)}} = r_L^{(2)} \left(z_L^{(1)}\right)^T$$

(which is, once again, a matrix).
Same thing can be obtained for R and LR :

$$\frac{\partial E_i}{\partial W_R^{(2)}} = r_R^{(2)} \left(z_R^{(1)}\right)^T$$
$$\frac{\partial E_i}{\partial W_{LR}^{(2)}} = r_{LR}^{(2)} \left(z_{LR}^{(1)}\right)^T$$

Where $z_{LR}^{(1)} = \begin{bmatrix} z_L^{(1)} \\ z_R^{(1)} \end{bmatrix}$

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 :

$$\frac{\partial E_i}{\partial b_L^{(2)}} = r_L^{(2)}$$

Same thing for R and LR. In matlab :

```
grad_B2L=r2L
grad_B2R=r2R
grad_B2LR=r2LR
```

c. First layer

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

$a_{L,j}^{(1)}$ influence $a^{(3)}$ through $a_{L,1}^{(2)} \dots a_{L,H_2}^{(2)}$ and also through $a_{LR,1}^{(2)} \dots a_{LR,H_2}^{(2)}$.

Which means that for a given j :

$$\frac{\partial E_i}{\partial a_{L,j}^{(1)}} = \frac{\partial E_i}{\partial a_{L,1}^{(2)}} \frac{\partial a_{L,1}^{(2)}}{\partial a_{L,j}^{(1)}} + \cdots + \frac{\partial E_i}{\partial a_{L,H_2}^{(2)}} \frac{\partial a_{L,H_2}^{(2)}}{\partial a_{L,j}^{(1)}} + \frac{\partial E_i}{\partial a_{LR,1}^{(2)}} \frac{\partial a_{LR,1}^{(2)}}{\partial a_{L,j}^{(1)}} + \cdots + \frac{\partial E_i}{\partial a_{LR,H_2}^{(2)}} \frac{\partial a_{LR,H_2}^{(2)}}{\partial a_{L,j}^{(1)}}$$

$$\frac{\partial E_i}{\partial a_{L,j}^{(1)}} = r(1)_L^{(2)} \frac{\partial a_{L,1}^{(2)}}{\partial a_{L,j}^{(1)}} + \cdots + r(H_2)_L^{(2)} \frac{\partial a_{L,H_2}^{(2)}}{\partial a_{L,j}^{(1)}} + r(1)_{LR}^{(2)} \frac{\partial a_{LR,1}^{(2)}}{\partial a_{L,j}^{(1)}} + \cdots + r(H_2)_{LR}^{(2)} \frac{\partial a_{LR,H_2}^{(2)}}{\partial a_{L,j}^{(1)}}$$

Remember that for a given m:

$$a_{L,m}^{(2)} = \sum_{n=1}^{H_1} W_L^{(2)}(m,n) g\left(a_{L,n}^{(1)}\right) + b_{L,m}^{(2)}$$

Then

$$\frac{\partial a_{L,m}^{(2)}}{\partial a_{L,j}^{(1)}} = W_L^{(2)}(m,j) g'\left(a_{L,j}^{(1)}\right)$$

This leads to :

$$\frac{\partial E_i}{\partial a_{L,j}^{(1)}} = \sum_{m=1}^{H_2} W_L^{(2)}(m,j) g_1'\left(a_{L,j}^{(1)}\right) r_L^{(2)}(m) + \sum_{m=1}^{H_2} W_{LR}^{(2)}(m,j) g_1'\left(a_{L,j}^{(1)}\right) r_{LR}^{(2)}(m)$$

$$\frac{\partial E_i}{\partial a_{L,j}^{(1)}} = g_1'\left(a_{L,j}^{(1)}\right) \left(r_L^{(2)}\right)^T \cdot W_L^{(2)}(:,j) + g_1'\left(a_{L,j}^{(1)}\right) \left(r_{LR}^{(2)}\right)^T \cdot W_{LR}^{(2)}(:,j)$$

And we can conclude :

$$r_L^{(1)} = g_1'\left(a_L^{(1)}\right) \therefore \left[\left(r_L^{(2)}\right)^T \cdot W_L^{(2)}\right]^T + g_1'\left(a_L^{(1)}\right) \therefore \left[\left(r_{LR}^{(2)}\right)^T \cdot W_{LR}^{(2)}(:,1:H_2)\right]^T$$

Where ∴ refers to the multiplication coordinate by coordinate. Notice that we take only half of the matrix $W_{LR}$.

This leads to the following matlab code :

```
gp1L=1-z1L.^2
gp1R=1-z1R.^2
r1L=gp1L.*((r2L'*w2L)'+(r2LR'*w2LR(:,1:H2))')
r1R=gp1R.*((r2R'*w2R)'+(r2LR'*w2LR(:,H2+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
```