Practical work 03 – 4/10/2018
Shallow Networks

Banfi Gregory (banfigre@students.zhaw.ch)
Benjamin Kühnis (bkuehnis@hsr.ch)

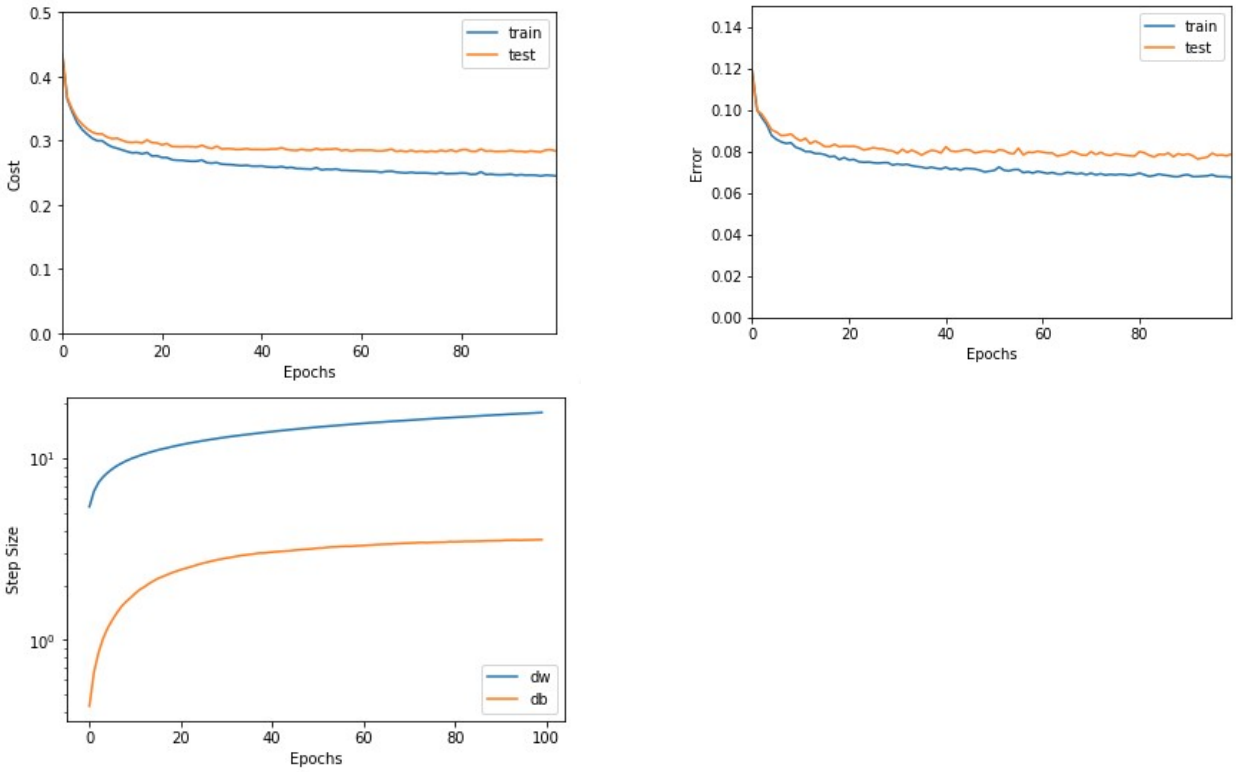
Exercice 1: **MBGD and Softmax**

1	learning rate: 0.6 epochs: 100 batch size: 500	`Training error / cost : 0.0671 / 0.2446` `Test error / cost : 0.0762 / 0.2828`
2	learning rate: 0.6 epochs: 100 batch size: 50	`Training error / cost : 0.0703 / 0.2469` `Test error / cost : 0.0902 / 0.3303`
3	learning rate: 0.2 epochs: 100 batch size: 50	`Training error / cost : 0.0651 / 0.2339` `Test error / cost : 0.0808 / 0.2911`
4	learning rate: 0.1 epochs: 100 batch size: 50	`Training error / cost : 0.0654 / 0.2377` `Test error / cost : 0.0771 / 0.2831`
5	learning rate: 0.1 epochs: 100 batch size: 100	`Training error / cost : 0.0676 / 0.2469` `Test error / cost : 0.0775 / 0.2825`
6	learning rate: 0.9 epochs: 100 batch size: 500	`Training error / cost : 0.0666 / 0.2403` `Test error / cost : 0.0794 / 0.2851`
7	learning rate: 2 epochs: 100 batch size: 500	`Training error / cost : 0.0689 / 0.2441` `Test error / cost : 0.0819 / 0.3031`
8	learning rate: 0.1 epochs: 100 batch size: 500	`Training error / cost : 0.0774 / 0.2769` `Test error / cost : 0.0831 / 0.2947`

We tried different values for the model. The best result we got using the parameters already presented in the exercise (first table's row). Anyway we noticed that we can have similar results with parameters values from table's row number 4, 5 and 6. The 4th row uses a smaller learning rate and batch size. If we do not decrease the batch size with a small learning rate we got something similar to the results of the last row. That is not so bad as result but it is actually the second worst of the entire table. The worst result we got is from a model with a big learning rate but a small batch size (row number 2). So we understand that the learning rate and the batch size go somehow proportional to each others in order to get the best result.

We did not play with the number of epochs because the result will be similar to the one get in the last exercise. Less iterations will fit better a large learning rate and a many iterations will fit good to a small learning rate, that moves slowly to the minimum.

Here the plot we got from our algorithm, the parameters were set as in row number 1.



Exercice 2: **Learning Function Representation**

$$J_{MSE}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - A_2)^2$$

$$\frac{dJ}{dw2}[J_{MSE}(\theta)] = \frac{1}{2m}\sum_{i=1}^{m}\frac{dJ}{dw2}[(y^{(i)} - A_2)^2]$$

$$= \frac{2}{2m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw2}[(y^{(i)} - A_2)]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw2}[(y^{(i)} - (\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}) + b_2))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw2}[(-(\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -\sigma'(w_{1,k}\cdot x^{(i)} + b_{1,k}))$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -(\sigma(z)\cdot(1 - \sigma(z)))$$

$$\mathrm{J}_{MSE}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - A_2)^2$$

$$\frac{dJ}{db2}[J_{MSE}(\theta)] = \frac{1}{2m}\sum_{i=1}^{m}\frac{dJ}{db2}[(y^{(i)} - A_2)^2]$$

$$= \frac{2}{2m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db2}[(y^{(i)} - A_2)]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db2}[(y^{(i)} - (\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}) + b_2))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db2}[(b_2)]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -1$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -1$$

$$\mathrm{J}_{MSE}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - A_2)^2$$

$$\frac{dJ}{dw1}[J_{MSE}(\theta)] = \frac{1}{2m}\sum_{i=1}^{m}\frac{dJ}{dw1}[(y^{(i)} - A_2)^2]$$

$$= \frac{2}{2m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw1}[(y^{(i)} - A_2)]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw1}[(y^{(i)} - (\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}) + b_2))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{dw1}[(-(\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot w_{2,k}\cdot -\sigma'(x^{(i)}))$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -(\sigma(x)\cdot(1 - \sigma(x)))$$

$$\mathrm{J}_{MSE}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - A_2)^2$$

$$\frac{dJ}{db1}[J_{MSE}(\theta)] = \frac{1}{2m}\sum_{i=1}^{m}\frac{dJ}{db1}[(y^{(i)} - A_2)^2]$$

$$= \frac{2}{2m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db1}[(y^{(i)} - A_2)]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db1}[(y^{(i)} - (\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}) + b_2))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot\frac{dJ}{db1}[(-(\sum_{k=1}^{n} w_{2,k}\cdot\sigma(w_{1,k}\cdot x^{(i)} + b_{1,k}))]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot w_{2,k}\cdot -\sigma'(1))$$

$$= \frac{1}{m}\sum_{i=1}^{m}\cdot(y^{(i)} - A_2)\cdot -(\sigma(1)\cdot(1-\sigma(1)))$$