Universitat Politècnica de Catalunya

Aprenentatge Automàtic

Reconeixement de lletres

Botet Colomer, Marc

González Sequeira, Enrique Alexandre

17 de gener 2018

Índex

1 Introducció **2**

2 Estudis previs **2**

3 Anàlisis de les dades **2**

4 Protocol de remostreig **4**

5 Predicció amb Models Lineals o Quadràtics **5**

5.1 K nearest neighbors 5

5.2 SVM quadràtic 6

5.3 LDA 8

5.4 QDA 8

6 Predicció amb models no lineals **8**

6.1 SVM amb RBF kernel 8

6.2 Random Forest 9

6.3 Xarxa neuronal amb una capa oculta 11

7 Model òptim **12**

8 Conclusions **13**

1 Introducció

L'objectiu d'aquest treball és reconèixer lletres a partir d'una sèrie de característiques. Per a dur a terme aquesta tasca tenim un conjunt de dades[4] amb vint mil exemples que inclou quina lletra és i 16 característiques per descriure tal lletra.

En primer lloc s'analitzarà el conjunt de dades per tal de saber la nostra distribució de les dades i poder comprovar que no hi hagi valors anormals o faltants, solucionant tals problemes si és necessari. També s'explicarà com s'han dividit les dades i quin criteri s'ha usat per avaluar els models i triar els *hiper-paràmetres* de cada model

Un cop estigui preparat els conjunt de dades s'intentarà predir, amb la màxima precisió possible, quina lletra correspon cada un dels exemples del conjunt de dades de prova. S'usarà diferents enfocaments per tal d'analitzar quin funciona millor. Primerament es provarà models lineals o quadràtics: QDA, LDA, SVM quadràtic. Seguidament es provarà els models no lineals: SVM amb RBF Kernel, Random Forest i xarxes neuronals amb una capa oculta. Finalment, es compararà tots els models i es triarà el model definitiu pel nostre.

2 Estudis previs

L'estudi del reconeixement de lletres és un tema tractat en multitud d'estudis. Aquest mateix conjunt de dades ha estat utilitzat en un conegut article *Letter Recognition Using Holland-style Adaptive Classifiers,1991* [1] on s'estudia les variacions dels classificadors d'estil Holland-style per aprendre a endevinar correctament les categories d'una lletra associada al vector de 16 atributs enters extrets d'escanejar imatges de les lletres. La millor precisió obtinguda va estar per sobre de un 80% utilitzant els classificadors Holland-style, en aquest treball es comprovarà si utilitzant models més populars als del article es pot aconseguir una precisió superior.

3 Anàlisis de les dades

En aquest treball hi ha un conjunt de dades de vint mil mostres sobre el reconeixement de lletres en una caixa rectangular. La naturalesa de les variables són numèriques i són nombres enters, les variables són factors al reconèixer la lletra com per exemple, valor horitzontal i vertical a la caixa, mitjana de píxels a la caixa, variància.

```
       V2              V3              V4              V5              V6
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 3.000   1st Qu.: 5.000   1st Qu.: 4.000   1st Qu.: 4.000   1st Qu.: 2.000
 Median : 4.000   Median : 7.000   Median : 5.000   Median : 6.000   Median : 3.000
 Mean   : 4.024   Mean   : 7.035   Mean   : 5.122   Mean   : 5.372   Mean   : 3.506
 3rd Qu.: 5.000   3rd Qu.: 9.000   3rd Qu.: 6.000   3rd Qu.: 7.000   3rd Qu.: 5.000
 Max.   :15.000   Max.   :15.000   Max.   :15.000   Max.   :15.000   Max.   :15.000

       V7              V8              V9              V10             V11
 Min.   : 0.000   Min.   : 0.0   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 6.000   1st Qu.: 6.0   1st Qu.: 3.000   1st Qu.: 4.000   1st Qu.: 7.000
 Median : 7.000   Median : 7.0   Median : 4.000   Median : 5.000   Median : 8.000
 Mean   : 6.898   Mean   : 7.5   Mean   : 4.629   Mean   : 5.179   Mean   : 8.282
```

```
3rd Qu.: 8.000   3rd Qu.: 9.0   3rd Qu.: 6.000   3rd Qu.: 7.000   3rd Qu.:10.000
Max.   :15.000   Max.   :15.0   Max.   :15.000   Max.   :15.000   Max.   :15.000

      V12              V13              V14              V15              V16
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
1st Qu.: 5.000   1st Qu.: 7.000   1st Qu.: 1.000   1st Qu.: 8.000   1st Qu.: 2.000
Median : 6.000   Median : 8.000   Median : 3.000   Median : 8.000   Median : 3.000
Mean   : 6.454   Mean   : 7.929   Mean   : 3.046   Mean   : 8.339   Mean   : 3.692
3rd Qu.: 8.000   3rd Qu.: 9.000   3rd Qu.: 4.000   3rd Qu.: 9.000   3rd Qu.: 5.000
Max.   :15.000   Max.   :15.000   Max.   :15.000   Max.   :15.000   Max.   :15.000

     V17
Min.   : 0.000
1st Qu.: 7.000
Median : 8.000
Mean   : 7.801
3rd Qu.: 9.000
Max.   :15.000
```

Les variables són:

- **V1-lettr:** lletra majuscula
- **V2-xbox:** posició horitzontal de la caixa
- **V3-ybox:** posició vertical de la caixa
- **V4 width:** ample de la caixa
- **V5-high:** altura de la caixa
- **v6-onpix:** nombre total de píxels
- **v7-xbar:** mitjana x dels píxels de la caixa
- **v8-ybar:** mitjana y dels píxels de la caixa
- **v9-x2bar** mitjana de la variancia de x
- **v10-y2bar** mitjana de la variancia de y
- **v11-xybar** mitjana de la correlació x i y
- **v12-x2ybr** mitjana de *x i *y
- **v13-xy2br** mitjana de x * y* y
- **v14-xege** mitjana del número d'arestes d'esquerra a dreta
- **v15-xegvy** correlació de x-ege amb y
- **v16-yege** mitjana del número d'arestes de baix a adalt.
- **v17-yegvx** correlació de y-ege amb x

En analitzar les dades es veu que tots el valors de les variables van entre 0 i 15, que no hi han valors faltants, com ve surt a la descripció del conjunt de dades, i a més a més fent un sumari de les dades es pot observar que efectivament tots els valors són entre 0 i 15 i no hi han valors anormals. A més la distribució de les lletres reconegudes és uniforme i, per tant, es pot dir que el conjunt de dades està ben distribuït en el nombre de mostres per cada lletra del alfabet.

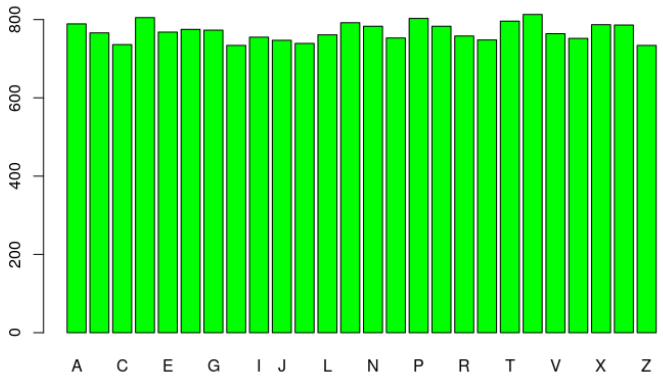

Figura 1: Distribució de les mostres en les lletres predites.

4 Protocol de remostreig

Per tal de poder evaluar els diferents models de la manera més equitativa possible s'ha dividit el conjunt de dades en dos subconjunts d'entrenament i de prova. Per fer-ho s'ha dividit el conjunt original mantenint la proporcionalitat de la variable V1 (lletra de l'alfabet que es reconeix) utilitzant un 70% de les mostres per al subconjunt de entrenament i el 30% restant pel subconjunt de prova. Per a tots els models s'utilitzen els mateixos conjunts per entrenar i provar per tal de mantenir la coherència.

Per tal d'ajustar els paràmetres òptims de cada model sense caure en un sobreajust, hem fet servir *Cross-Validation*, per a cada model hem utilitzat una variant de *Cross-Validation* que s'ajustés millor, per exemple per a models més complexes hem utilitzat *5-fold Cross-Validation* i en models més simples *10-fold Cross Validation* per ajustar el temps d'execució. Altrament, en altres hem utilitzat *Leave-One-Out Cross-Validation*. Per utilitzar *Cross-Validation* utilitzem la llibreria de R *caret*. [2]

Durant la resta del treball prendrem com a mesura de l'error i de precisió com les respectives fórmules:

$$\text{error} = 1 - (\frac{\text{observacions correctes}}{\text{observacions totals}})$$

$$\text{precisió} = (\frac{\text{observacions correctes}}{\text{observacions totals}})$$

5 Predicció amb Models Lineals o Quadràtics

En primer lloc es discutirà els models lineals o quadràtics que s'ha emprat en aquesta pràctica: K nearest neighbors, LDA , QDA i SVM quadratic.

5.1 K nearest neighbors

El paràmetre principal per a KNN és k, el nombre de veïns que ha de tenir en compte l'algorisme. Per trobar la k més precisa s'ha usat *Leave-One-Out Cross-Validation*(LOOCV) per provar quin valor de k s'ajusta millor al model i dóna un error menor.

k	Accuracy	Kappa
1	**0,9541919**	**0,9523578**
2	0,9447021	0,9424881
3	0,9494827	0,9474599
4	0,9481270	0,9460500
5	0,9490546	0,9470148
6	0,9451302	0,9429332
7	0,9419194	0,9395936
8	0,9402783	0,9378867
9	0,9384945	0,9360312
10	0.9372815	0.9347697
11	0.9349269	0.9323209
12	0.9326436	0.9299463
13	0.9305744	0.9277941
14	0.9293614	0.9265327
15	0.9284338	0.9255680

Taula 1: Precisió de diferents valors de k utilitzant LOOCV.

Com es pot veure a la taula 2 el valor que dóna un error menor és quan el valor de k és 1. Un cop ajustat el valor de k s'entrena el model i es comprova el seu error amb el conjunt de prova. Els resultats obtinguts de precisió 95.42%, és a dir, un error de 4.58%. En la taula **??** la matriu de confusió.

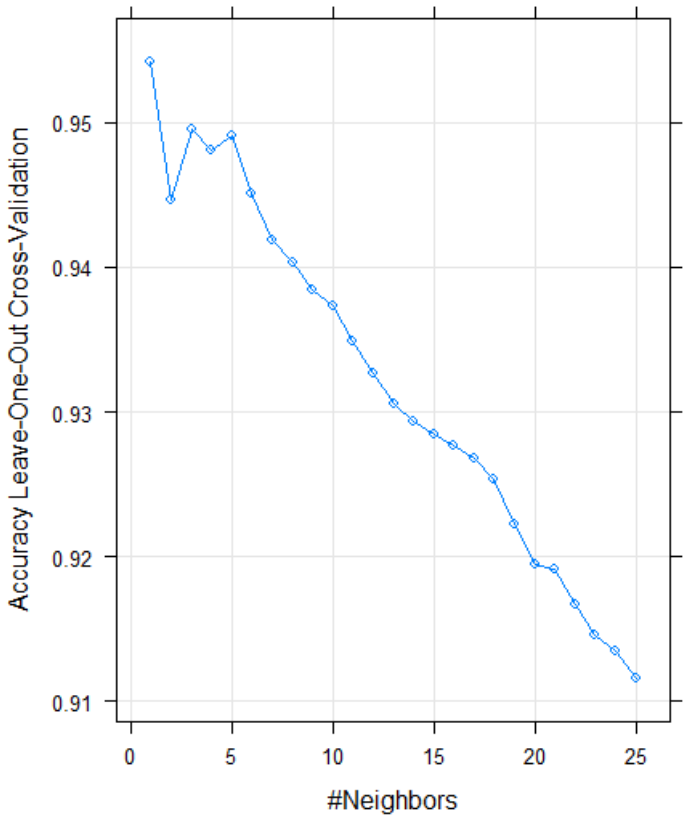

Figura 2: Precisió de CV en funció de k

Tal com es pot veure a la Figura 2 a mesura que s'augmenta k la precisió del model disminueix progressivament, excepte pels valors de 3 a 7 que té més variància encara que el punt amb més precisió és $k = 1$.

5.2 SVM quadràtic

Per a classificar mitjançant Màquines de Vectors de Suport(SVM) s'ha escalat les dades, s'utilitza *5-fold Cross-Validation* per tal de trobar el millor valor pel paràmetre C, utilitzant un grau de 2 per tal de que sigui SVM quadràtic. El paràmetre C serveix per reduir el sobreajust. Usem a *Cross-Validation* els possibles valors de C com: 0.01,0.1,1,10,100. Els resultats obtinguts són els següents:

C	Accuracy	Kappa
0.01	0.9360691	0.9335089
0.1	**0.9563326**	**0.9545840**
1	0.9528362	0,9509477
10	0,9513383	0,9493899
100	0,9513383	0,9493899
1000	0,9513383	0,9493899

Taula 2: Evolució de la precisió de diferents valors de C utilitzant 5-fold Cross-Validation.

Com es pot observar el valor de C que ens dóna *Cross-Validation* com a optim és 0.1 amb molt poca diferencia amb 1 (diferencia en el quart decimal). Un cop ajustat el valor de C es prediu el conjunt de prova amb la C abans trobada, el resultat obtingut és d'una precisió de 95.86% que vol dir un error de 4.14%. A continuació es mostra la matriu de confusió utilitzada per calcular l'error.

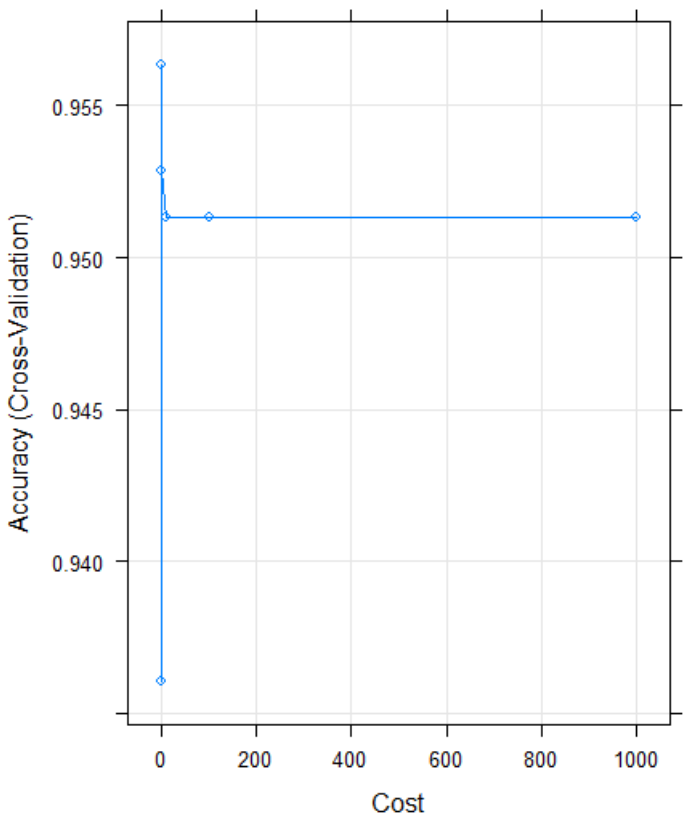

Figura 3: Evolució de la precisió de CV en funció de C

5.3 LDA

En aquesta secció s'obté un model LDA i a continuació es comprova el model mitjançant *Leave-One-Out Cross-Validation*(LOOCV). L'error de LOOCV és de 29.80%. Seguidament es mostra la matriu de confusió utilitzada per calcular l'error de prova.

El model aconsegueix un 70.04% d'encert en el conjunt de dades de prova.

5.4 QDA

Per QDA també utilitzem *Leave-One-Out Cross-Validation*(LOOCV) per tal de calcular el model òptim. L'error de LOOCV és de 11.54477. A continuació es mostra la respectiva matriu de confusió al calcular l'error de test.

El model aconsegueix un 88.47% d'encert en el conjunt de dades de prova.

Després de veure els resultats de LDA i QDA és hora de comparar-los. El mètode quadràtic és més precís que el lineal amb un error de 11.52% contra un error de 29.80%.

6 Predicció amb models no lineals

6.1 SVM amb RBF kernel

En aquesta secció s'explicarà el desenvolupament del mètode de Màquines de Vectos de Suport (SVM en anglès) usant el kernel *Radial Basis Function* per tal de tenir un model no lineal.

Aquest mètode necessita dos paràmetres, C amb l'objectiu de reduir el sobreajust i σ. Per tal d'establir la millor combinació d'ambdós paràmetres s'ha usat *5-fold Cross-Validation* en el test d'entrenament tal com s'ha explicat a la secció 4. La decisió d'usar *5-fold* i no un altre mètode més precís (ja sigui una k més gran o repetir el procés x vegades) ha estat perquè el temps d'execució d'aquest procés ja era bastant elevat. Per realitzar aquest mètode s'ha escalat els valors d'ambdós conjunts de dades usant la funció `scale()` de R.

S'ha decidit usar el següent conjunt de valors de C i de σ.

$$C : 0.01, 0.1, 1, 10, 100, 1000$$

$$\sigma : 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4$$

C	sigma	Accuracy	Kappa	AccuracySD	KappaSD
0.01	0.0312	0.3740	0.3482	0.0051	0.0053
0.01	0.0625	0.4634	0.4415	0.0044	0.0046
0.10	2.0000	0.0953	0.0570	0.0056	0.0058
0.10	4.0000	0.0466	0.0062	0.0010	0.0010
1.00	0.1250	0.9521	0.9501	0.0026	0.0027
1.00	0.2500	0.9609	0.9593	0.0023	0.0023
10.00	0.0312	0.9510	0.9490	0.0022	0.0023
10.00	4.0000	0.5496	0.5311	0.0057	0.0059
100.00	**0.1250**	**0.9680**	**0.9668**	**0.0012**	**0.0012**
100.00	2.0000	0.8129	0.8054	0.0082	0.0086
100.00	4.0000	0.5496	0.5311	0.0057	0.0059
1000.00	1.0000	0.9296	0.9268	0.0030	0.0031
1000.00	4.0000	0.5496	0.5311	0.0057	0.0059

Taula 3: Subconjunt dels resultats de *5-fold Cross-Validation* usant C i σ

Com es pot observar en la taula 3 els millors paràmetres pel model són: $C = 100$ i $\sigma = 0.1250$ que s'obté un encert del 96.8% en *Cross-Validation*. En la figura 4 es pot observar l'evolució de l'encert en la predicció en funció dels diferents valors de C i σ.

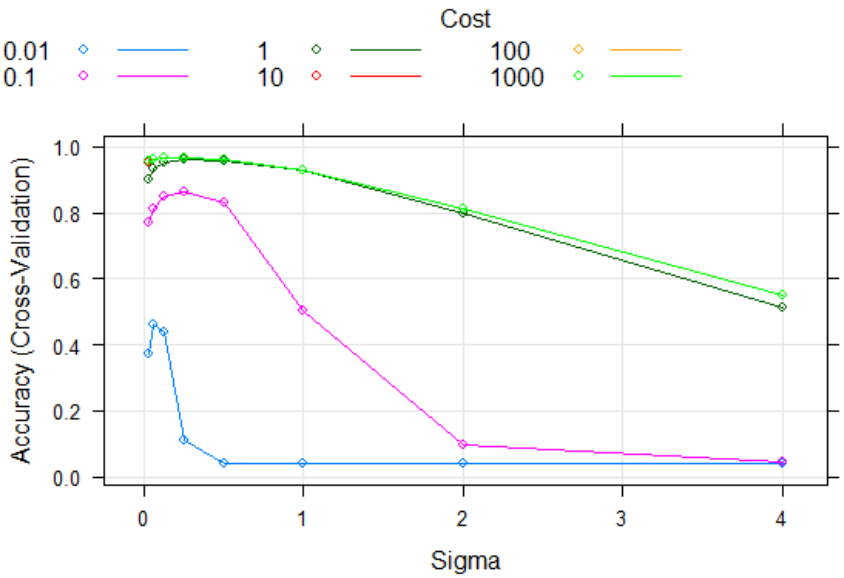

Figura 4: Evolució de la predicció de Cross-Validation en funció de C i σ

El model aconsegueix un 97.54% d'encert en el conjunt de dades de prova.

6.2 Random Forest

En aquesta secció s'explicarà el desenvolupament del mètode de *Random Forest*[3]. Per a trobar el nombre d'arbres adequat pel nostre conjunt de dades s'ha usat diferents nombres d'arbres entre 50 i 1000 per entrenar i s'ha escollit el nombre d'arbres que dóna un *Out-of-bag* menor. En la

figura 5 es pot observar l'evolució del nombre de *Out-of-bag*, nombre d'errors, en funció del nombre d'arbres usat. S'aconsegueix l'error més petit usant 900 arbres amb un encert d'entrenament del 96.1%.

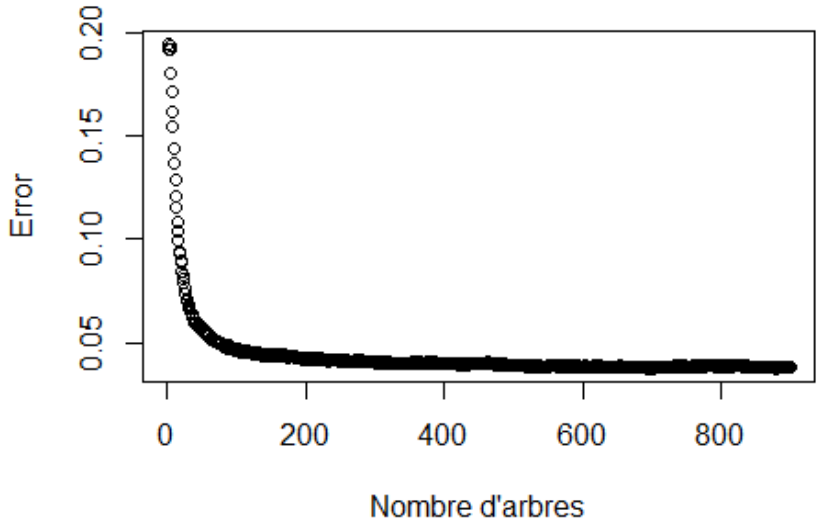

Figura 5: Evolució del nombre de *Out-of-bag*

Un avantatge que ofereix el mètode de *Random Forest* és que mostra la importància de cada variable en el model. Com es pot observar en la figura 6 la variable més important en el nostre model és la número 14, la qual correspon al número mitjà d'arestes de dreta a esquerra. Per l'altre banda, la variable menys important correspon a la número 5, l'alçada de la capsa a on és la figura.

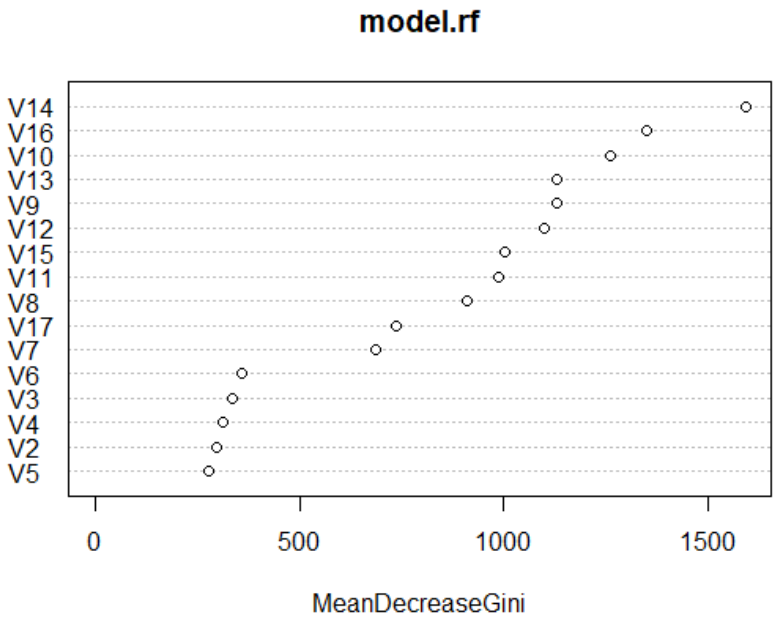

Figura 6: Importància de cada variable

El model aconsegueix un 96.27% d'encert en el conjunt de dades de prova. També el model obté un 0.9847 de *F1 score* aquest valor tan pròxim a 1 indica que la distribució de la predicció està ben balancejada entre les 26 classes diferents.

6.3 Xarxa neuronal amb una capa oculta

En aquesta secció s'explicarà el desenvolupament del mètode de xarxes neuronals, en concret s'usarà el mètode de perceptró multicapa amb una capa oculta.

Primer de tot es necessita determinar el nombre de neurones que disposarà la capa oculta. Sabem que la capa d'entrada tindrà 16 neurones, una per cada característica, i la capa de sortida 26 neurones, una per cada classe. Per la capa oculta usarem *5-fold Cross-Validation* per tal de determinar el nombre de neurones a usar. De la mateixa manera, per tal d'evitar el sobreajust en el conjunt de dades d'entrenament i que el nostre model pugui generalitzar millor, usarem regularització. Per tant, en realitzar *Cross-Validation* trobarem la millor combinació de neurones juntament amb el seu valor de regularització.

Com és possible observar a la taula 4 els millors paràmetres són 100 neurones en la capa oculta usant el valor 0.1 per la regularització. A la figura 7 es pot observar com per la majoria de valors de regularització segueix una evolució semblant, obtenint una empitjora dels resultats a partir de 40 neurones sense usar regularització, probablement perquè sense la regularització sobreajusta.

size	decay	Accuracy	Kappa	AccuracySD	KappaSD
28	0.50	0.8895	0.8851	0.0044	0.0046
28	1.00	0.8801	0.8753	0.0078	0.0082
40	0.00	0.8986	0.8946	0.0108	0.0113
70	0.00	0.9025	0.8986	0.0080	0.0083
70	0.01	0.9318	0.9291	0.0071	0.0074
82	0.50	0.9496	0.9476	0.0033	0.0035
82	1.00	0.9385	0.9360	0.0072	0.0075
94	0.50	0.9514	0.9495	0.0040	0.0042
94	1.00	0.9368	0.9343	0.0067	0.0070
100	0.00	0.9124	0.9089	0.0099	0.0103
100	**0.10**	**0.9567**	**0.9550**	**0.0046**	**0.0048**
100	0.50	0.9507	0.9487	0.0066	0.0068

Taula 4: Subconjunt dels resultats de *5-fold Cross-Validation*

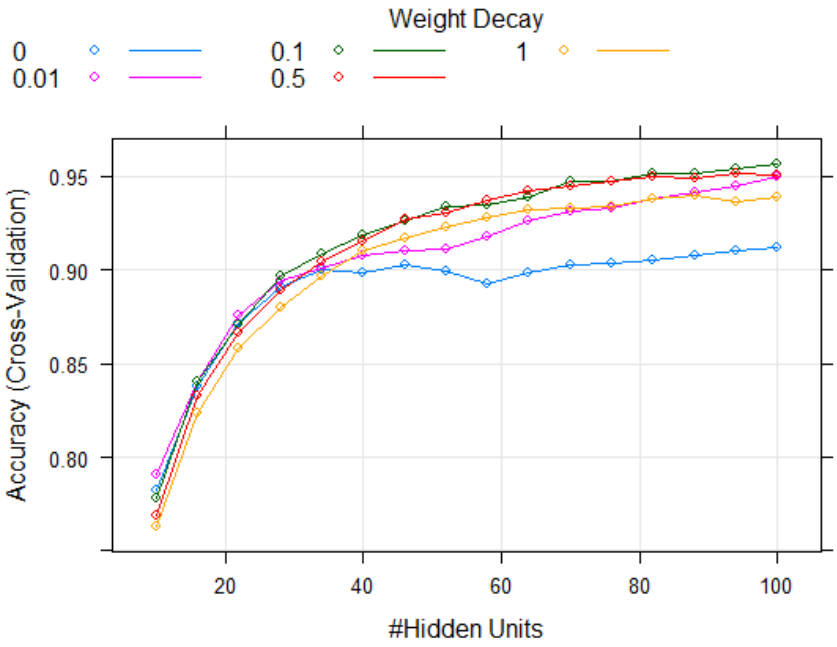

Figura 7: Evolució de la predicció de *Cross-Validation*

El model aconsegueix un 95.32% d'encert en el conjunt de dades de prova.

7 Model òptim

Un cop vistos tots els models es pot veure un resum en aquesta taula.

	Mètode	Error entrenament	Error validació
Lineals	QDA	88.45 %	88.47 %
	LDA	70.19 i %	70.19 %
	KNN	95.41 %	95.41 %
	SVM Quadràtic	95.63 %	95.85 %
No Lineals	**SVM Radia**l	**96.80** %	**97.54** %
	Random Forest	96.10%	96.27%
	Xarxes neuronals	95.67%	95.32%

Taula 5: Mètode utilitzat i precisió i validació obtinguts per a la predicció de lletras.

Com es possible observar a la taula 5 el model que dóna millors resultats tant en precisió d'entrenament com precisió de validació és *SVM Radial* amb un 97.54, un resultat excel·lent. En la taula 6 es pot observar la matriu de confusió la qual, gràcies al alt percentatge d'encert del nostre model la majoria de números estan a la diagonal.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	235	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	221	0	0	3	1	0	1	1	0	1	1	0	0	0	1	0	2	1	0	0	2	0	0	0	0
C	0	0	215	0	2	0	2	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
D	0	0	0	239	0	0	1	7	0	1	0	0	0	3	0	0	0	1	0	0	0	0	0	1	0	0
E	0	0	2	0	218	1	2	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
F	0	0	0	0	0	227	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	1	0	2	0	0
G	0	0	2	0	1	0	225	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
H	0	1	0	0	0	0	0	202	0	1	1	1	0	0	0	1	0	0	0	0	2	0	0	0	0	0
I	0	0	0	0	0	0	0	0	220	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	3	214	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
K	0	1	0	0	1	0	0	4	0	0	215	0	0	0	0	0	0	4	0	1	0	0	0	2	0	0
L	0	0	0	0	2	0	0	0	0	0	0	224	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	1	0	0	0	0	0	0	0	0	0	0	236	0	0	0	0	0	0	0	2	1	1	0	0	0
N	0	0	0	1	0	0	0	0	0	0	0	0	0	228	1	0	0	4	0	0	1	0	0	0	0	0
O	0	0	1	1	0	0	0	1	0	0	0	0	0	0	219	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	232	2	0	0	1	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	230	0	0	0	0	0	0	0	0	0
R	0	2	0	0	0	0	0	2	0	0	3	1	0	2	0	0	0	214	0	0	0	0	0	0	0	1
S	0	0	0	0	2	1	0	0	0	0	0	1	0	0	0	0	0	0	223	0	0	0	0	0	0	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	0	0	0	0	1	0
U	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	238	0	0	0	0	0
V	0	3	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	223	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	224	0	0	0
X	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	230	0	0
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	234	0
Z	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	217

Taula 6: Matriu de confusió del conjunt de dades de prova predit pel model SVM RBF

8 Conclusions

A l'hora enfrontar la complexitat de la predicció del reconeixement de lletres es pot concloure que hi ha uns quants models que aconsegueixen sobrepassar el 80% de precisió que s'aconseguia en l'article[1] mencionat. Clarament es pot inferir que els mètodes no lineals donen millors resultats per norma general que els lineals, encara que, com s'ha vist anteriorment, hi ha alguns mètodes no lineals que donen resultants molt precisos superiors al 90% com és el cas de SVM quadràtic.

Per la naturalesa del nostre conjunt de dades i la seva simplicitat, que no necessita cap tipus de preprocessat potser ens ha limitat a l'hora de poder explotar al màxim tots els coneixements estudiats en classe com seria el cas de reducció de dimensionalitat i tractament de valors anormals i/o faltants.

En un principi es van mirar altres mètodes per treballar com Naive Bayes, però pels seus mal resultats i poca comoditat per treballar amb aquests models es van acabar descartant.

Aquest treball ens ha servit per introduir-nos en el món de l'aprenentatge automàtic, un món del qual n'hem sentit a parlar molt. Ens ha permès donar-nos compte de les dificultats i limitacions d'aquesta disciplina com, per exemple, la gran quantitat de poder computacional que es necessita per treballar en conjunts de dades molt grans, ja que en els nostres ordinadors tardava hores només tenint vint mil exemples. Però tanmateix ens ha fet reflexionar sobre la multitud d'aplicacions en el món real que poden estalviar-nos una gran quantitat de temps i millorar la nostra vida.

Una possible extensió d'aquest treball seria estudiar el comportament de totes les variables a l'hora de predir una lletra, i estudiar quina de les 16 variables és més útil i quina ho és menys per a arribar a la conclusió. També es podria comparar els resultats amb nous enfocaments de reconeixement de lletres que no usen característiques concretes com les que teníem en el nostre conjunt de dades, sinó que la seva entrada són tots els píxels de la imatge.

Referències

[1] Peter W Frey i David J Slate. "Letter Recognition Using Holland-Style Adaptive Classifiers". A: *Machine Learning* 6 (1991), pàg. 161.

[2] Max Kuhn. *Caret package*. 2017. URL: https://cran.r-project.org/web/packages/caret/caret.pdf.

[3] Fortran original by Leo Breiman, R port by Andy Liaw Adele Cutler i Matthew Wiener. *Package RandomForest*. 2015. URL: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf.

[4] David J. Slate. *Letter Recognition Data Set*. 1991. URL: https://archive.ics.uci.edu/ml/datasets/letter+recognition.