Optimització Matemàtica:

Pattern recognition with Single Layer Neuronal Network

Marina Rosell Murillo, Pau Lozano García

Prof. Francisco Javier Heredia

Grau en Ciència i Enginyeria de Dades, UPC

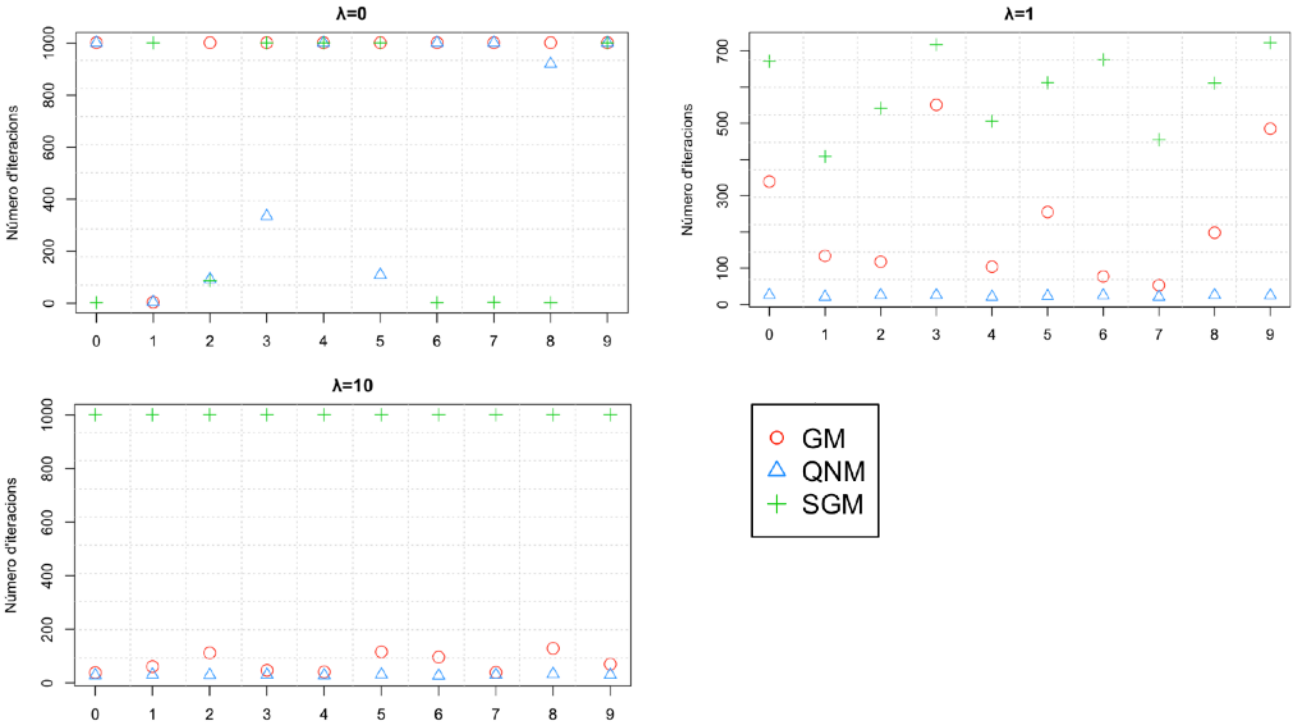
2019/2020

Index:

1. Convergence of the algorithms:

a) Global convergence

En primer lloc, per determinar quins dels algorismes tenen convergència global mirarem quantes iteracions fa cada mètode fins a arribar a l'òptim de la funció objectiu. El límit màxim d'iteracions que hem fixat pels algorismes és 1000, per tant, considerarem que els algorismes que no aconsegueixin convergir dins d'aquest rang d'iteracions, no tenen convergència global.



La convergència dels mètodes està força lligada amb el valor de λ. Aquest paràmetre té una funció regularitzadora a la funció objectiu, que multiplica el quadrat de la norma dels paràmetres, cosa que farà que la funció s'assembli més o menys a una funció quadràtica. Per això, influeix a la convergència dels algorismes.

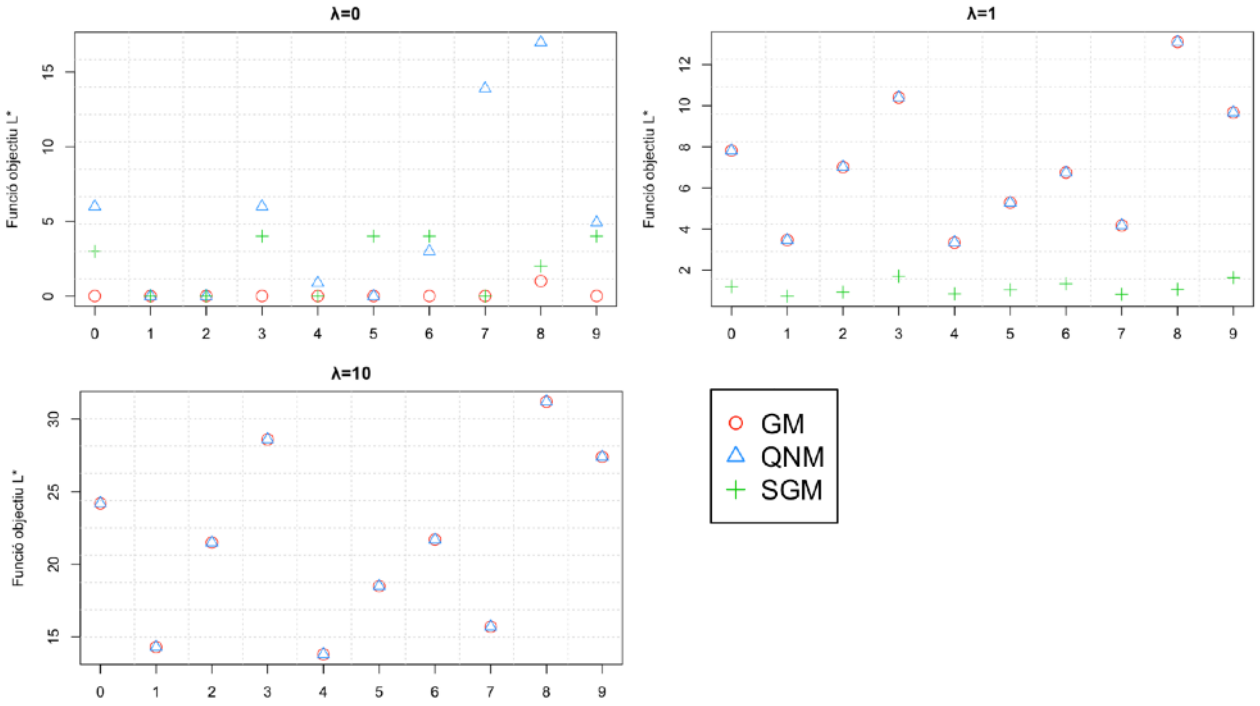
$$\tilde{L}(X^{TR}, y^{TR}, \lambda) = L(w; X^{TR}, y^{TR}, \lambda) + \lambda \cdot \frac{\|w\|^2}{2}$$

En el cas que $\lambda = 0$, cap cas del mètode del Gradient convergeix, només hi ha una excepció al número 1, que triga tan sols 4 iteracions, el mètode Quasi newton fa exactament el mateix, deduïm que és degut a que el punt inicial és molt aprop del mínim, i per tant, és tracta d'una coincidència. Del mètode Quasi Newton i del Gradient Conjugat convergeixen la meitat dels casos.

En el cas que $\lambda = 1$, tots el casos amb tots els algorismes convergeixen. Amb l'algorisme de Quasi Newton convergeixen en molt poques iteracions, totes al voltant de 25. Amb el mètode del Gradient, en fa una mica més, i el mètode de Gradient Estocàstic és el que més iteracions fa.

Per últim, quan $\lambda = 10$, el mètode del Gradient Estocàstic no en convergeix en cap cas, però en canvi els mètodes del Gradient i Quasi Newton convergeixen en molt poques iteracions, tot i que sempre en fa menys el Quasi Newton.

Hem volgut comprovar, per cada número, si tots els algorismes que convergeixen ho fan al mateix punt, i hem obtingut el següent gràfic:



Per a $\lambda = 10$, quan fem servir el mètode del Gradient Estocàstic, la funció objectiu tendeix a infinit. Això és degut a que a l'algorisme de resolució no apliquem el BLS amb les Wolfe Conditions, i tot i que la direcció és de descens, la longitud de pas és massa gran i fa que la funció objectiu empitjori molt en cada pas, fins que se'n va a l'infinit positiu, per això no apareix a les gràfiques.

Veiem que com més gran és λ, en general, més gran és el valor de la loss function, el motiu és que, com hem vist abans a la fòrmula, la loss function esta formada pel MSE sumant el paràmetre de regularització que és la norma dels pesos multiplicada per λ, i és lògic pensar que com més gran sigui aquesta λ, més creixerà també el valor de la funció. Per tant, que el valor de la funció objectiu sigui major entre les diferents λ, no implica que tinguem més error de predicció, ja que el mínim de la funció serà el mateix i no es veurà afectat.

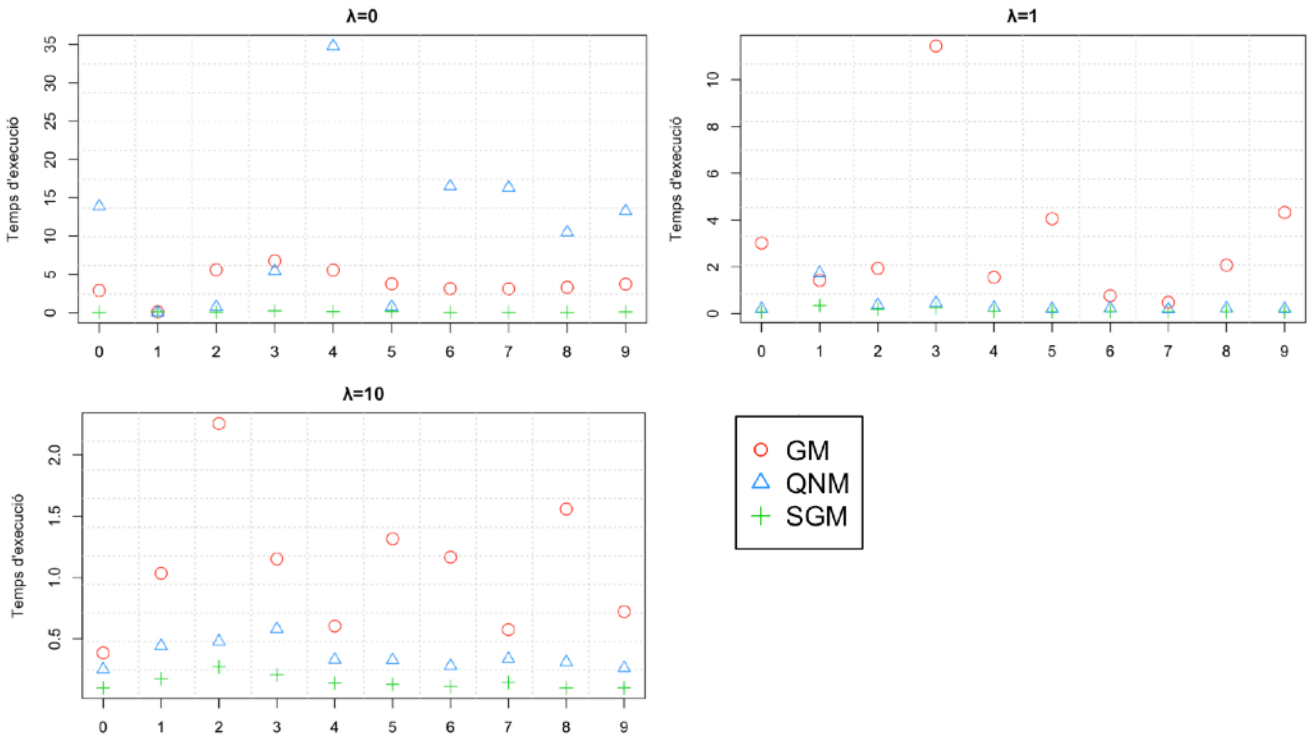
Quan $\lambda = 0$, observem que hi ha casos (números 3 i 8) on el mètode Quasi Newton convergeix, però el valor de la funció objectiu en aquest punt és major al valor de la funció objectiu del mètode del Gradient, tot i que aquest no hagi convergit. Això ho atribuïm a que l'algorisme de Quasi Newton ha convergit a un mínim local i no al global, perquè si no, no existiría cap valor inferior de la funció.

Com era d'esperar, quan la λ creix, la funció de pèrdua pren el mateix valor pel mètode del Gradient i Quasi Newton, ja que regularitzant la funció, disminuïm la probabilitat de que algun d'aquests algorismes convergeixi a un mínim local, i per tant, els dos mètodes convergeixen al mateix mínim, el global. És per això que en la nostra execució quan λ no és 0 observem que obtenim exactament els mateixos resultats en ambdós mètodes (tot i que no sempre succeirà per una λ tan petita).

Quan $\lambda = 1$, al mètode del Gradient Estocàstic, la funció objectiu convergeix a valors inferiors. Això no és una contradicció amb que els altres mètodes hagin trobat el mínim global, perquè aquest mètode al fer servir un petit percentatge de les mostres per crear el predictor, la seva funció objectiu i el seu mínim seran diferents.

b) Local convergence

En el següent gràfic podem veure el temps d'execució en segons que ha trigat cada algorisme en executar-se.



Com era d'esperar el mètode del Gradient Estocàstic és el que menys triga en tots els casos (fins i tot quan arriba al màxim), ja que encara que faci moltes iteracions el cost d'aquestes és molt petit ja que fa servir una proporció molt petita d'observacions.

Tot i que cada iteració del mètode Quasi Newton requereix més cost computacional que el mètode del Gradient, i per tant més temps, donat que té convergència quadràtica fa moltes menys iteracions que el mètode del Gradient, que té convergència lineal, i és per això que el resultat de Quasi Newton és més ràpid. Quan els dos algorisme convergeixen sempre tindrem que el mètode Quasi Newton triga menys que el mètode del Gradient.

Quan augmenta λ, el mètode del Gradient trigua menys temps i iteracions en convergir, ja que estem apropant la funció objectiu a una quadràtica (la norma dels pesos al quadrat) i sabem que el mètode és més efectiu com més convexa sigui la funció i més rodones siguin les corbes de nivell d'aquesta.

En canvi, amb el mètode de Quasi Newton, sí que observemuna reducció important del temps de $\lambda = 0$ a $\lambda = 1$, però observant detalladament els temps de $\lambda = 1$ i $\lambda = 10$ veiem que no hi ha cap millora, excepte amb el dígit 1, que sembla haver alguna anomalia concreta:

QNM	0	1	2	3	4	5	6	7	8	9
$\lambda = 1$	0.1991	1.7346	0.3681	0.4578	0.2433	0.1985	0.2176	0.1902	0.2178	0.2003
$\lambda = 10$	0.2513	0.4431	0.4785	0.5811	0.3294	0.3270	0.2805	0.3350	0.3107	0.2613

Quan $\lambda = 0$ el comportament dels algorismes és irregular, com hem comentat anteriorment, els mètodes poden convergir a mínims locals o no convergir en absolut. Així doncs, quan els dos fan el màxim nombre d'iteracions el mètode del gradient és més ràpid perquè fa càlculs més senzills.

c) General performance

Sobre el mètode del Gradient, podem dir que com ja sabiem és un algorisme amb convergència lineal i per tant, és lent. Però com hem comentat abans, a mesura que augmentem el valor de λ, triga menys iteracions en convergir ja que la funció s'aproxima a una quadràtica. El valor de la loss function sempre és menor o igual al de Quasi Newton.

Sobre el mètode del Gradient Estocàstic, el que més descata és la seva velocitat, tot i que no fa menys iteracions que els altres mètodes ja que també té convergència lineal. El valor de la seva loss function és inferior al dels altres mètodes, però no poder atribuïr això a un millor reconeixement, ja que ha fet servir una proporció d'observacions menor per crear aquesta loss function.

Per últim, sobre el mètode Quasi Newton, el nombre d'iteracions és el més baix ja que aquest algorisme té convergència quadràtica. Quan λ pren valors diferents a 0, el punt al que convergeix sempre és el mateix que el mètode del gradient, en canvi quan $\lambda = 0$, veiem que molts cops la convergència és a un mínim local i la loss function és major.

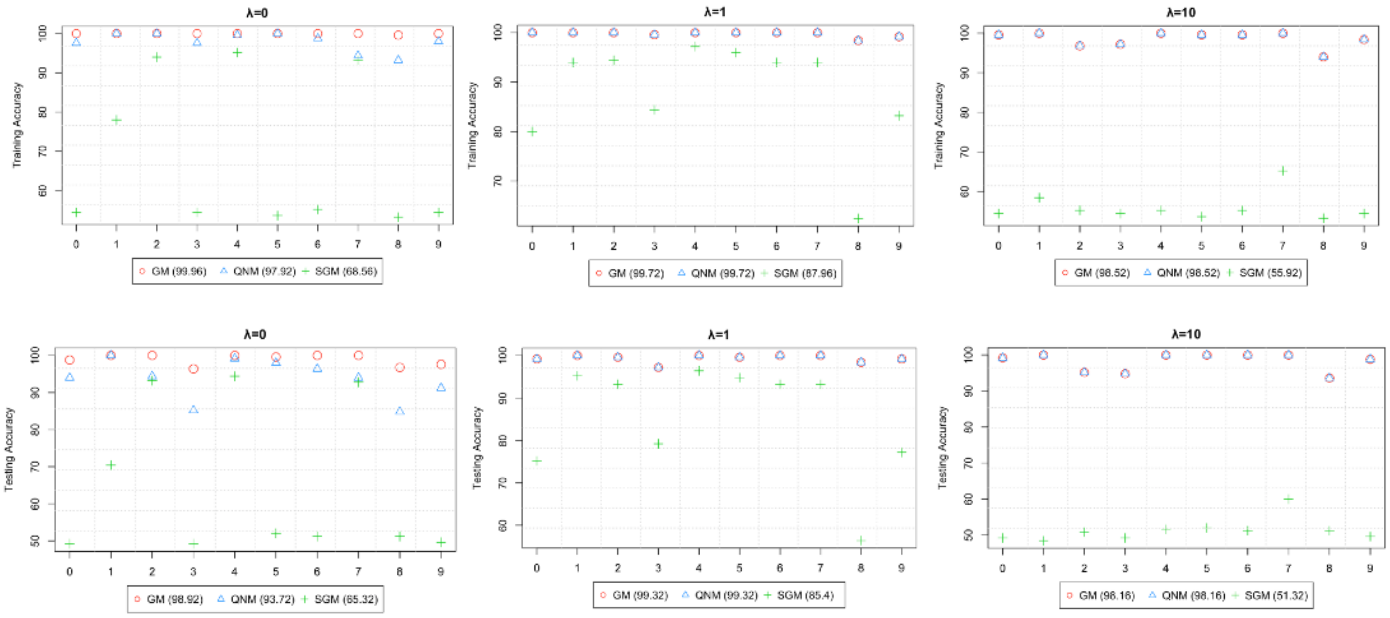
Basant-nos en la convergència global i local dels algorismes, podem concloure que les combinacions més apropiades λ-algorisme són les següents:

- $\lambda = 1$ i mètode del Gradient Conjugat, ja que tot i que fa un nombre d'iteracions bastant elevat, quasi sembre convergeix i el temps que triga és molt baix. També pot ser un bon mètode aparentment, ja que els valors de les funcions objectiu són molt baixos cosa que ens indica que la predicció serà bona (per la proporció de les mostres que pren per fer el training).
- $\lambda = 10$ i mètode Quasi Newton, perquè en molt poques iteracions i molt poc temps arribem al mateix mínim golbal que el mètode del Gradient

2. Recognition accuracy

a) Recommended λ-algorithm combination

En aquesta gràfica comparem la precisió de reconeixement dels números segons les diferents combinacions λ-algorisme:



A cada llegenda hi ha el percentatje mig d'encert de cada mètode amb cada λ, tant pel training com pel testing.

Veiem que, el mètode del Gradient Estocàstic ens dona una precisió de reconeixement inferior al altres dos mètodes, tant en el training com en el testing. Obtenim els millors resultats amb aquest mètode quan $\lambda = 1$, un 85% d'encert al test, en canvi els resultats són dolents quan $\lambda = 0$, i molt pitjors quan $\lambda = 10$, ja que tenim un 65% i 51% respectivament, això és causat perquè la mostra que agafa de les observacions per crear la loss function possiblement no sigui suficientment representativa.

Amb els mètodes del Gradient i Quasi Newton, no canvia la precisió de reconeixement en funció de λ, només hi ha petites variacions que nosaltres atribuïm a l'atzar. Les precisions d'aquests dos mètodes sempre són iguals o molt semblants entre elles i mai inferiors al 95%.

Com hem vist abans, en $\lambda = 0$ als casos en què el mètode de Quasi Newton convergia a un mínim local on el valor de la loss function era superior, hem obtingut més error de predicció tant al training com al testing. Quan no arriba a convergir la seva loss function és superior, i per tant, també hi ha més error.

En resum, amb el mètode del Gradient sempre obtenim els millors resultats, però és un mètode molt lent. El mètode del Gradient Estocàstic l'algorisme triga molt poc temps, però perdem molta precisió de reconeixement. I per últim, el mètode Quasi Newton no arriba a ser tant ràpid com el Gradient Estocàstic, però és bastant ràpid, i la precisió d'encert és lleugerament inferior a la obtinguda amb el mètode del Gradient quan $\lambda = 0$, i exactament igual quan λ és 1 o 10.

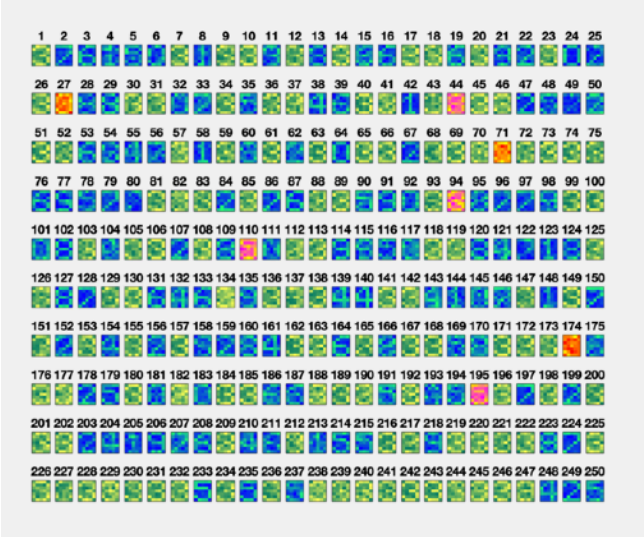
Per tant, considerem que la millor combinació que hauríem d'utilitzar és el mètode de Quasi Newton amb $\lambda = 1$, ja que obtindrem una precisió de reconeixement del 99% en un temps relativament reduït: al voltant de 0,3 segons per cada dígit.

b) Is there any digit specially difficult to identify?

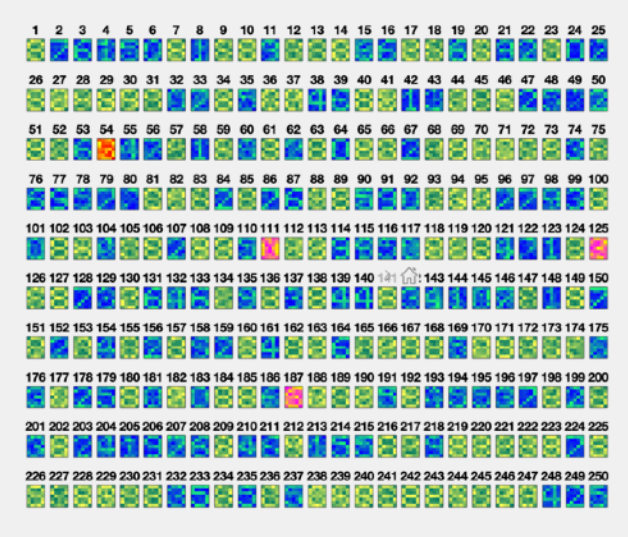
Els dígits que semblen ser més difícils d'identificar per a tots els mètodes, i per tant, hem obtingut menys precisió al test són el 3 i el 8. Cosa que és bastant raonable ja que aquests dos números tenen una forma bastant semblant i poden ser difícils de distingir per la màquina. Tot i així, amb el mètode recomanat a l'apartat anterior no obtenim una precisió inferior al 95% en cap d'aquests dos dígits.

Hem près $\lambda = 1$ i l'algorisme del mètode de Quasi Newton i hem realitzat la funció uo_nn_Xyplot pels objectius 3 i 8 i hem obtingut el següents gràfics:

Xyplot amb el dígit: 3



Xyplot amb el dígit: 8



Els casos en verd, han sigut 3 o 8 (per cada gràfica respectivament) ben classificats, els blaus són dígits diferents a 3 o 8 també ben classificats. Els de color vermell són falsos negatius, és a dir, sí eren 3 o 8 però ha considerat que no; i els rosats són falsos positius: no eren 3 o 8, però ha considerat que sí.

En el cas dels dígits a les posicions 44 i 94, al segon plot veiem que és ben classificat com un 8, i al primer plot és un fals positiu, és a dir ha confós el 8 per un 3. En el cas del dígit 125 passa el mateix però al contrari: és un 3 ben classificat al gràfic de l'esquerra, però al gràfic de la dreta ha sigut confós per un 8.

A part d'aquestes classificacions errònies observem més confusions amb altres números diferents, però entre aquests dos dígits la proporció és major, per tant, reafirmem el que hem dit abans: al tenir formes semblants, els dos dígits tendeixen a confondre's.

Hem fet servir els nostres DNIs com a seeds: tr_seed = 49784363 i te_seed = 54409254