

APA: Aprenentatge Automàtic (TEMA 7)

Grau en Enginyeria Informàtica - UPC (2017/18)

Lluís A. Belanche, belanche@cs.upc.edu

Entrega: 8 Gener 2018

Els problemes marcats **[G]** són de grup; els problemes/apartats marcats **[R]** són per fer-se en R

Objectius:

1. Conèixer l'arquitectura MLP de xarxes neuronals i saber-la ajustar a diferents tipus de problemes
2. Conèixer les funcions d'error més adequades segons el tipus de tasca

Problema 1 El problema d'aprenentatge contradictori 1 **[R,G]**

Sigui $\mathbb{B} = \{0, 1\}$. Suposem que tenim un conjunt d'exemples (INPUTS) d'entrada $\mathbf{x}_n \in \mathbb{B}^{10}$ que apareixen c_n cops ($1 \leq n \leq 4$) en una mostra de *training* de mida $N = 52$. Tenim també les corresponents sortides conegudes (TARGETS) $\mathbf{t}_n \in \mathbb{B}^7$:

	INPUTS	c_n	TARGETS
\mathbf{x}_1	1010011010	16	1,2,3,4,3,2,1
\mathbf{x}_2	1101011101	17	2,4,2,1,2,4,2
\mathbf{x}_3	0010100101	14	2,5,3,2,1,1,0
\mathbf{x}_4	0101010101	5	2,5,3,2,1,1,0

Els TARGETS expressen els nombre de '1' a cada posició (de la 1 a la 7) dels corresponents c_n INPUTS. Pel vector \mathbf{x}_4 pot haver més d'un '1' a la sortida; pels altres, hi ha exactament un '1' a la sortida (comproveu les sumes). Noteu que, a causa de les "contradiccions", l'error en el conjunt de *training* mai pot ser zero.

1. Doneu una interpretació a l'existència de les "contradiccions" a la mostra de dades
2. Volem resoldre el problema amb una xarxa neuronal MLP. Quantes neurones hi ha d'haver a la capa de sortida, quina funció d'error hauríem d'usar i quina funció d'activació?
3. Un cop correctament entrenada la xarxa, quin comportament podem esperar que mostri? En altres paraules, quina seria la sortida lògica per cadascun dels \mathbf{x}_n ?
4. Entreneu una xarxa MLP amb la rutina `nnet{nnet}` i reporteu els resultats obtinguts.
5. Calculeu l'error mínim possible (teòric) en les dades donades i compareu-lo amb l'error empíric obtingut.

.....

Problema 2 El problema de la paritat [R,G]

El problema de la paritat és una generalització de la XOR a d entrades binàries. La sortida ha de ser 1 si el número d'entrades a 1 és imparell i 0 si aquest número és parell.

1. Per què aquesta funció pot ser molt complexa d'aprendre? (no val dir simplement que no és separable linealment).
2. Construïu un escenari d'aprenentatge general per una d fixa, usant una xarxa MLP. Heu de decidir el tipus de tasca i la funció d'error. Imagineu que un petit percentatge (posem el 5%) de les sortides vénen amb la paritat erròniament calculada.
3. Entreneu una xarxa neuronal MLP amb la rutina `nnet{nnet}` per aprendre la tasca, en el cas de $d = 10$. Heu de fer tres estudis separats, prenent conjunts d'aprenentatge de mida creixent: 200, 500 i 1000. Caldrà que estimeu la millor arquitectura, cosa que podeu fer per *cross-validation*, usant regularització.
4. Reporteu els resultats de predicció dels tres estudis en un conjunt de test de mida 10^4 . Què observeu?

.....

Problema 3 El problema d'aprenentatge contradictori 2

Suposem que entrenem una xarxa MLP fent servir un únic exemple, que es repeteix 100 cops a la mostra de dades. La xarxa té una sola neurona de sortida, amb activació identitat. L'entrenament es fa de la següent manera: 80 de les vegades se li subministra un '1' a la sortida i les altres 20 un '0', fins que l'*error quadràtic* total és mínim.

1. Doneu una interpretació a l'existència de les "contradiccions" a la mostra de dades
2. Quina serà la sortida de la xarxa quan li presentem l'exemple, un cop acabat l'entrenament?
3. Quin és l'error mínim assolible (teòric)?

.....

Problema 4 Anàlisi de resultats amb la xarxa MLP

Un alumne ha estat jugant amb una xarxa MLP entrenada amb la rutina `nnet{nnet}`. Es tracta d'un problema de regressió bidimensional d'una sola sortida. Utilitzant exactament la mateixa crida a `nnet` tres vegades, tenim els següents resultats:

# weights: 21	# weights: 21	# weights: 21
initial value 143.78535	initial value 147.09410	initial value 140.510964
iter 10 value 82.67353	iter 10 value 70.01521	iter 10 value 87.626321
iter 20 value 7.343014	iter 20 value 34.64103	iter 20 value 33.779907
iter 30 value 3.130778	iter 30 value 21.04111	iter 30 value 7.576811
iter 40 value 2.817692	iter 40 value 6.781165	iter 40 value 7.027161
iter 50 value 2.803824	iter 50 value 1.729377 <skipped>
iter 60 value 2.781383	iter 60 value 1.051125	iter 100 value 6.448934
iter 70 value 2.743949	iter 70 value 0.725090	iter 110 value 6.448800
iter 80 value 2.310192	iter 80 value 0.318526	iter 120 value 6.448505
iter 90 value 0.148647	iter 90 value 0.136883	final value 6.446125
iter 100 value 0.031570	iter 100 value 0.116817	converged
iter 110 value 0.018951	iter 110 value 0.111052	
..... <skipped> <skipped>	
iter 860 value 0.000156	iter1940 value 0.001235	
iter 870 value 0.000155	iter1950 value 0.001235	
iter 880 value 0.000151	iter1960 value 0.001230	

```

iter 890 value 0.000145   iter1970 value 0.001225
iter 900 value 0.000087   iter1980 value 0.001212
final value 0.000087     iter1990 value 0.001206
converged                iter2000 value 0.001198
                           final value 0.001198
                           stopped after 2000 iterations

```

1. Comenteu completament tots aquests resultats. Per què tenim resultats diferents si hem fet la *mateixa* crida a `nnet`?
2. Quin és el nombre de neurones ocultes a la xarxa?
3. Raonar si és possible triar una d'aquestes xarxes com “millor” en funció dels resultats que veiem. En cas afirmatiu, expliqueu com; en cas negatiu, expliqueu com es faria.

.....

Problema 5 Pràctica amb la xarxa MLP 1 [R]

Aquesta és una tasca usada com a *benchmark* en la literatura. Definim $f : [-1, 1]^2 \rightarrow \mathbb{R}$ com:

$$f(x_1, x_2) = 4 \sin(\pi x_1) + 2 \cos(\pi x_2) + \epsilon$$

on $\epsilon \sim N(0, 0.5^2)$ és soroll normal amb mitjana zero i desviació estàndar 0.5.

1. Entreneu una xarxa neuronal MLP amb la rutina `nnet{nnet}` per aprendre la tasca. Heu de fer 4 estudis separats, prenent conjunts d'aprenentatge de mida creixent: 100, 200, 500 i 1000, mostrejats de manera aleatòria uniformement en $[-1, 1]^2$. Caldrà que estimeu la millor arquitectura, cosa que podeu fer per *cross-validation*, usant regularització.
2. Reporteu els resultats de predicció dels 4 estudis en un conjunt de test de mida 1024 obtingut de crear exemples a intervals regulars en $[-1, 1]^2$.
3. Repetiu els experiments usant regressió lineal amb i sense regularització en els mateixos conjunts de dades i compareu els resultats obtinguts amb els de la xarxa MLP; noteu que podeu usar simplement la rutina `nnet` amb `size=0`.

.....

Problema 6 Pràctica amb la xarxa MLP 2 [R]

Aquesta és també una tasca usada com a *benchmark* en la literatura. Definim $g : \mathbb{R} \rightarrow \mathbb{R}$ com

$$g(x) = \begin{cases} \sin(x)/x & \text{si } x \neq 0 \\ 1 & \text{si } x = 0 \end{cases}$$

i fem $f(x) = g(x) + \epsilon$, on $\epsilon \sim N(0, 0.1^2)$ és soroll normal amb mitjana zero i desviació estàndar 0.1.

1. Entreneu una xarxa neuronal MLP per aprendre la tasca. Heu de fer 3 estudis separats, prenent conjunts d'aprenentatge de mida creixent: 100, 200 i 500, mostrejats de manera aleatòria uniformement¹ en $[-20, 20]$. Caldrà que estimeu la millor arquitectura, així com els paràmetres de la xarxa, cosa que podeu fer per *cross-validation*, usant (o no) regularització.
2. Reporteu els resultats de predicció dels 5 estudis en un conjunt de test de mida 1000 obtingut de crear exemples a intervals regulars en $[-20, 20]$.

¹Noteu que la funció està ben definida, donat que $\lim_{x \rightarrow 0^+} \sin(x)/x = 1 = \lim_{x \rightarrow 0^-} \sin(x)/x$ i per tant la funció g és contínua a tot arreu.

3. Repetiu els experiments usant regressió lineal amb i sense regularització en els mateixos conjunts de dades i compareu els resultats obtinguts amb els de la xarxa MLP; noteu que podeu usar simplement la rutina `nnet` amb `size=0`.

.....

Problema 7 Reconeixement de lletres amb la xarxa MLP [R,G]

Prenem les lletres A, B, C, ..., Z codificades digitalment en una quadrícula de 7 x 5. Per exemple, A es codifica com:

```
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
```

L'arxiu `letters.txt` conté codificacions de 26 lletres, cadascuna representada com un vector de longitud 35. La tasca és dissenyar una xarxa neuronal que classifiqui imatges (possiblement corruptes) com a lletres de l'alfabet.

1. Dissenyeu una funció que generi versions corruptes d'una lletra, a còpia de canviar un cert número de bits de manera aleatòria. Una manera senzilla és generar primer el número de bits corruptes –p.e. amb una Poisson ($\lambda = 1.01$)– seleccionar els bits concrets (uniformement) i després invertir-los.
2. Dissenyeu una funció que, partint de les lletres netes (arxiu `letters.txt`), generi unes dades corruptes que usarem com a mostra de *training*, de mida N .
3. Entreneu una xarxa MLP per aprendre la tasca. Caldrà que estimeu la millor arquitectura, cosa que podeu fer per *cross-validation*, usant regularització.
4. Reporteu els resultats de predicció en una mostra de test gran –també generada per vosaltres, i de manera anàloga a la de *training*.

.....

Problema 8 Permeabilitat de roques amb la xarxa MLP [R]

Es disposa de 48 mesures de roques d'un dipòsit de petroli. L'objectiu és modelar la permeabilitat en funció de l'àrea, el perímetre i la forma. En primer lloc transformem les dades per ajudar a l'ajust del model:

```
library(datasets)
data(rock)
?rock

rock.x <- data.frame(area = scale(rock$area), perim = scale(rock$peri),
                     shape = scale(rock$shape))
rock.y <- log(rock$perm)
```

Entreneu una xarxa MLP per aprendre la tasca. Donat el baix número d'exemples, useu *leave-one-out cross-validation* i regularització per trobar la millor xarxa. Per avaluar el model, feu una gràfica de resposta predita vs. observada i guieu-vos per l'error quadràtic predictiu.

.....

Problema 9 Pràctica amb la xarxa MLP 3 [R, G?]

Aquesta tasca és la identificació d'un sistema no lineal continu SISO (*single-input/single-output*), a partir d'un número de punts obtinguts mostrejant consecutivament la dinàmica $u \rightarrow y$. Definim:

$$y(k) = y_1(k-1) + y_2(k-1)$$

on

$$\begin{aligned} y_1(k) &= 2.5y(k) \sin\left(\pi e^{-u^2(k)-y^2(k)}\right) \\ y_2(k) &= u(k)(1+u^2(k)) \end{aligned} \tag{1}$$

Així, la sortida $y(k)$ depèn de l'entrada anterior $u(k-1)$ i de la sortida anterior $y(k-1)$. Entreneu una xarxa MLP per aprendre la tasca. Preneu un conjunt d'aprenentatge de mida $N = 500$ construït posant $y(0) = 0$ i excitant aleatòriament el sistema usant un senyal $u(k)$ mostrejat uniformement en $[-2, 2]$. Caldrà que estimeu la millor arquitectura, cosa que podeu fer per *cross-validation*, usant regularització. **Nota:** aquest problema es pot fer tant en modalitat individual com grupal.

.....