



# DD2437 Presentation Lab 1

Samuel Leonardo Gracio 08/10/18  
Martin Verstraete  
Mastafa Foufa

# Part I: Classification with a single-layer perceptron

## Linearly-separable data

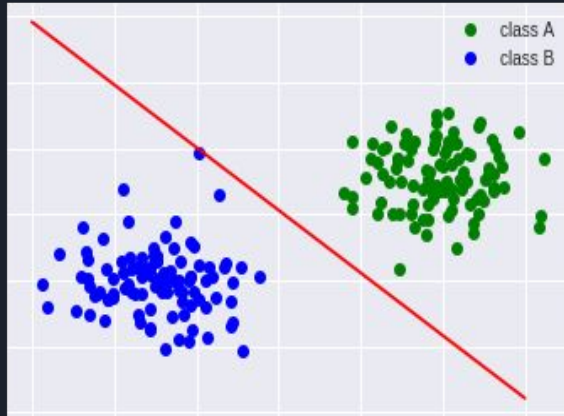


Figure 1: Classification with Delta Rule

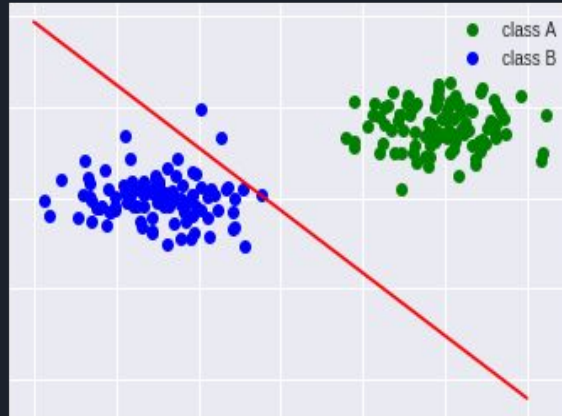


Figure 2: Perceptron Learning rule

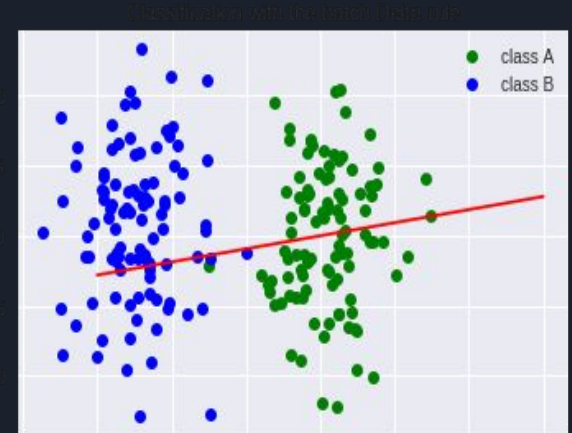


Figure 3: Delta rule without the bias term in the weight matrix

# Non linearly-separable data

Classification with the batch Delta rule

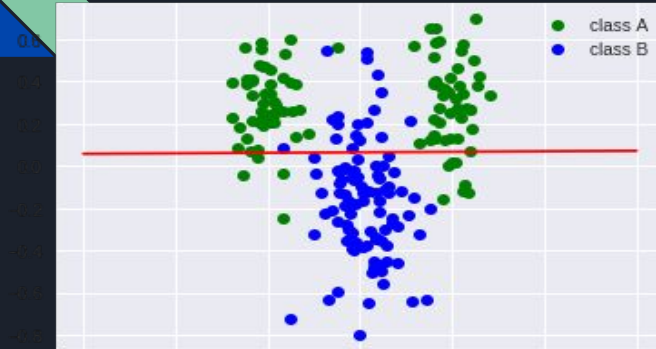


Figure 4 : Delta Rule with non linearly separable dataset

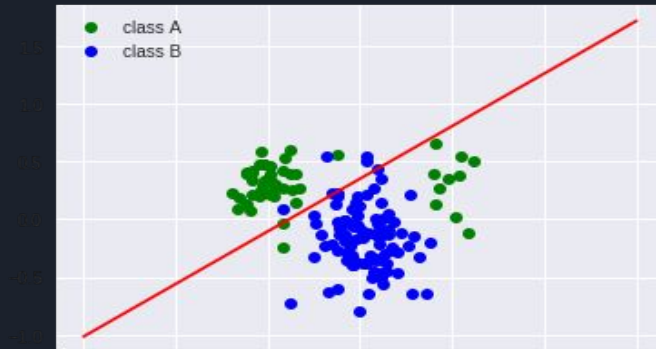


Figure 5: Fourth scenario

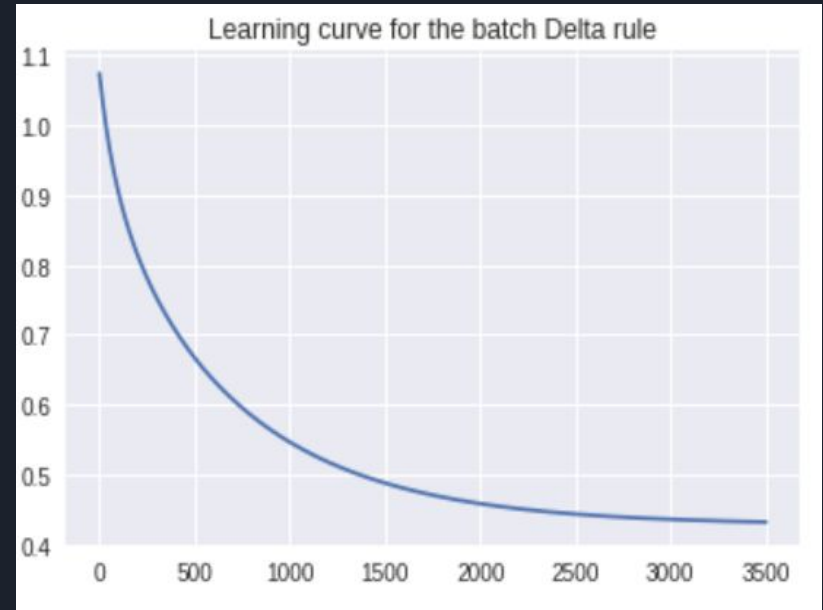


Figure 6: Learning curve

# Part I: Classification with a two-layer perceptron

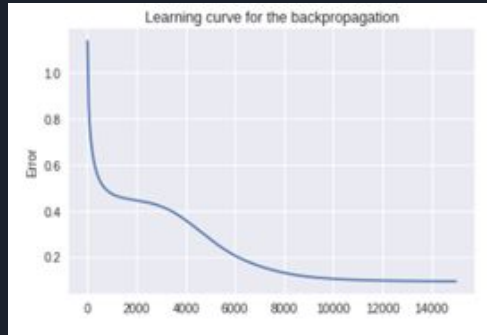


Figure 6

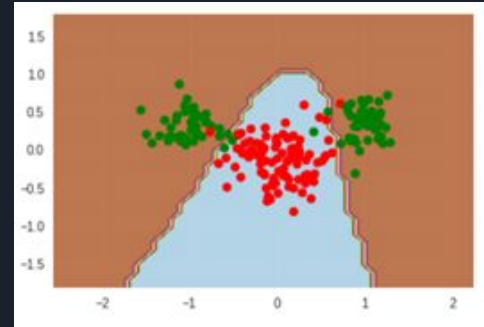


Figure 7

## Misclassifications for the fourth scenario

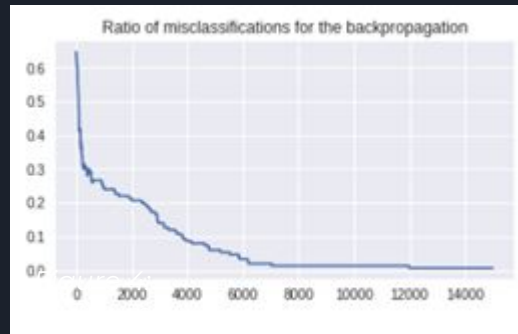


Figure 8: Learning curve



Figure 9: Error curve

# Part I: Classification with a two-layer perceptron

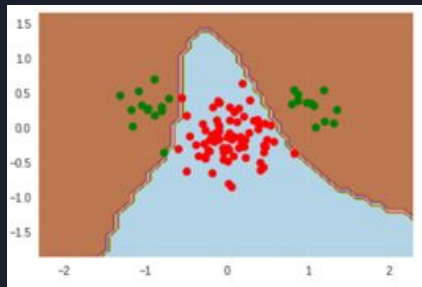


Figure 10: First scenario

Ratio of  
misclassifications = 0.03

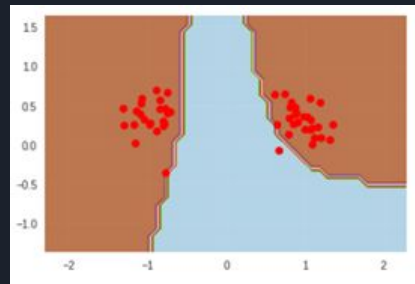


Figure 11: Second scenario

Ratio of  
misclassifications = 0.04

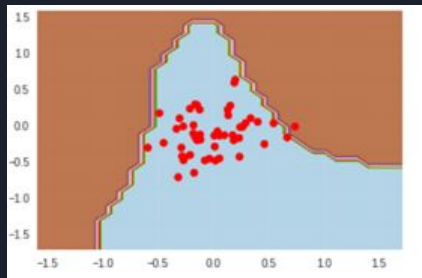


Figure 12: Third scenario

Ratio of  
misclassifications = 0.08

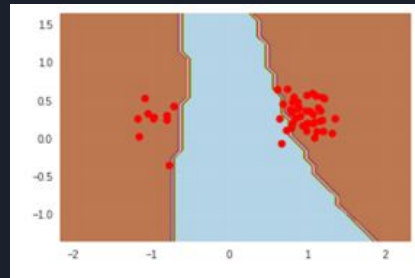
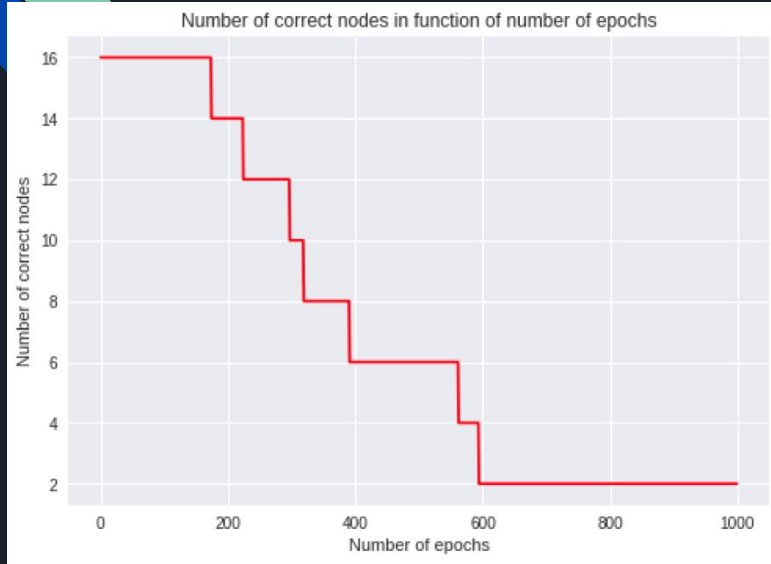


Figure 13: Fourth scenario

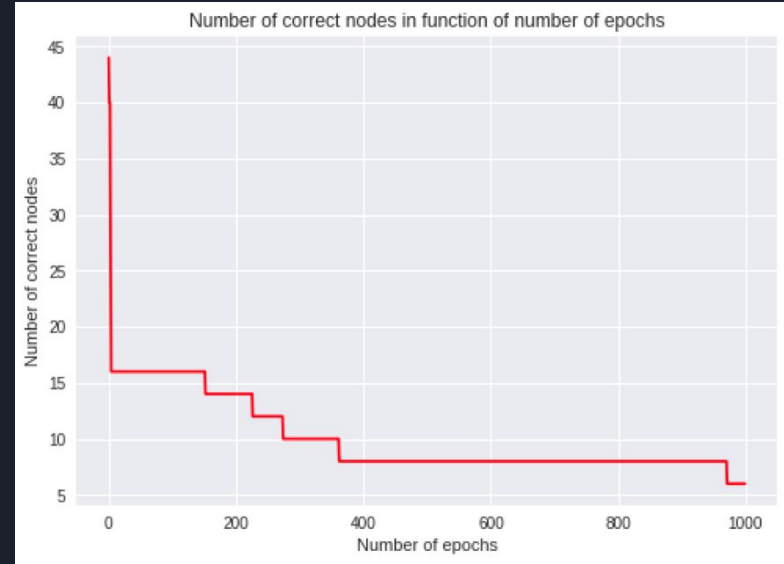
Ratio of  
misclassifications = 0.12

# Autoencoder



*Number of correct nodes for 8-3-8*

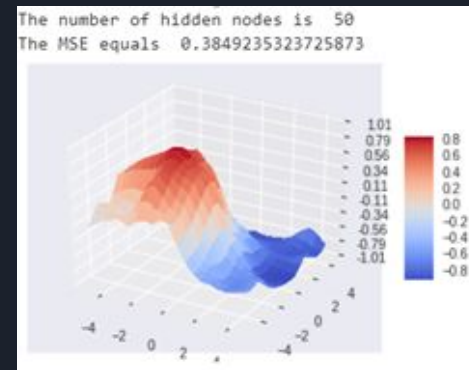
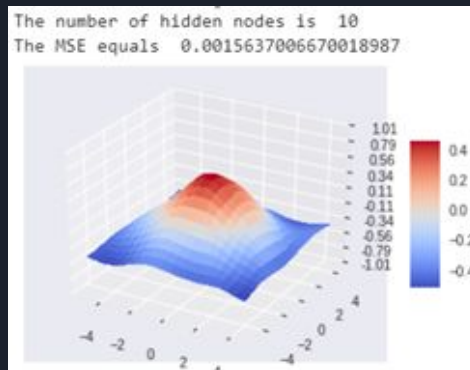
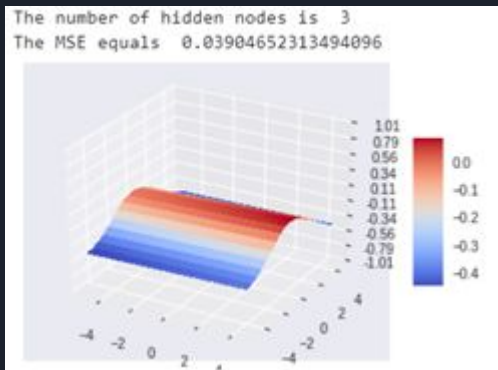
- Always converging, inputs = outputs
- Internal code = compressing the data (size = 8) into lower dimension (size = 3)



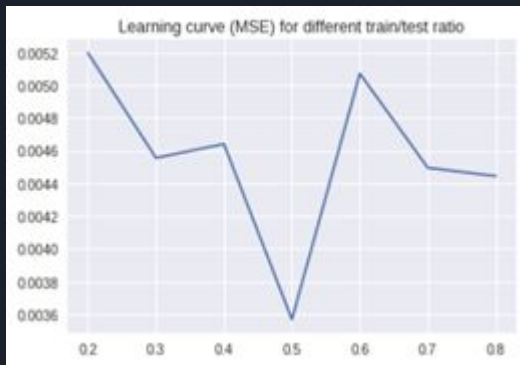
*Number of correct nodes for 8-2-8*

- If size of hidden layer = 2, not able to code
- Auto-encoders are good to reduce the dimension of the data.

# Function approximation



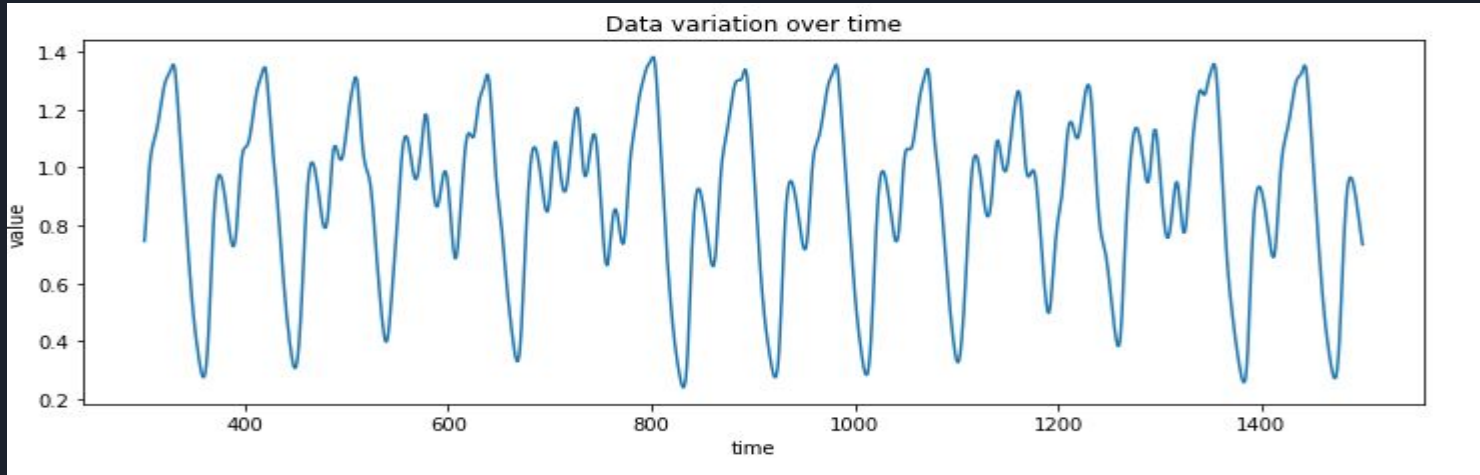
*Different approximations of the Gaussian function*



The train/test ratio does not have a huge impact on the error which stays relatively low. But it seems that the error is the lowest when there as many training samples as test samples.

Increasing convergence with eta

## Part II : Time series

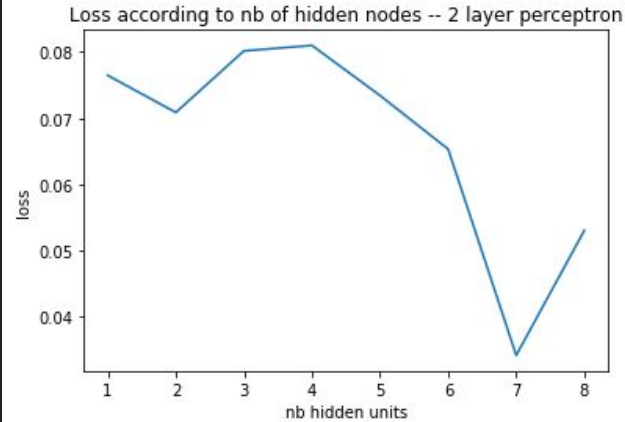


Influence of the number of hidden nodes, of the regularization term  
Comparison between two and three layers  
Comparison between the computed time series and the expectation

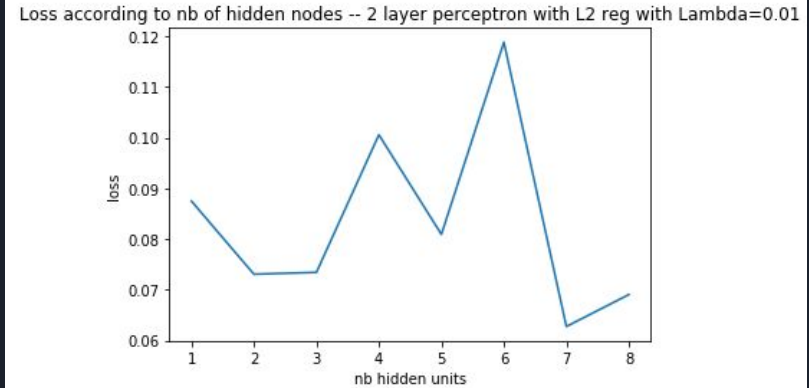


# Two-layer perceptron

Best error mse: 0.03419303819537163

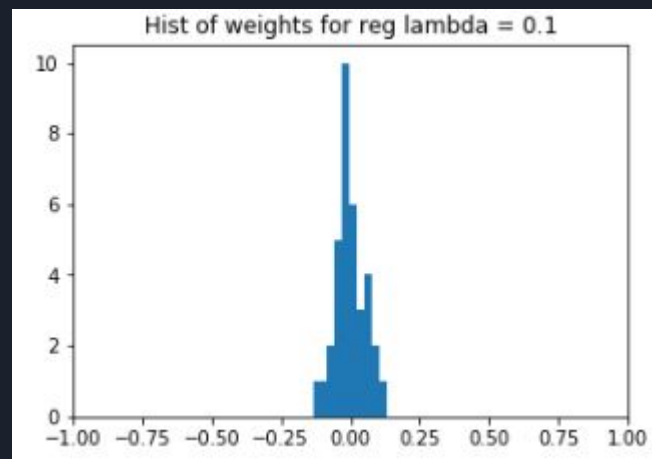
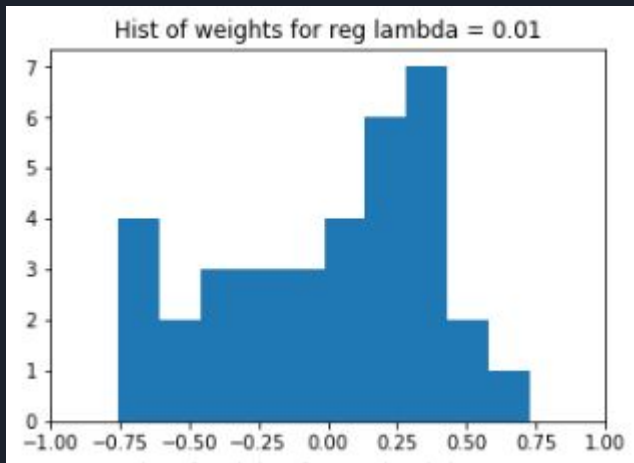


Best error mse: 0.0627233624458313



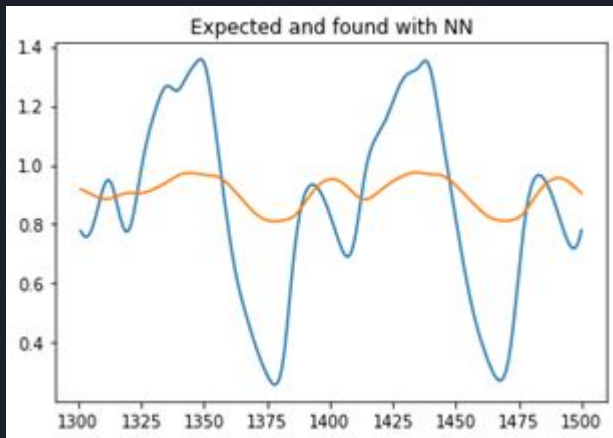
Without penalty, the lowest error is obtained with 7 hidden nodes and equals 0.03.

With a regularization term, it is still with 7 hidden but it equals 0.06.



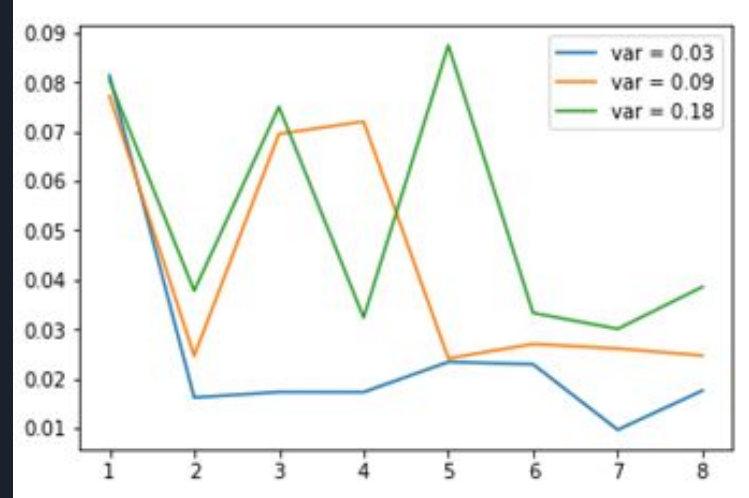
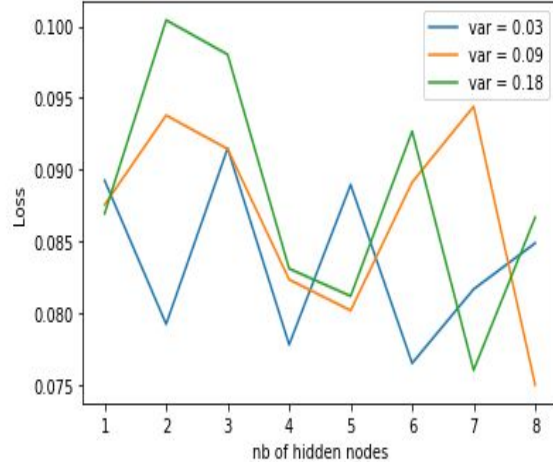
This is our result with 7 hidden nodes and a regularization term equal to 0.65.

We need to use an additional layer



# Three-layer perceptron

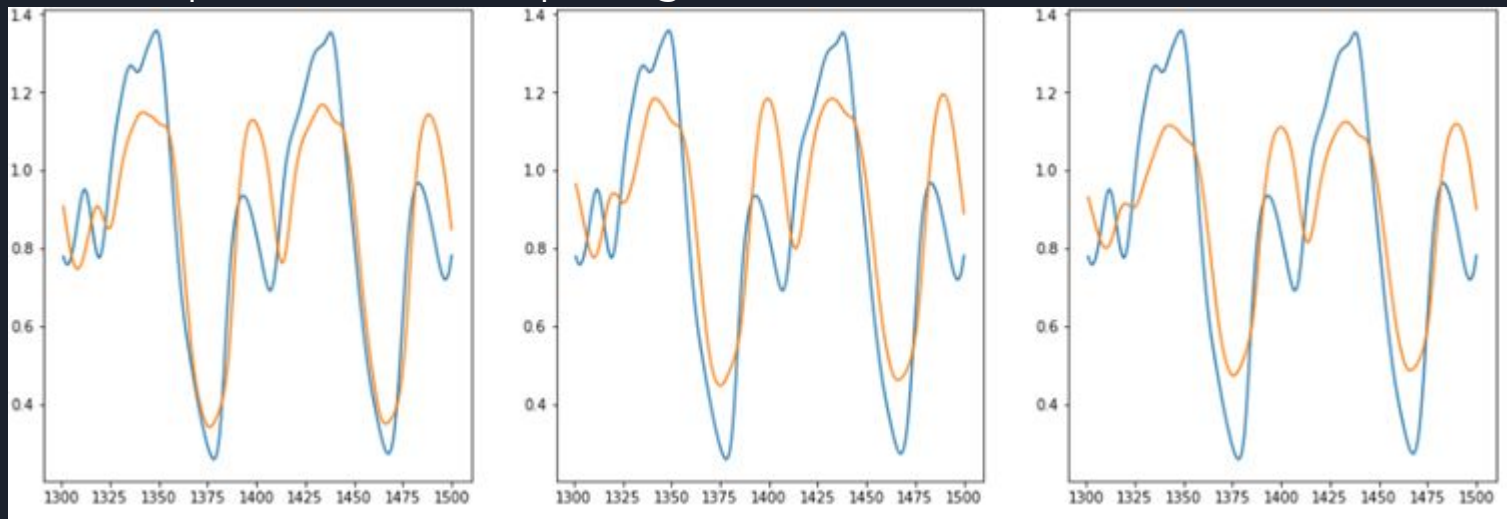
Loss vs nb of hidden nodes in second hidden layer -- Noisy gaussian data with reg in hidden layers



Validation error for different values of noise, with and without regularization

We can see that the error is lower without regularization so we won't use any

## Expected and computing values of the time series



First, the approximation of the time series for three values of variance of noise

In the bottom-left corner, the approximation with the two layer perceptron

