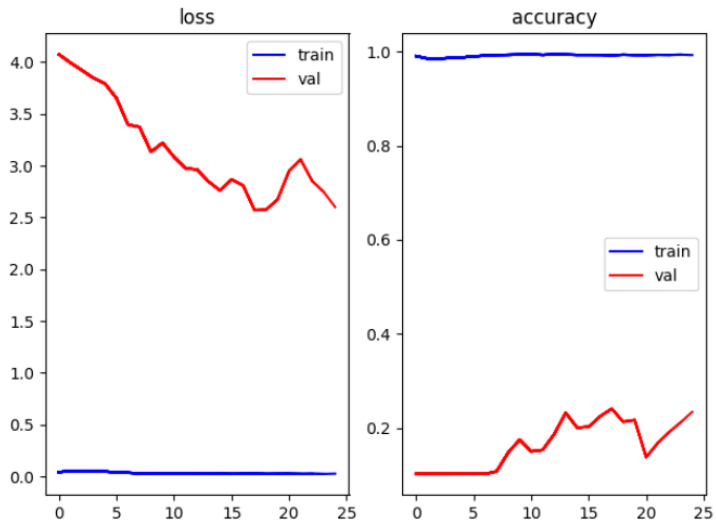
Progressió Pràctica 2 APC

Resultats amb el model donat

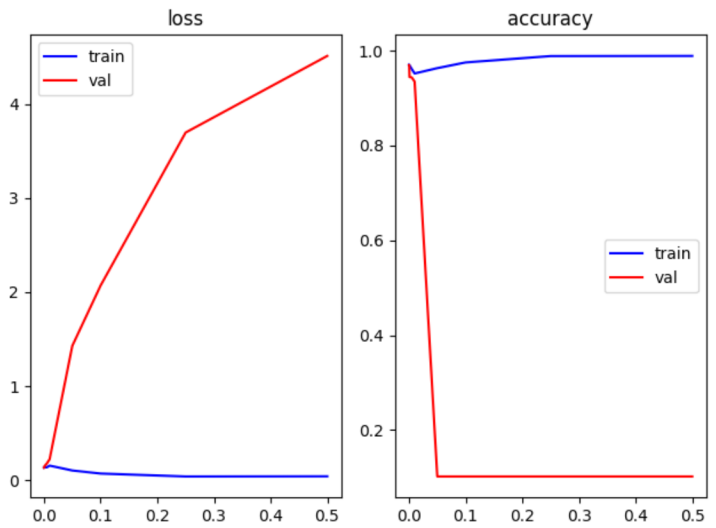
Inicialment tenim un model amb un accuracy molt elevat en el train i molt baix en la validació. Tot i això millora una mica (molt poc a mesura que avancen els epochs)



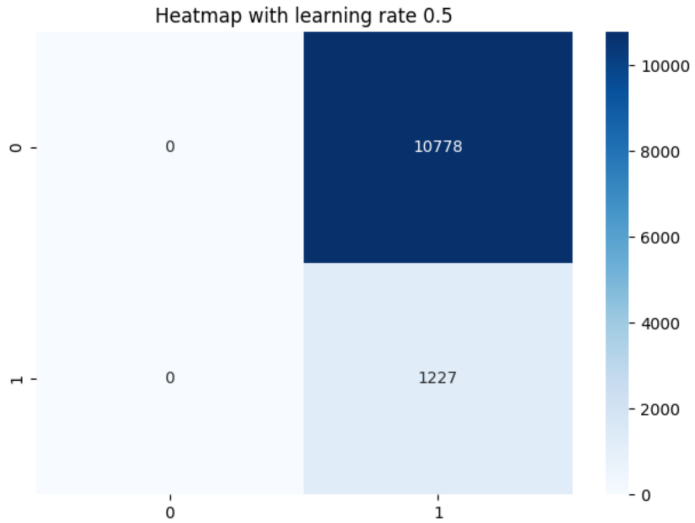
Donat el nostre criteri, considerem que el learning rate té un paper molt important en el rendiment del model. Tot i que la resta de paràmetres són també molt influents, decidiem començar l'estudi dels paràmetres per aquest en concret.

Estudi Learning Rate

S'entrena i es valida el model amb les mateixes dades però canviant el valor del learning_rate. S'obtenen els següents resultats:

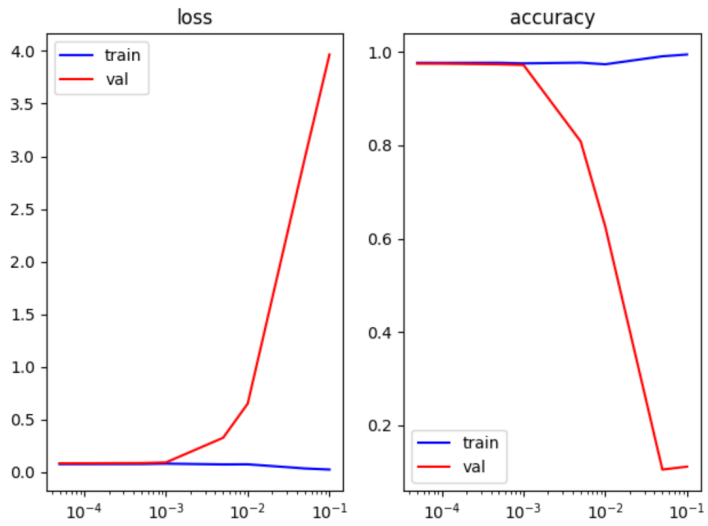


Donat a que els valors a provar eren: *[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5]*, no es pot apreciar be les diferencies en els primers valors donada l'escala de l'eix. Tot i això veiem com abans de 0.1 s'equilibra força però amb uns resultats molt baixos. A més hem creat una matriu de confusió per cada learning rate.



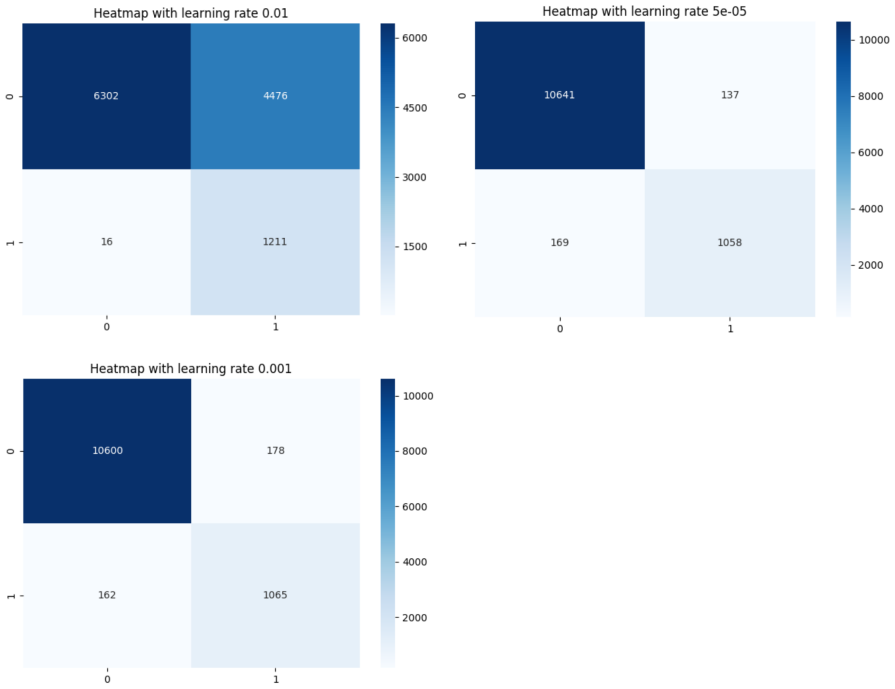
Podem veure com de dolent és el resultat amb valors elevats de *lr*.

Redefinim una nova llista eliminant el rang recent estudiat i obtenim els següents resultats.



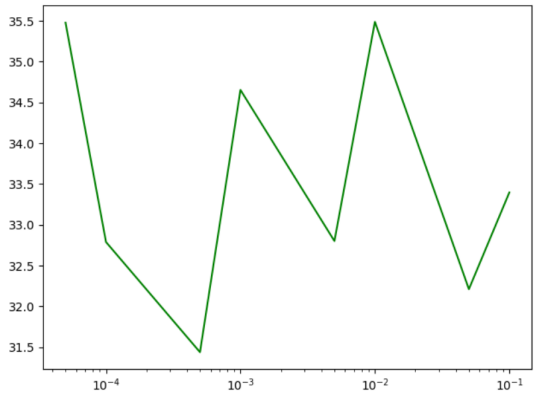
Com podem observar els resultats són molt millors per a valors petits de learning rate (menors de 10^{-3}). Observem el heatmap de 3 casos:

- El pitjor de l'estudi
- El millor
- Un intermig amb valors molt semblants al millor



És evident que el primer genera molt d'error al classificar si la mostra és de la classe del target o no (classifica un ≈ 40% de les vegades com que si que ho és, quan realment no)

Per si de cas, vam voler comprovar el temps d'execució necessitat per cada learning rate, en cas de que per a lr molt petits es necessités més temps, podriem considerar el 3r cas dels que acabem de mostrar com a solució òptima.



Tot i semblar que és força aleatori, realment la diferència no és exageradament gran (varien en un rang de 5s) però en cas de que realment tingui un motiu, llavors prioritzarem als valors intermitjos de cada ordre (5×10^{-k}).

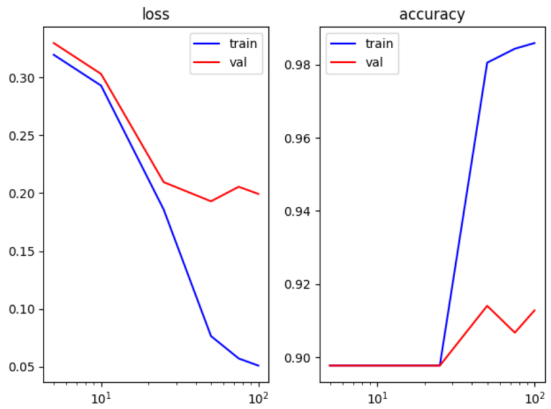
Anotació

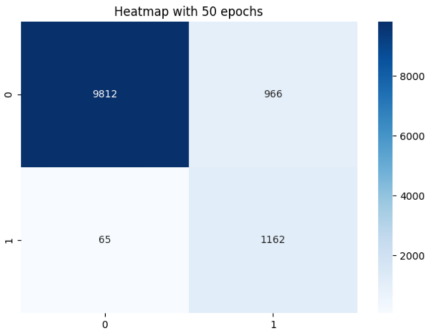
Donades les matrius de confusió podem observar com el model no tendeix al que teniem pensat: classificar-ho tot com a no target donat a que hi ha moltes més mostres negatives que no pas positives -> accuracy fàcilment adaptable a això.

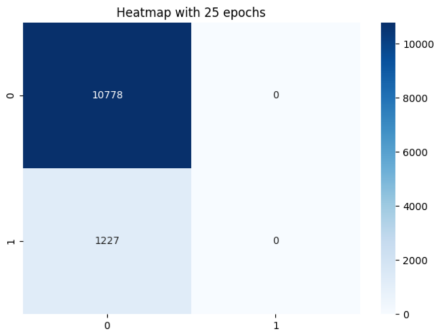
Tot i això, seria interessant estudiar el comportament en funció del recall, doncs cara l'apartat B ens interessa que tingui una tasa d'encert el màxim d'alta per a les mostres que són el target realment.

Estudi Epochs

Per a fer aquest estudi utilitzem un lr=0.0005 seguint els resultats del cas anterior. Generem un model per cada valor d'epochs i avaluem el seu resultat. Obtenim el següent:





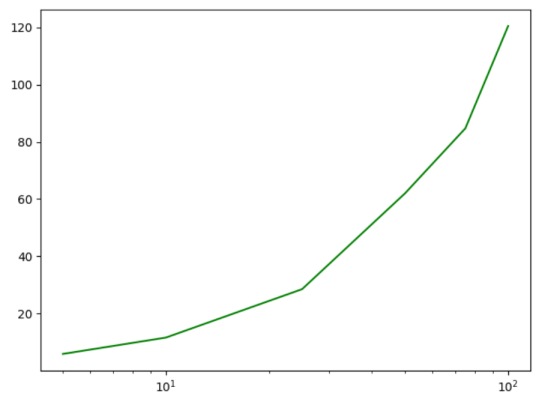


> **Dubte:** Com es possible que donat els mateixos valors pels paràmetres que en l'anterior test ens donin resultats tant diferents? (epochs == 25).

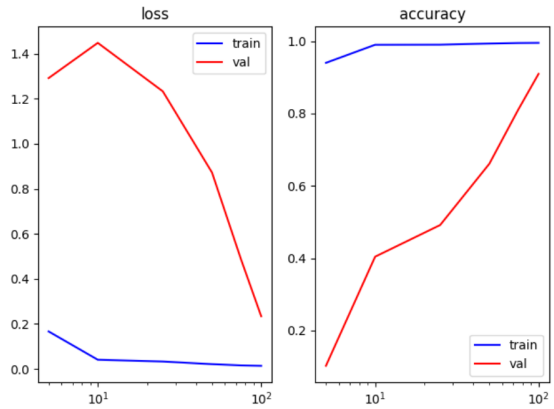
El millor resultat és per epochs = 50 i veiem com per a 25 el model es conforma en classificar-ho tot com a no target (ja ho hem comentat abans els motius i tal).

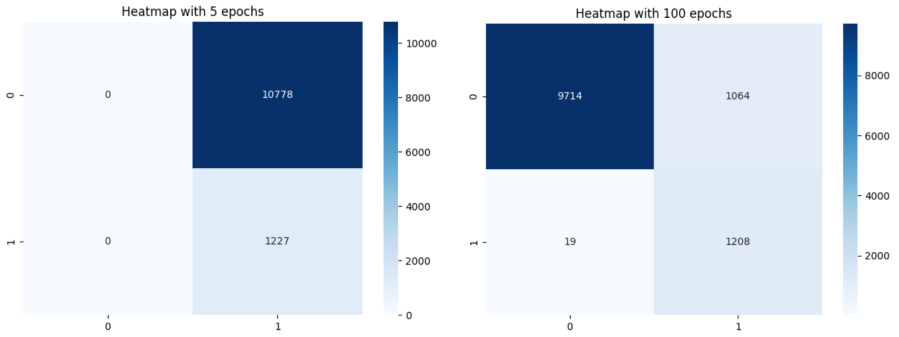
A partir de 50 epochs sembla que hi ha overfitting tot i que realment tenim bons resultats tant a val com train.

El temps d'execució és evident que augmenta com a més epochs, doncs es fan més cicles.



FEM UN SEGON TEST AMB LEARNING RATE = 0.005

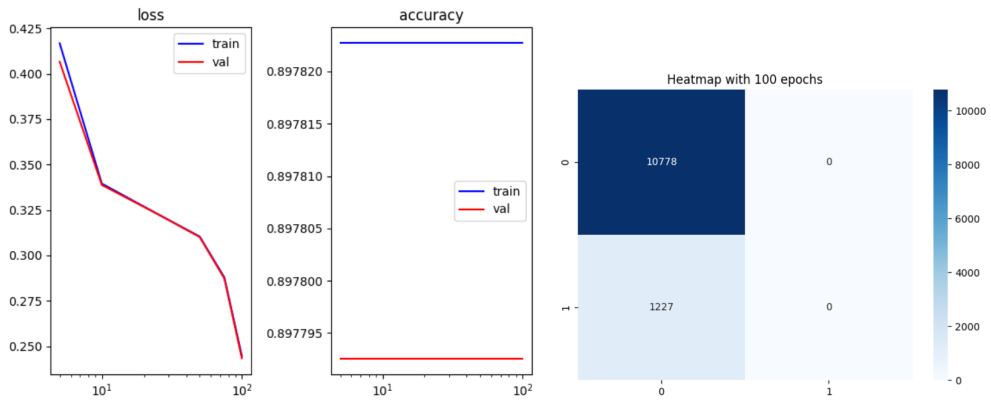




Inicialment és un model amb molt d'error a diferència del previ (començava directament classificant tot com a no target). A mesura que avança millora en la bona direcció. Tot i això amb 100 epochs no té suficient doncs encara classifica amb un 50% d'error la classe target (que és el que més ens interessa).

FEM UN TERCER TEST AMB LEARNING RATE = 0.00005

Terrible. Ens trobem amb contradiccions respecte l'estudi anterior:

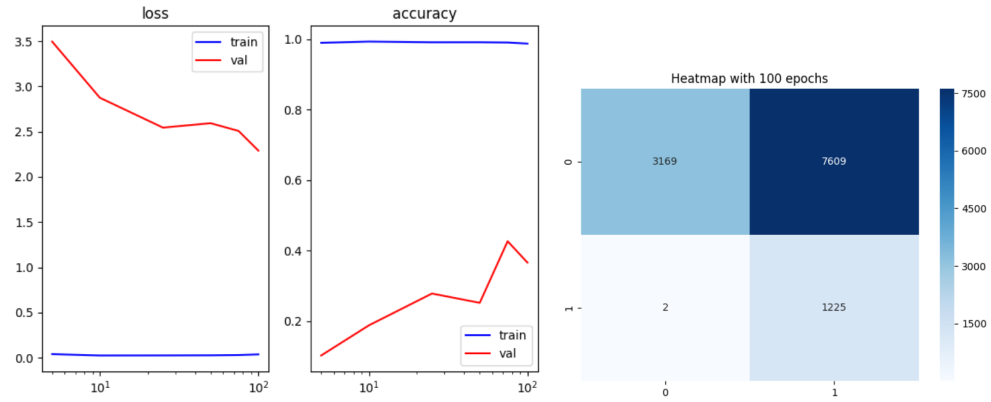


El heatmap és igual per a tots els valor de epoch. Té lògica que sigui d'aquesta manera. Tot i això és extrany que el comportament de lr normal estigui en el cas més alt mentre que té un rendiment molt alt amb el valor de lr intermig.

Caldria confirmar la validesa de l'estudi anterior. En cas que fos valid, analitzar el perquè d'aquesta diferència.

ULTIM TEST PER SORTIR DE DUBTES AMB LR = 0.05

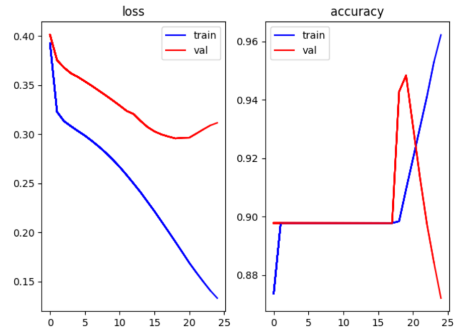
Genera més confusió doncs ens reafirma que el lr òptim és d'ordre menor. Aquests són els resultats:



Tenim un overfitting de la hostia. Potser s'ha de filar encara més prim amb el learning rate.

Decisió valor epochs i lr

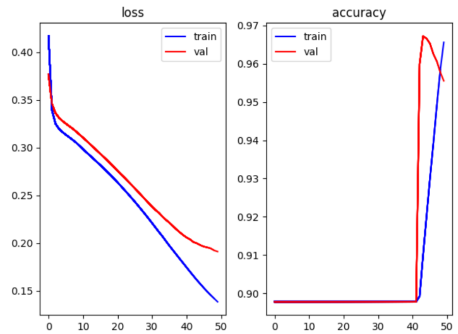
Per a seguir provant paràmetres es decideix establir un valor per epochs i lr. Amb lr = 0.001 i epochs = 25 obtenim aquest resultat:



Clarament el model arriba al overfitting després de 18-20 epochs, així que retallarem aquest valor.

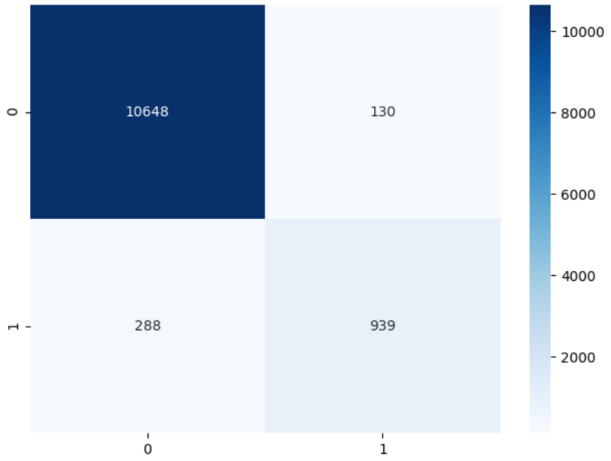
Tot i això ens adonem que el model en aquest cas termina amb un accuracy del 90%, però perque ho categoritza tot com a no target (problema de l'accuracy ja comentat).

Redefinim valors: lr = 0.0005 i epochs = 50.



Veiem com a partir de 41-42 epochs comença a tenir overfitting, retallem de nou. Realment necessitem 45 epochs.

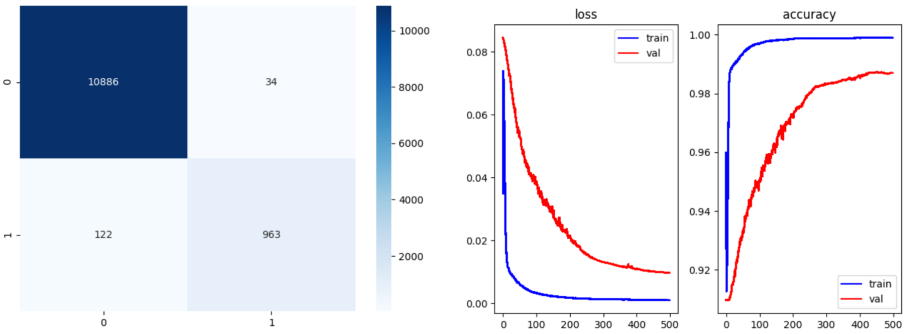
Obtenim aquests resultats:



Ja no tenim el problema del accuracy tot i que té valors força alts de confusió.

Update

Vale, es una merda perque no funcionava amb tots els casos. Toca augmentar el lr a 0.01 i pujar considerablement el valor dels epochs per a tenir un model casi perfecte. Provem amb 500:



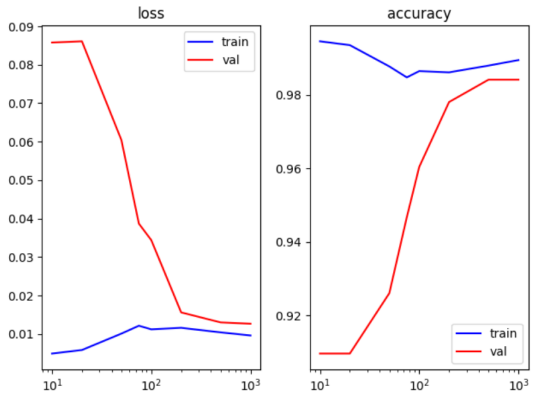
Realment s'estabilitza força als 250-300 però consumeix massa temps. Hem de trobar la manera d'optimitzar això.

Batch size

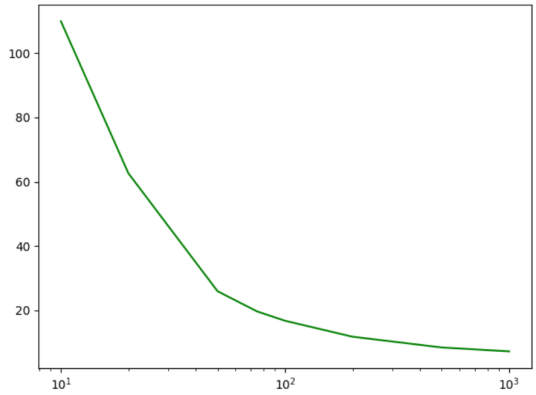
Aqui es comença a veure la llum. Trets importants:

- Contra més gran, entrena més rapid i equival a tenir un learning rate també més gran.
- Al cas contrari, més lent però teòricament més precís.

Per a diferents valors tenim els següents resultats:



Podem observar com per a majors valors del batch_size millors resultats. Realment no és ben bé així. El que està passant és que estem accelerant el procés. Teòricament hauriem d'arribar a resultats similars si modifiquem el batch_size sempre i quan li permetem realitzar els epochs corresponents. Ha estat entrenat amb 25 epochs, si ho fessim amb 200 per als valors més petits, hauriem de tenir un resultat similar a les mides més grans amb menys epochs d'entrenament.



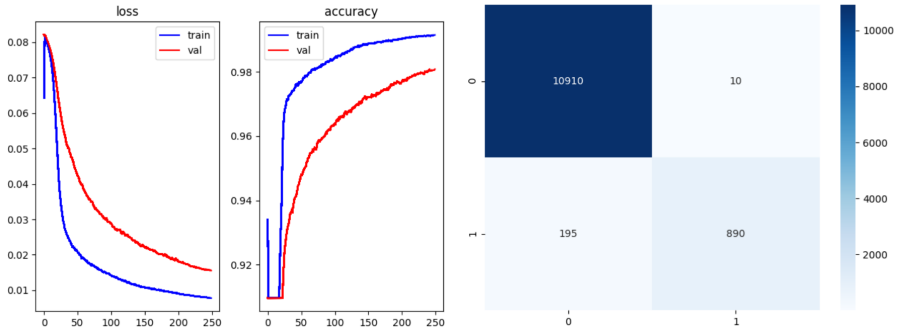
A més podem veure com a mesura que augmentem el tamany, disminueix el temps d'execució. Això ens dona molt d'avantatge, amb valors grans:

- Arribem a bons resultats abans
- Podem augmentar els epochs d'entrenament per a millorar els resultats.

Tot i això no convé posar el valor més alt possible. Considerem que un intermig entre 500 i 1000 és un bon valor.

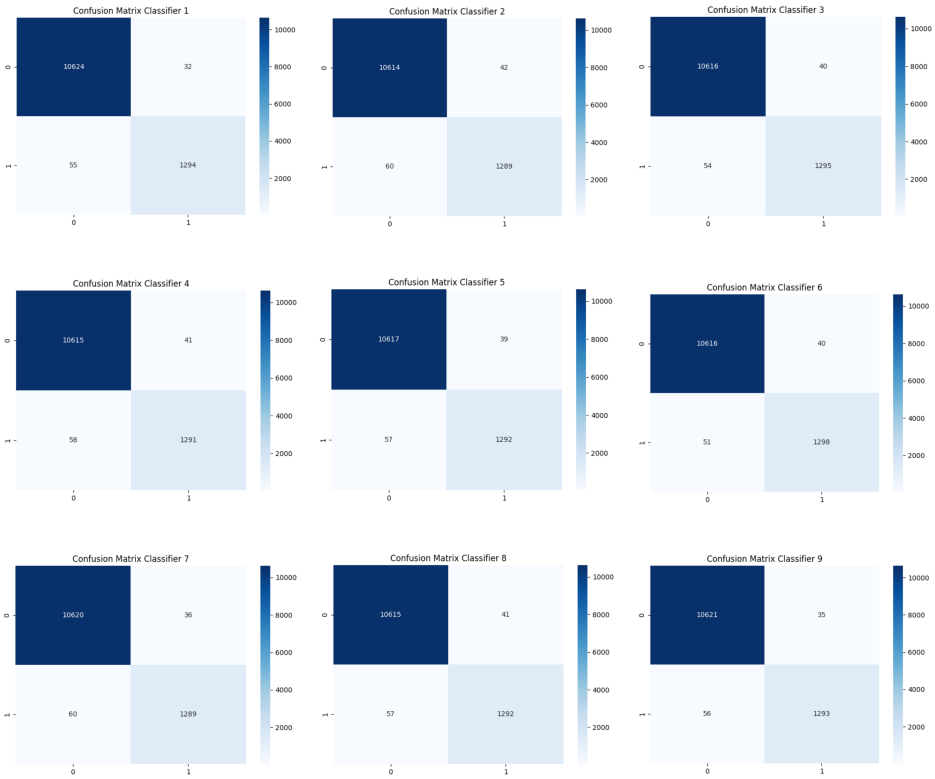
Paràmetres fins a aquest punt

- **Learning rate:** 0.05
- **Epochs:** 250
- **Batch Size:** 750



Creació de tots els classificadors binaris

Per a poder fer això possible cal modificar moderadament el codi aportat. D'aquesta manera en comptes de calcular sols per un model, se'n genera una llista amb tantes posicions com classes hi ha. El resultat d'entrenar cada un dels models és el següent:



Els resultats són molt similars entre models, per tant podem concloure que el model és igual de funcional per a tots els targets.