

PW-08 version 2

{dorian.magnin, noemien.kocher}@master.hes-so.ch

1. The Perceptron and the Delta rule

1_activation_function.ipynb

- Observe the shape of the different activation functions proposed.
- Observe the effects of modifying the weight. How the shape of the function changes? How the first derivative changes?

Pour la fonction linéaire, le poids change la valeur de la pente.

Pour la sigmoïde, le poids "aplatit les fonctions jusqu'à les "renverser" (symétrie horizontale)

Pour la tangente hyperbolique, le poids produit un effet similaire à la sigmoïde

Pour le reste, nous sommes passé à Python 2.7 et le widget javascript ne fonctionne plus...

- Implement the activation function of a rectified Linear Unit (ReLU)

```
def ReLU(neta):  
    output=np.copy(neta)  
    output[neta<0]= 0  
    d_output = np.ones(len(neta))  
    d_output[neta<0] = 0  
    return (output, d_output)
```

Impossible d'afficher le graph!

2_perceptron.ipynb

- Use the sliders to change the weights of the perceptron and observe the effects on its output
- Select different activation functions and observe the output of the perceptron for different weight configurations

3_MLP.ipynb

- Use the sliders to change the values of the connection weights and biases, and observe the resulting changes in the MLP output
- Change the activation function and observe the changes

4_delta_rule.ipynb

You are free to modify the learning rate (ALPHA) and the number of iterations (NUMBER_OF_EPOCHS).

Try different 2D classification problems and observe the behaviour of the algorithm in terms of:

- Learning rate needed
- Convergence speed
- Oscillations

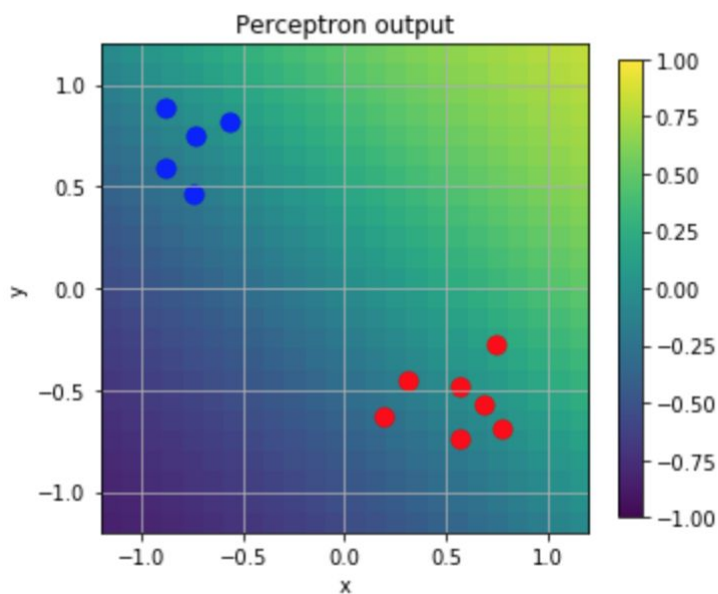
Bare in mind that, in the current implementation, the parameters (weights and bias) are initialized randomly every time you launch the cell.

Create some datasets as it is shown, and perform the following tests:

1. What happens if the boundaries between both classes are well defined?

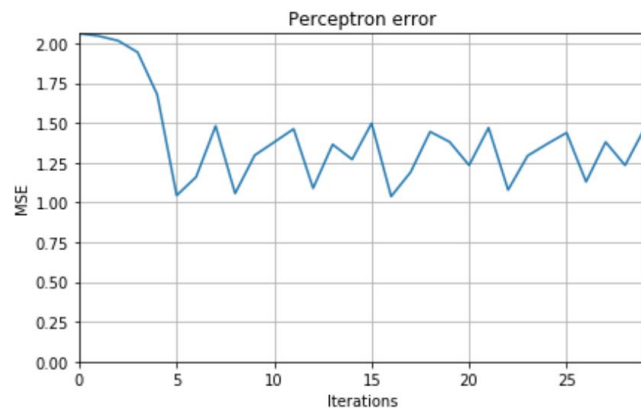
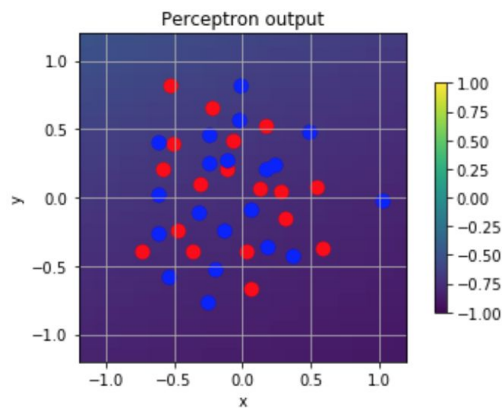
La séparation se fait bien, la frontière est claire. L'erreur diminue rapidement.

TypeError: unsupported operand type(s) for *: 'NoneType' and 'float'



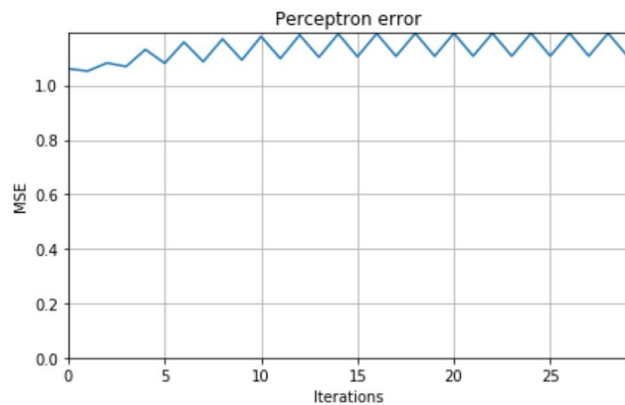
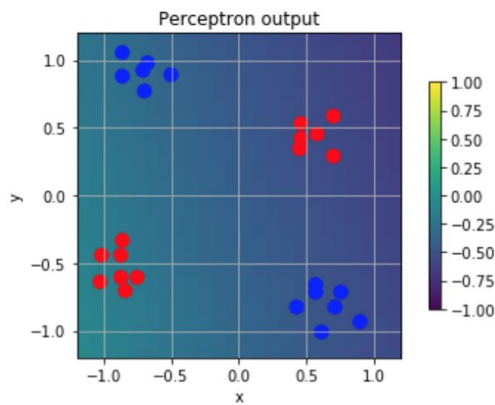
2. What happens if the classes overlap? What could you say about oscillations in the error signal?

La séparation n'est pas bonne et le signal d'erreur oscille sans diminuer.



- What happens if it is not possible to separate the classes with a single line? What could you say about local minima?

La fonction essaie tant bien que mal de trouver une séparation mais celle-ci change constamment. L'erreur ne diminue pas mais remonte.



2. Backpropagation

You are free to modify the learning rate (ALPHA) and the number of iterations (NUMBER_OF_EPOCHS).

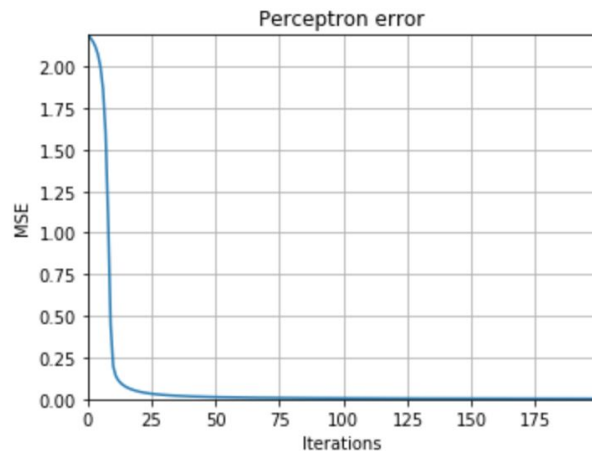
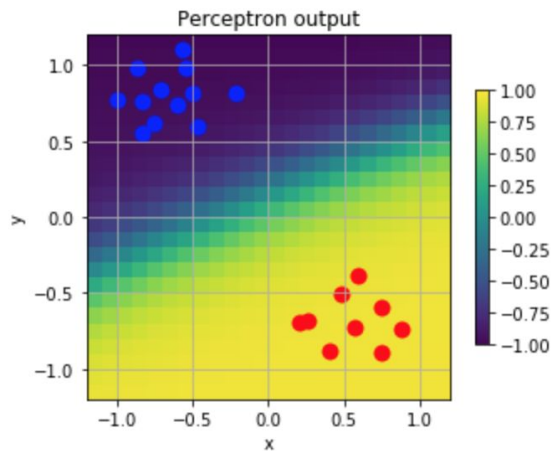
Try different 2D classification problems and observe the behaviour of the algorithm in terms of:

- Learning rate needed
- Convergence speed
- Oscillations

Bare in mind that, in the current implementation, the parameters (weights and bias) are initialized randomly every time you launch the cell

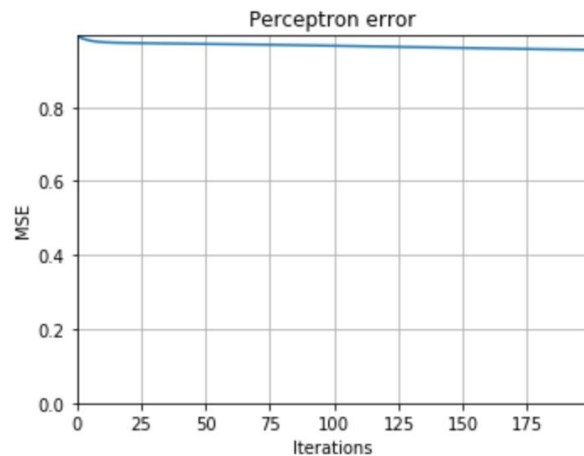
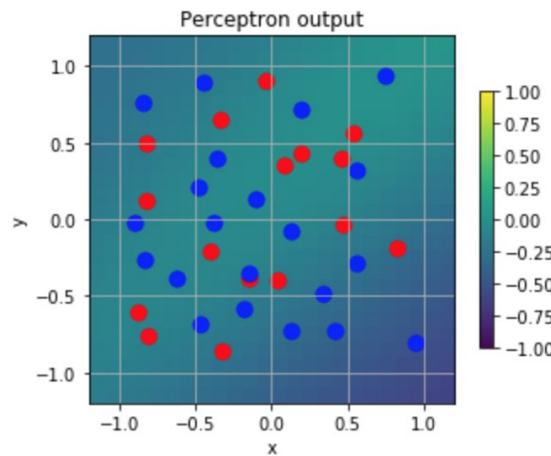
- What happens if the boundaries between both classes are well defined?

La séparation se fait bien, la frontière est claire. L'erreur diminue rapidement.



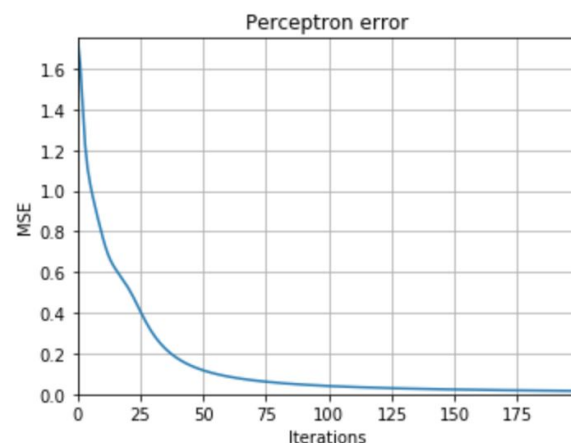
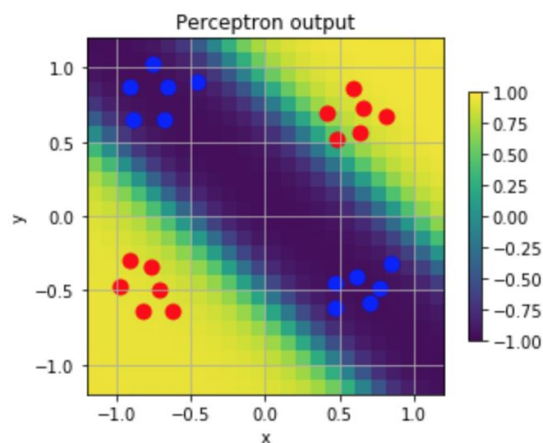
2. What happens if the classes overlap? What could you say about oscillations in the error signal?

La séparation n'est pas bonne mais le signal d'erreur n'oscille pas et a tendance à diminuer puis se stabiliser.



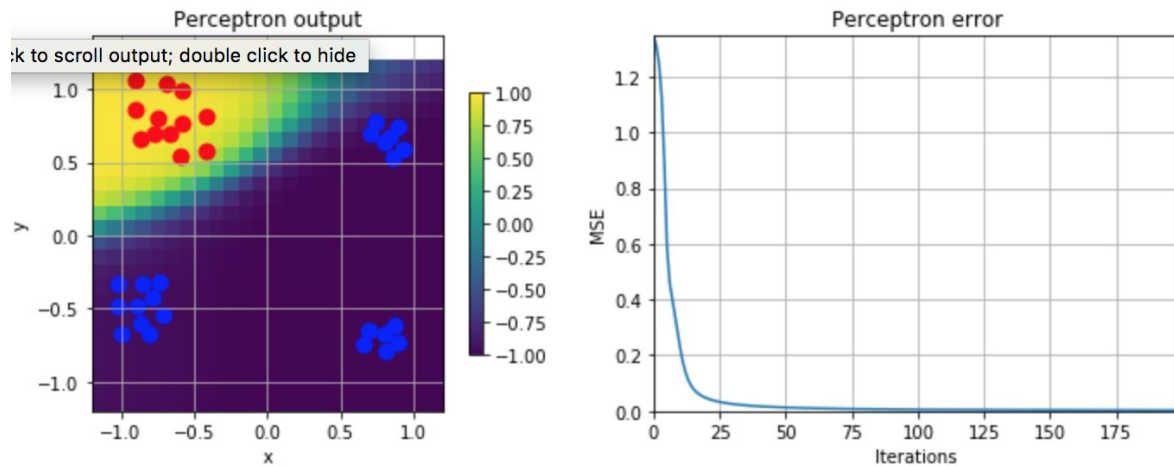
3. What happens if it is not possible to separate the classes with a single line? What could you say about local minima?

Le système a réussi à classer correctement les groupes. L'erreur n'oscille pas et diminue jusqu'à se stabiliser.



4. What happens if the points of one of the classes are separated in subgroups (blobs)?

Le système classe correctement tous les groupes et l'erreur tombe à 0.

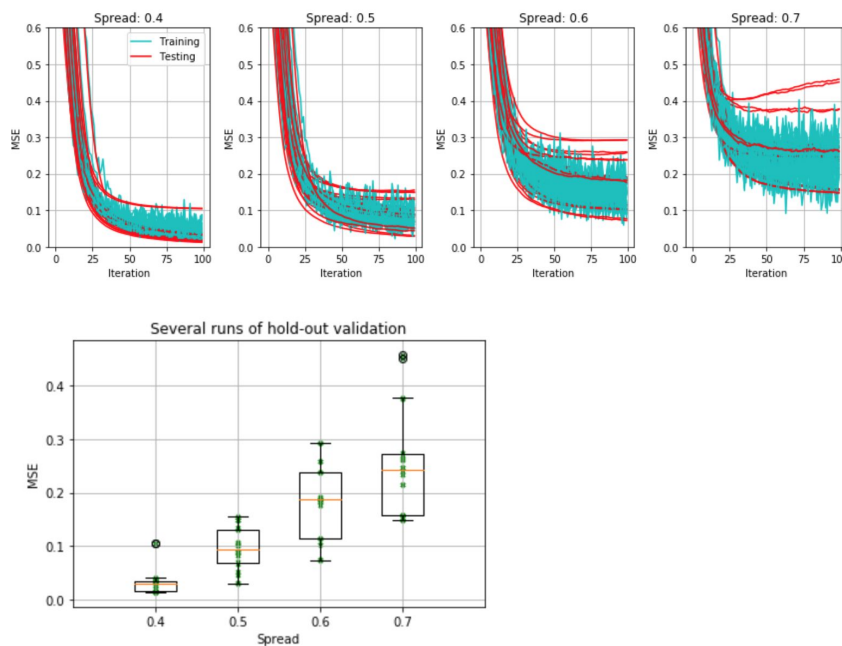


Source code with momentum

Nous avons utilisé la solution fournie.

- Run notebooks 7 and 8, provide the final plots MSE vs spread and comment the difference between results

Hold out validation:

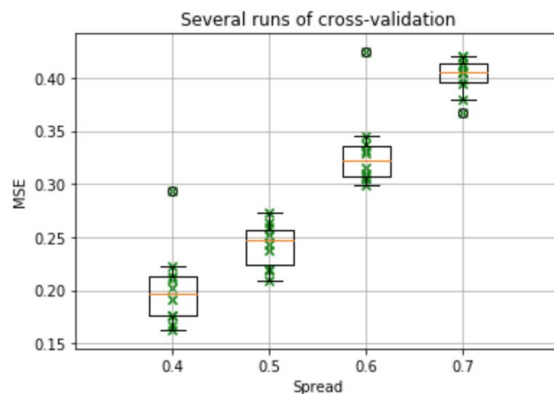


On peut voir deux différences:

- Le MSE grandit en fonction de l'étendue du dataset

- Le MSE oscille en fonction de plus en plus avec le nombre d'itérations et un plus grand "spread".

Cross validation:

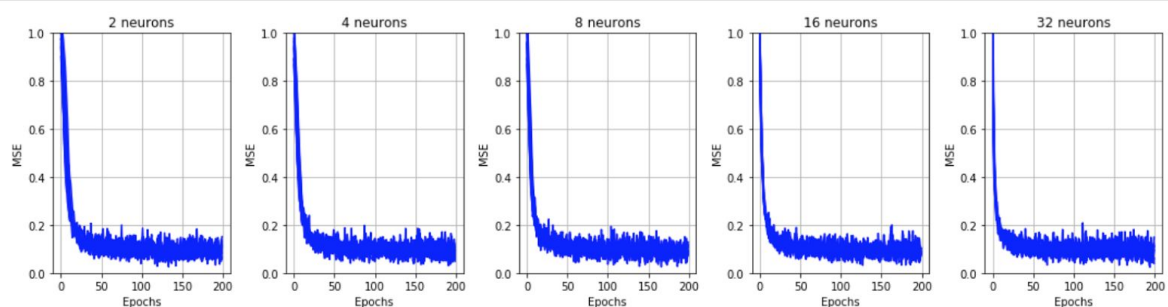


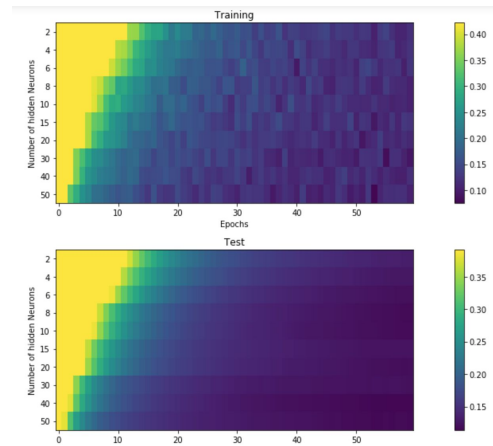
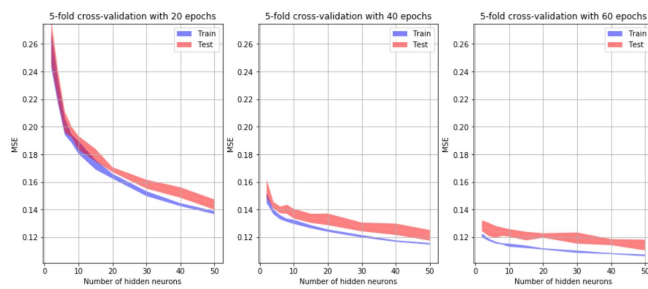
Le graph ressemble pas mal au premier, ce qui était à prévoir. La première différence est que le changement du spread sur le cross-validation réduit l'oscillation après plusieurs itérations. Mais on peut aussi voir que le Hold Out Validation montre de meilleurs résultats en général:

La médiane est entre 0.02 et 0.25 pour le Hold Out Validation, et entre 0.21 à 0.43 pour le cross Validation.

- Run notebook 9 for three different spread values (e.g., 0.3, 0.5 and 0.7), describe the final model chosen and justify your selection (e.g., based on the plots of MSE vs parameters)

Avec un spread de 0.6:





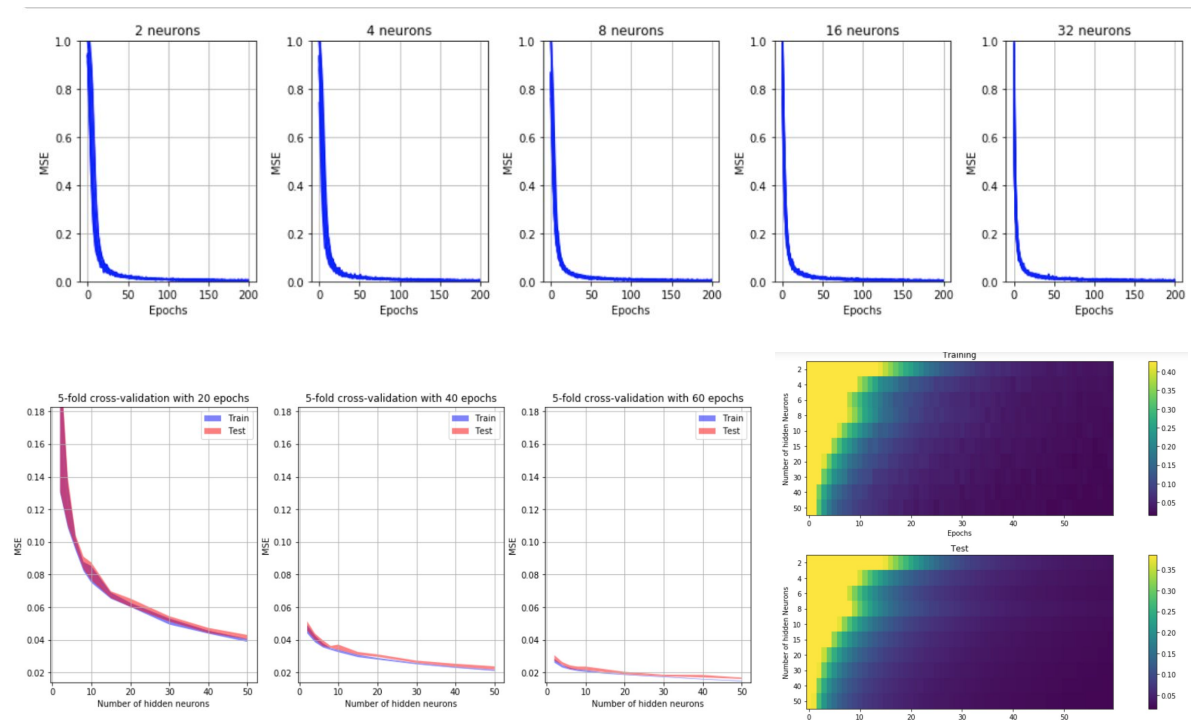
MSE training: 0.136377381704

MSE test: 0.142188577313

Confusion matrix:

```
[[ 98.  2.]
 [  6. 94.]]
```

Avec un spread de 0.35:



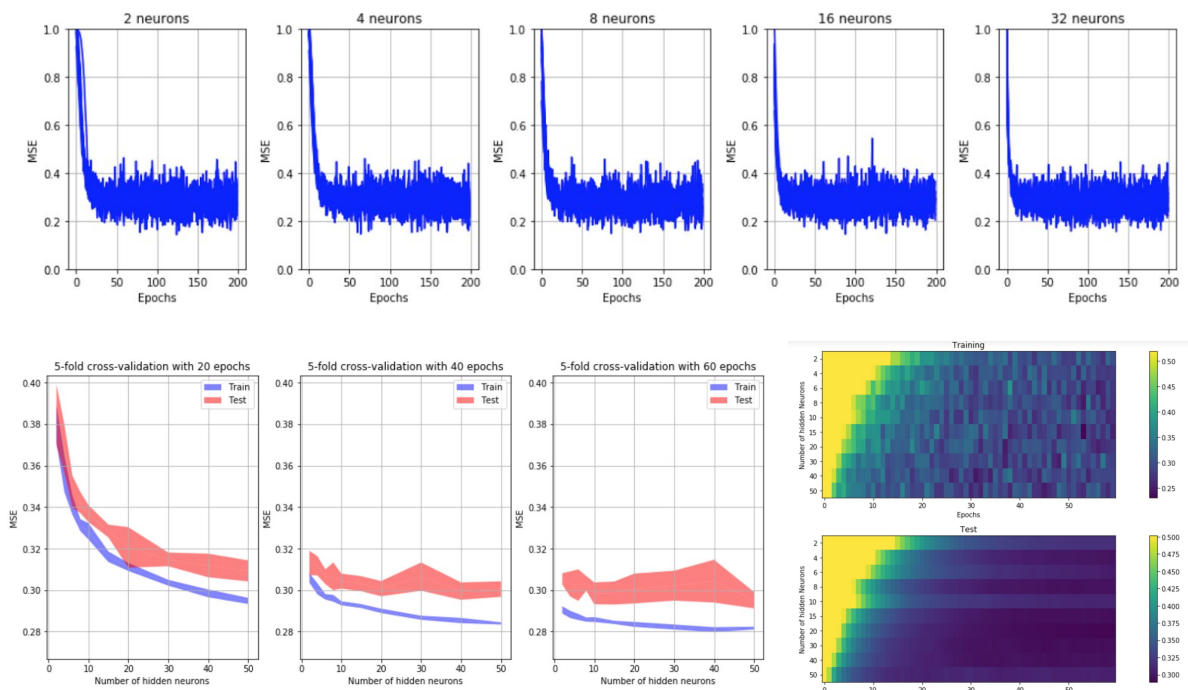
MSE training: 0.0420691277903

MSE test: 0.0428412490904

Confusion matrix:

```
[[ 100.  0.]
 [  0. 100.]]
```

Avec un spread de 0.8:



MSE training: 0.301120255298

MSE test: 0.309727598611

Confusion matrix:

```
[[ 92.  8.]
 [ 13. 87.]]
```

Après avoir lancé le notebook en changeant le nombre d'époques et de neurones cachés, on peut en conclure que:

- Le MSE ne se réduit plus beaucoup après environ 50 itérations. Mais que au moins 20 à 30 itérations sont nécessaires pour minimiser l'erreur.
- Entre 40 et 60 itérations, le nombre de neurones dans la couche cachée n'affecte plus trop le résultat. 4 à 8 neurones sont nécessaires pour avoir un bon résultat.
- Augmenter le nombre d'itérations ou le nombre de neurones a le même effet sur le training set et le test set.

Dans notre cas, le meilleurs résultat est donné par 40 à 50 pour l'époque et 4 à 8 pour le nombre de neurones dans la couche cachée. Pour le spread, 0.35 nous donne le meilleurs résultat. Le MSE n'oscille pas et le résultat du training set est aligné avec celui de test set.