Projet Final de IFT6390

Partie de Reseau à convolution

# L’influence de hyperparametres sur MNIST pour le reseau de convolution. (Pour question : Que pouvez-vous offrir comme suggestion à quelqu’un qui ferait face à ce problème dans le monde "réel"? (Algorithme proposé, hyper-paramètre, cueillette de données, etc.)

Pour les donnees de MNIST, on va constater L’influence sur les hyper-parametres traditionnels (Taux d’apprentissage, batchsize, regularisation) en fixant la taille de neurons couches cachees :

Reseau à convolution=

ConvolutionLayer[20, {5, 5}], // 20 filtres, 5 par 5

ElementwiseLayer[Ramp], // activation ReLU

PoolingLayer[{2, 2}, {2, 2}], // Pooling filtres, 2 par 2

ConvolutionLayer[50, {5, 5}], // deuxieme couche a convolution, 50 filtres, 5 par 5

ElementwiseLayer[Ramp], // activation ReLU

PoolingLayer[{2, 2}, {2, 2}], // deuxieme couche pooling, filtres 2 par 2

FlattenLayer[],

LinearLayer[400], // Multi-layer Perceptron, 400 neurones

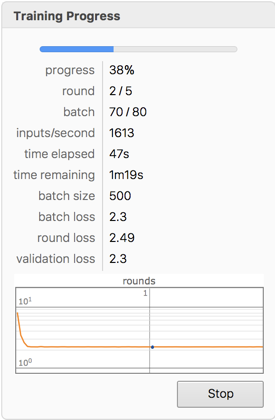
ElementwiseLayer[Ramp], // activation ReLU

LinearLayer[10], // couche sortie

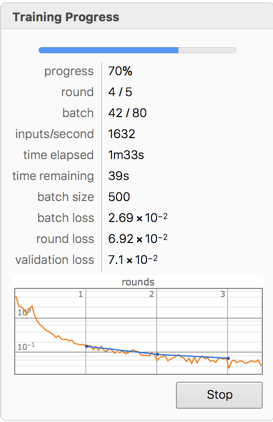
SoftmaxLayer[]} // softmax, sortie pour la classe probable.

## Concentrer Taux d’apprentissage, (Epouque=5, batchsize=500, regularisation=0 )

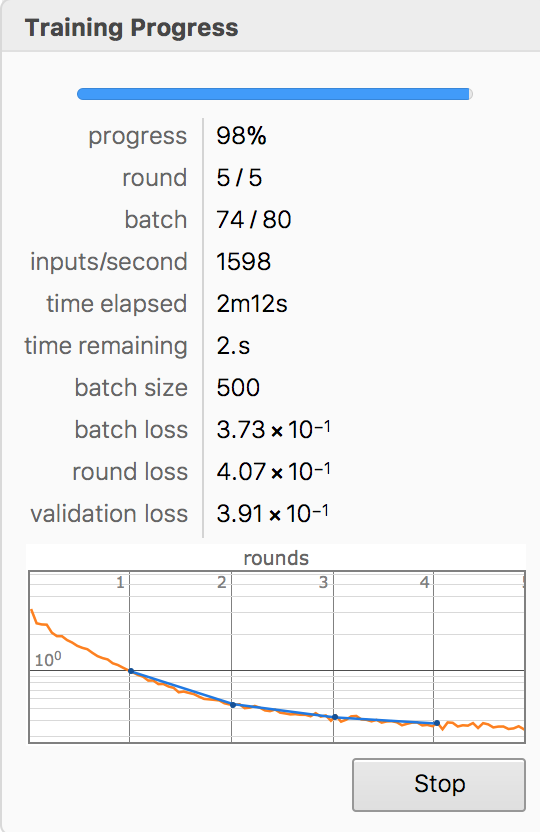
### Taux d’apprentissage=5, trop grand, ca marche pas



### Taux d’apprentissage=4, ca marche, mais pas stable, pour entrainer 3 fois, taux d’erreur={0.0142, 0.0438, 0.0133}.

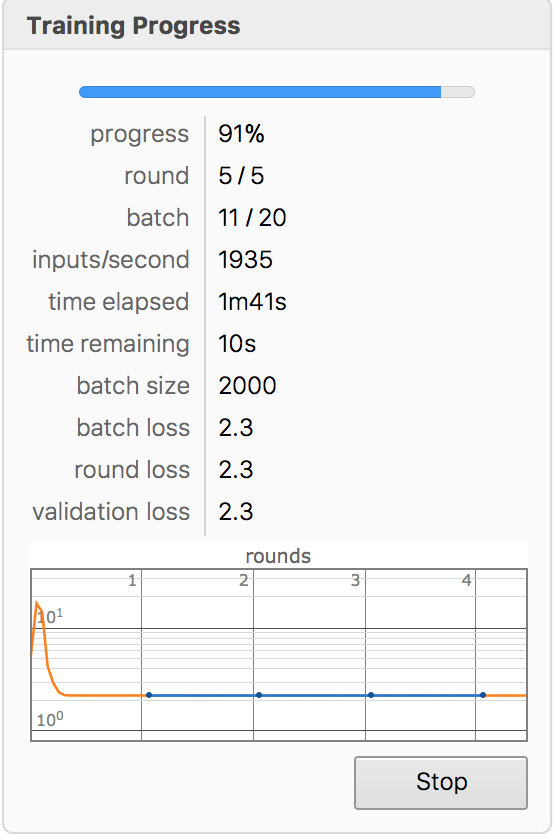


### Si taux d’apprentissage est trop petit(0.05), la gradient va desendre lentement. pour entrainer 3 fois, taux d’erreur={0.0901,0.0942,0.1069}

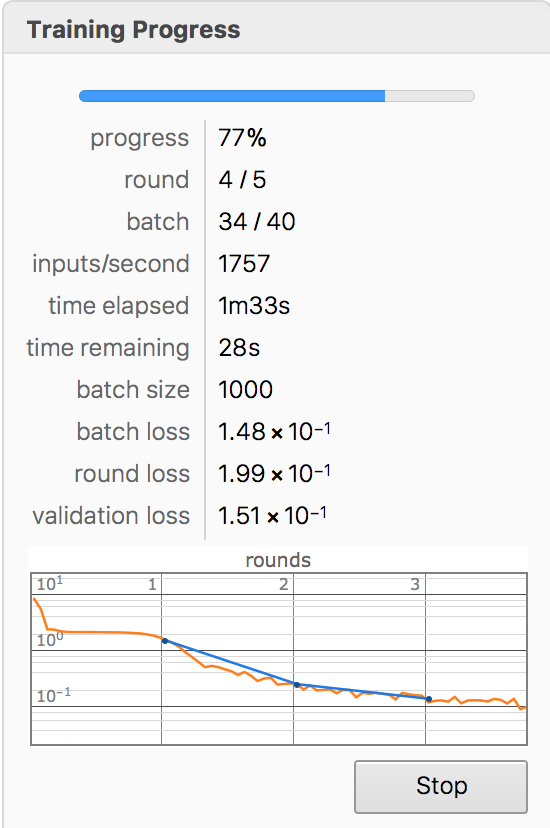


## Concentrer batchsize, (Epouque=5,taux d’apprentissage=3 , regularisation=0)

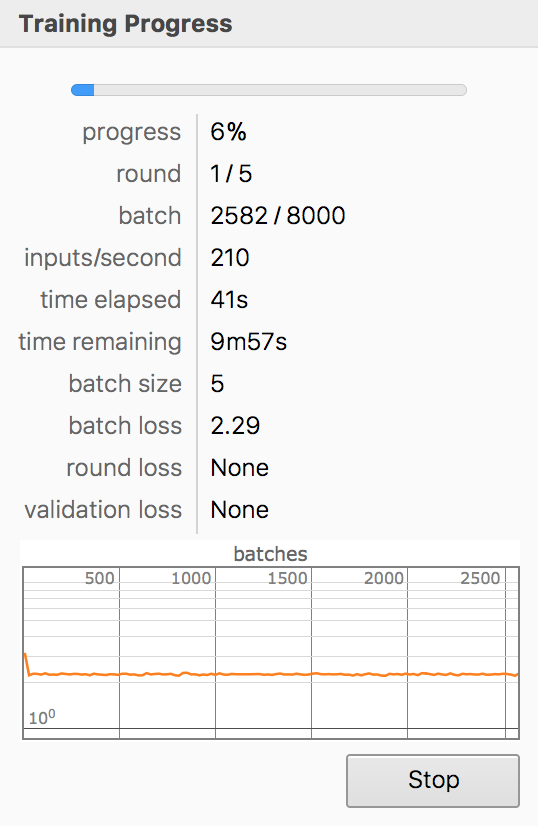
### batchsize=2000, ca marche pas



### batchsize=1000, ca marche , mais pas stable, pour entrainer 3 fois, taux d’erreur= {0.0297, 0.0657, 0.0205}

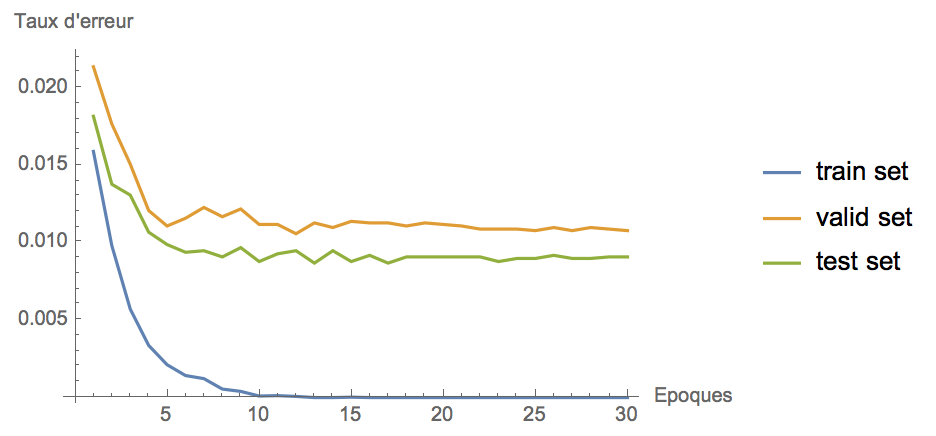


### batchsize=5, ca marche pas, calculer lentement et la gradienne ne descent pas.

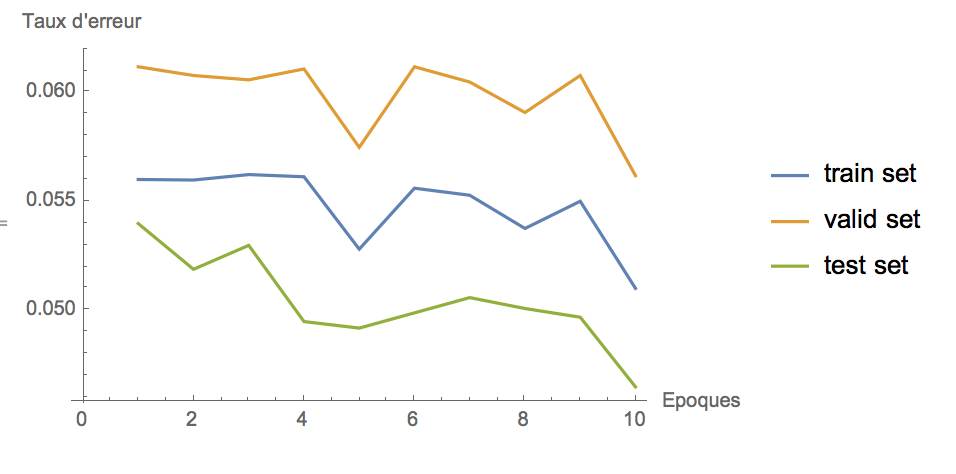


## Concentrer regularisation, (Epouque=30,taux d’apprentissage=1.8 , batchsize =100)

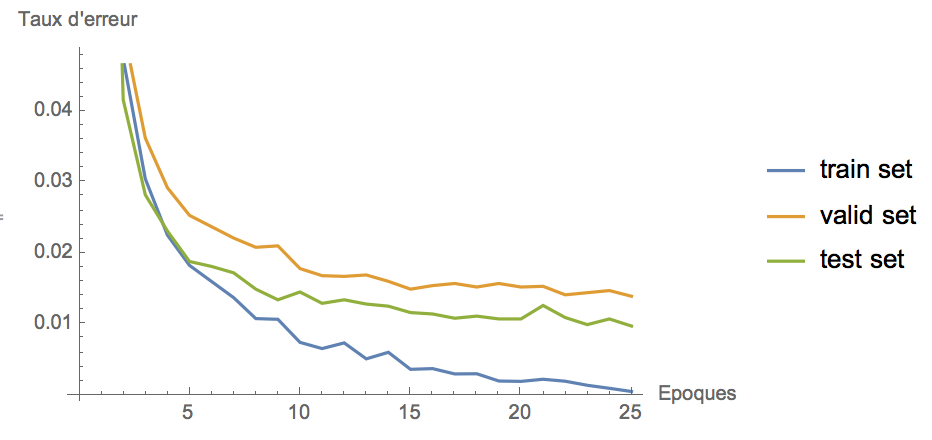
### regularisation =0. 000000000001, On suppose que le modele deviendrais sur-apprentissage. Le fait, c’est la variance d’erreur entre ensemble de train et ensemble de test augumente au debut (epoque<8). Mais enfin le taux d’erreur dentrainement=0 et le taux d’validation et le taux de test restent environ 0.010. Effectivement, La variance existe, mais grace à le nombre de l’ensemble de donnees assez grand et la competence du reseau a convolution, le taux d’erreur sur test est assez petit.



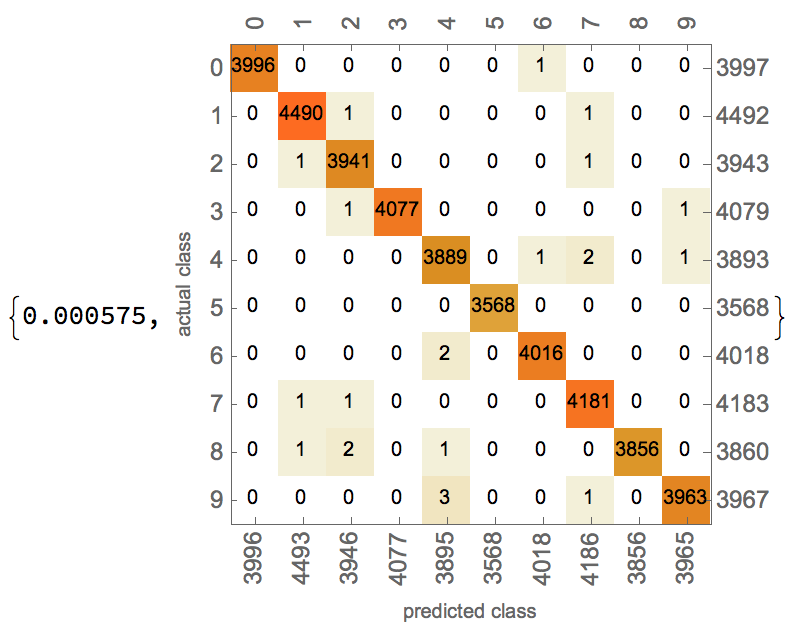
### regularisation =0.05, le modele devient sous-apprentissage, la variance n’est pas grand, mais le biaise est grand.



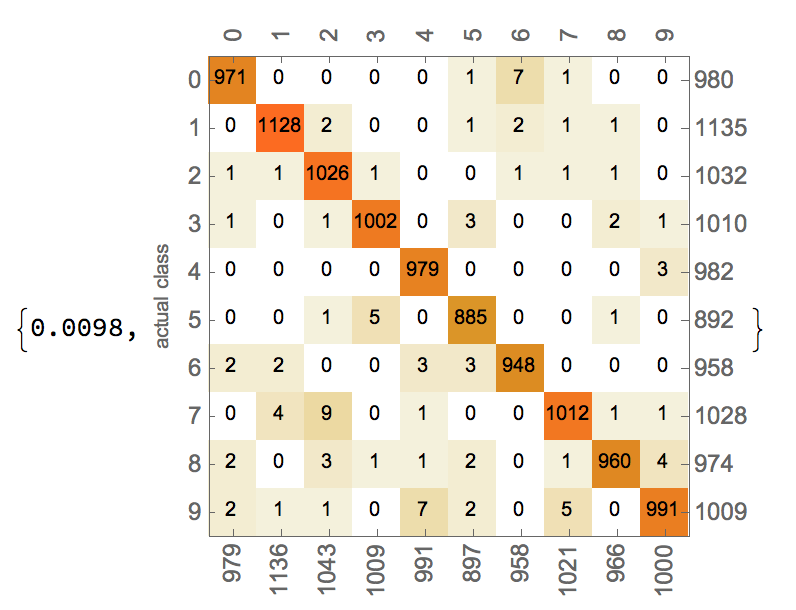
## Enfin, on choisit les hyper-parametres comme:Epouque=25,taux d’apprentissage=1.7 , batchsize =240, regularisation=0.000001 . le taux d’erreur sur test=0.0098.



Le taux d’erreur et la matrice de confusion sur l’ensemble de train :



Le taux d’erreur et la matrice de confusion sur l’ensemble de test:

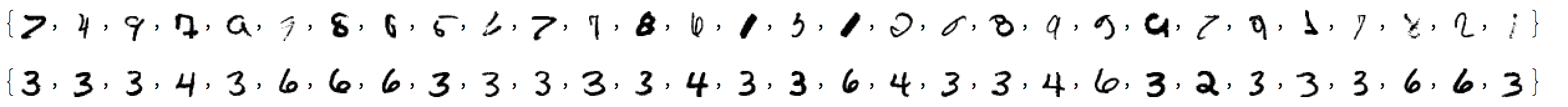


Conclusion :

1. On conseille d’essayer quelques valeurs sur certains de hyper-parametres en fixant les autres hyper-parameters pour obtenir des experience sur la question reel. Car pour marcher une fois, de plus il y a bcp de hyper-parametres, ca va durer bcp de temps, il faut choisir quelques combine de hyper-parametres selons l’experience, enfin faire une boucle sur les combines de hyper-parametres pour caleculer tout les resultats pour choisir les meilleurs hyper-parametres ;
2. si on a un ensemble de donnee assez grand, on peut entrainer une modele plus capacité sans avoir peur de sur-apprentissage.

## Question : Est - ce que les examples " difficiles " et " faciles " d' une base de données sont les mêmes pour tous les algorithmes choisis?

Apres avoir le reseau a convolution entrainé, On a choisit 30 images que le reseau predit les probabilités plus haut pour les autres classes en calculant la valeur d’entropie, c-t-d on a choisit les 30 images les plus grands entropies, au contraire on a choisit 30 images que le reseau predit le bon probabilité le plus haut. On constate que les images avec le plus haut entropie sont bien sûr le plus confus et ambigu, évidemment ce sont les écritures le pire, pas claire pas foncé et interrompu, donc le réseau les apprend les plus difficiles, et les écritures de la deuxième liste sont jolie, claires et foncés. On constate aussi que le chiffre 3 et 6 sont plus facile à apprendre, mais selon la matrice confusion precedent, le 7 est bcp au risque d'être identifié comme le 2, le 0 7 est bcp au risque d'être identifié comme le 6.



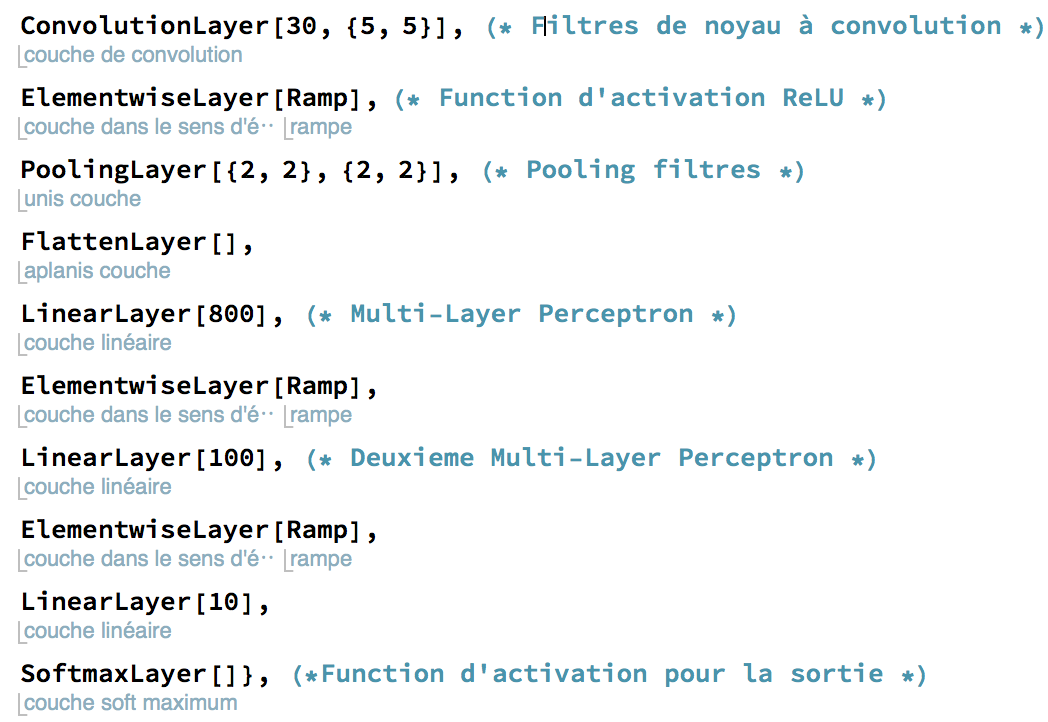
# L’influence de taille de filtre et taille de pooling (hyper-parametres) sur CIFAR – 10 pour le reseau de convolution.

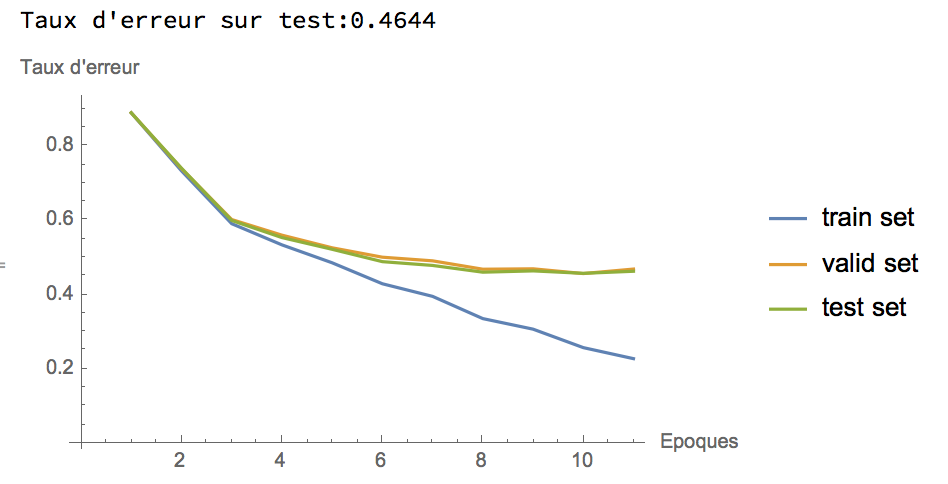
Pour les donnees de CIFAR-10, on va constater L’influence sur les hyper-parametres de couche neurone (taille de filtre et taille de pooling) en fixant les autres hyper-parametres.

Les autres hyper-parametres sont comme : epoque = 10~20, mini BatchSize = 200~300,

Regularization= 0.0000001, Taux d’apprentissage de descent gradient = 2.4, deux couche de Multi-layer Perceptron(800 neurones et 100 neurones) pour la sortie.

## Un couche de convolution et un couche de pooling. 30 filtre de convolution 5 par 5, pooling 2 par 2, pour boucler 10~15 epoque :



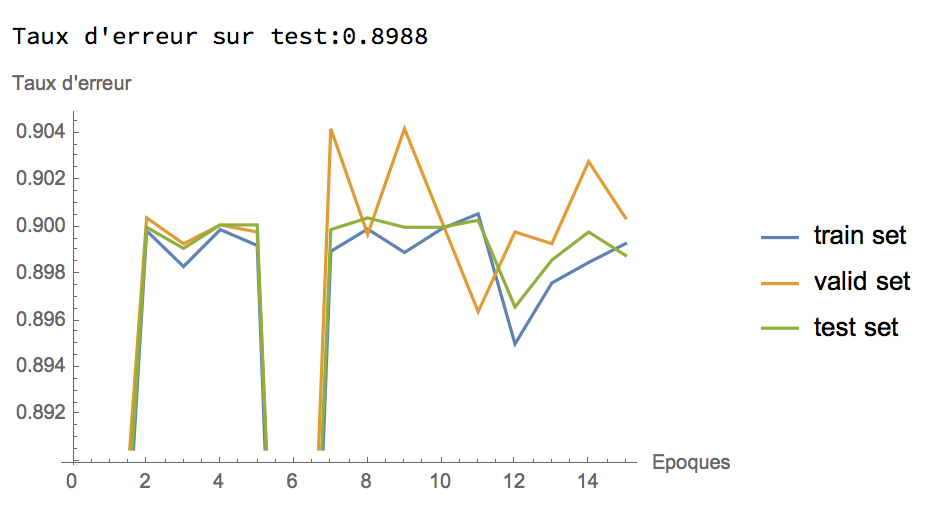


Selon le resultat, on a obtenu un modele sous-apprentissage. Un reseau de convolution avec seulement une couche de convolution n’a pas beaucoup de capacité.

## Deux couches de convolution et deux couches de pooling (2 par 2 )

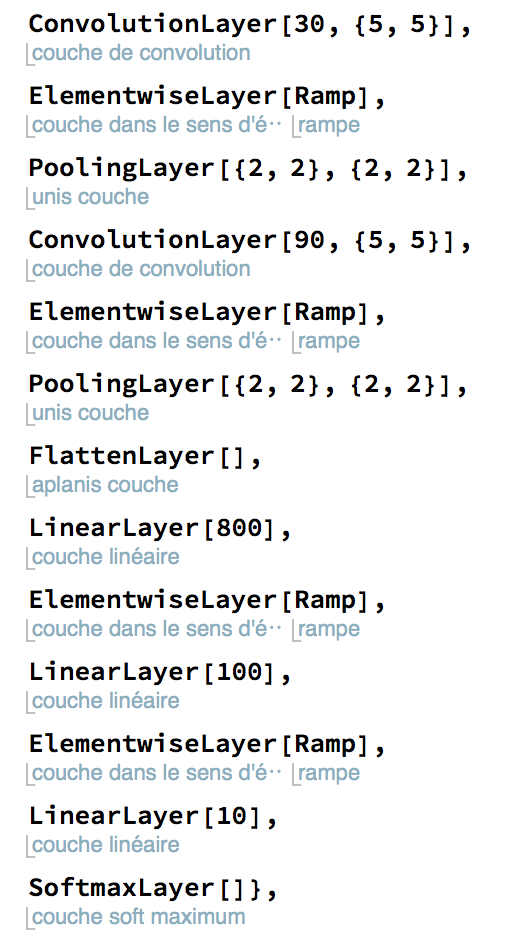
### La premiere couche de filtre convolutionelle est 30, 9 par 9 , La deuxieme couche de filtre convolutionelle est 90, 9 par 9. Le noyau est un peu plus grand pour les images original 32\*32.

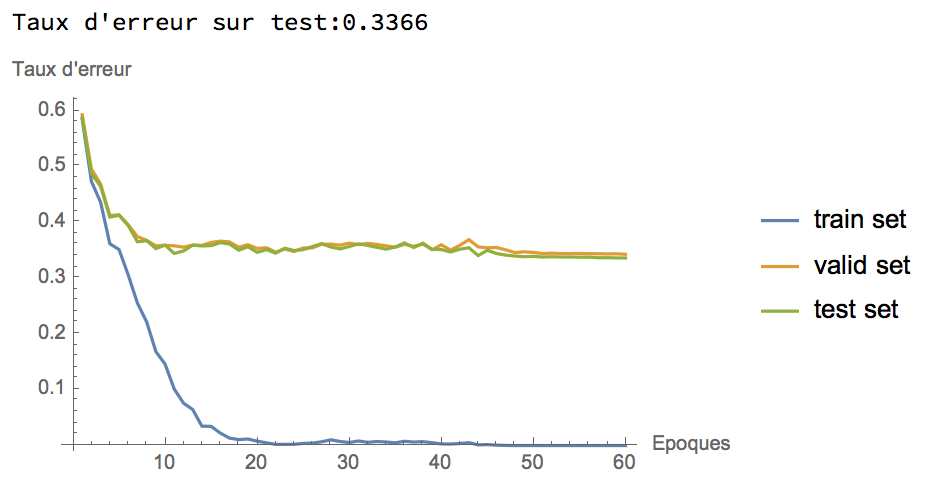




Selon le resultat, la taille de filtres convolutionelles est tres importante. Ca ne marche pas avec la taille 9 par 9 pour les donnees 32 par 32.

### On re-prend la taille de filtres convolutionelle à 5 par 5.



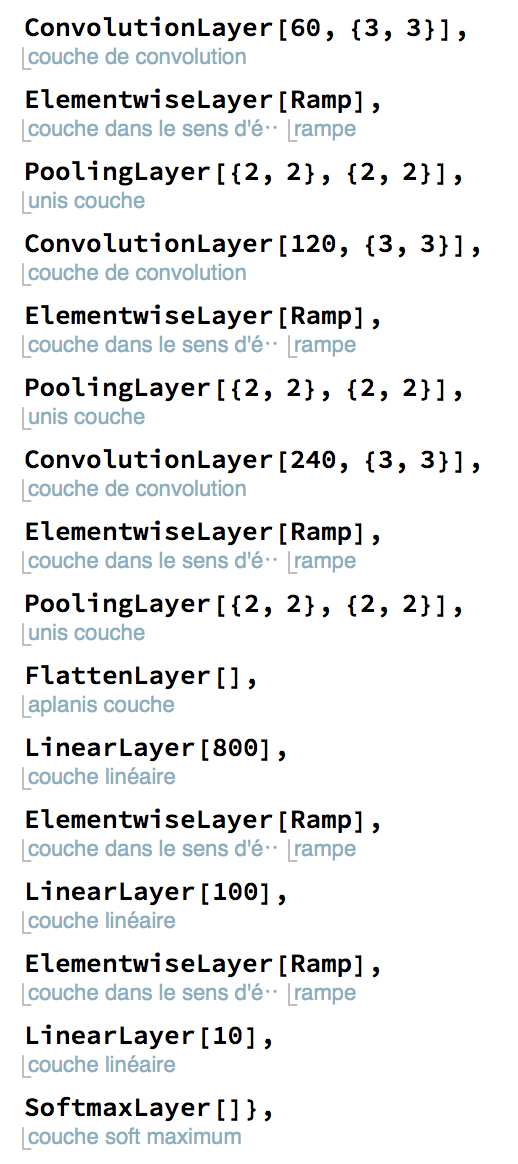


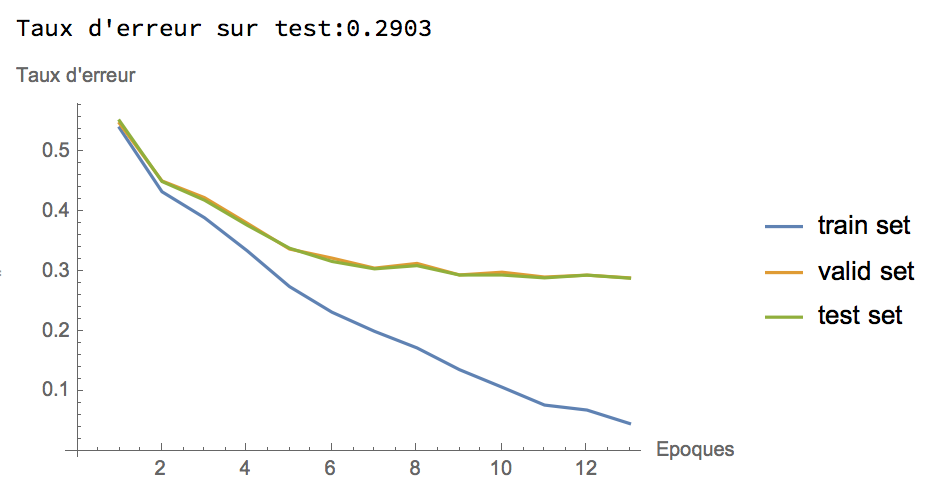
Selon le resultat, avec 2 couches convolutionelle, filtres 5 par 5, on a diminue evidemment le taux d’erreur sur test( 33%). Donc, en diminuant la taille de filtre convolutionelle et en augumentant le nombre de couche convolutionelle, la resultat est meilleur que les deux reseaux precedents.

## Trois couches de convolution et Trois couches de pooling:

## Enfin, on augumente le couche de convolution jusqu’à 3, on diminue encore la taille de filtre de convolution jusqu’à 3 par 3. On a fait 2 fois :

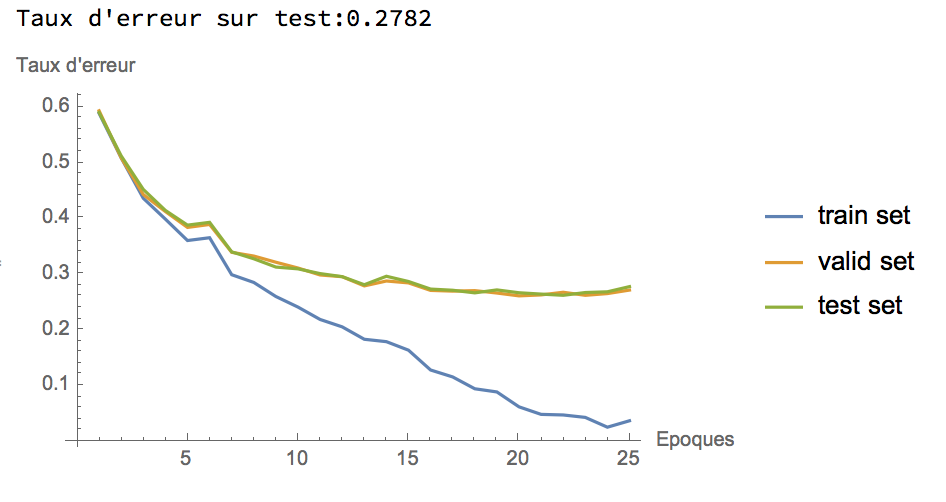
La premiere fois, pour 15 epoques, le taux d’erreur sur test diminue jusqu’à 29.08%





La deuxieme fois, on choisit la function max pour  la premiere couche de pooling, la function moyenne pous la deuxieme et troisieme couches, le taux d’erreur sur test diminue jusqu’à 27.82%

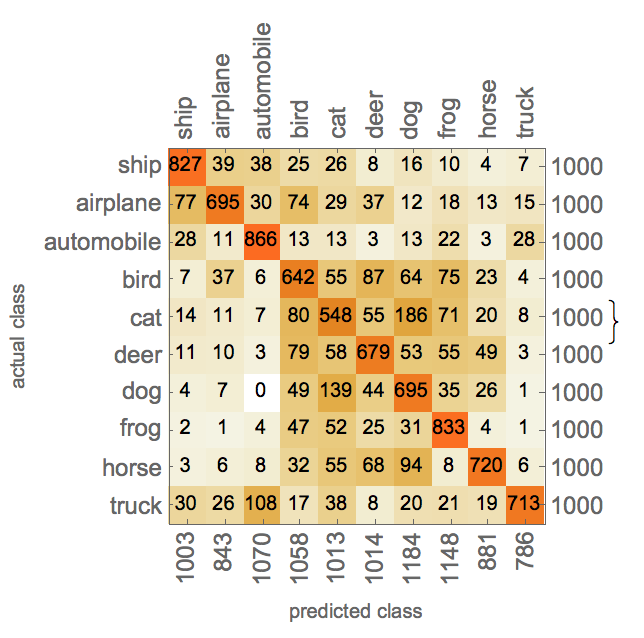




Conclusion :

1. Le nombre de couche convolutionelle et la taille de la filtre convolutionelle influence la capacité de la modele. Normalement la taille de la filtre 3 par 3 est meilleure que 5 par 5, et 3 couches convolutionelle est meilleur que 1 et 2 couches.
2. La combinaison de la function pooling (max, moyenne, moyenne) est meuiller que (max, max, max).

## Question : Est - ce que les examples " difficiles " et " faciles " d' une base de données sont les mêmes pour tous les algorithmes choisis?



Les 30 images avec la valeurs d’entropie le plus haut :



Les 30 images avec la valeurs d’entropie le plus bas:



Conclusion :

Selon la matrice de confusion, On constate que il y a 108 de truck qui sont mal classifiés à automobile, il y a environs 310 de dog et cat melangé. Il y a 77 de aireplanes qui sont mal classifiés à ship.

Selon les images avec le plus haut entropie, on voit ils ne sont pas très claire, il y a des bruits dans le décor ou la couleur du décor est similaire à l'objet, donc le réseau les identifie le plus difficile, mais les objets de la deuxième liste sont claires et la couleur du décor sont souvent au contraire de l'objet.

# Question : "Comparez la performance et les caractéristiques de chacun des algorithmes choisis sur les ensembles de données.Faites des recommandations afin de sélectionner des algorithmes, et notez vos observations sur vos ensembles de données.", "une brève description de chacun Comparez la performance (taux d'erreur, log-vraisemblance, etc.) des différents algorithmes sur chaque base de données. Est-ce qu'un des algorithmes performe systématiquement mieux que les autres? Pourquoi? "

Conclusion :

--Selon les resultats de 3 algorithmes, on constatue que la reseau convolutionnelle de neurones est beaucoup plus meuiller pour le domaine d'image.

**(Il faut ecrire plus )**