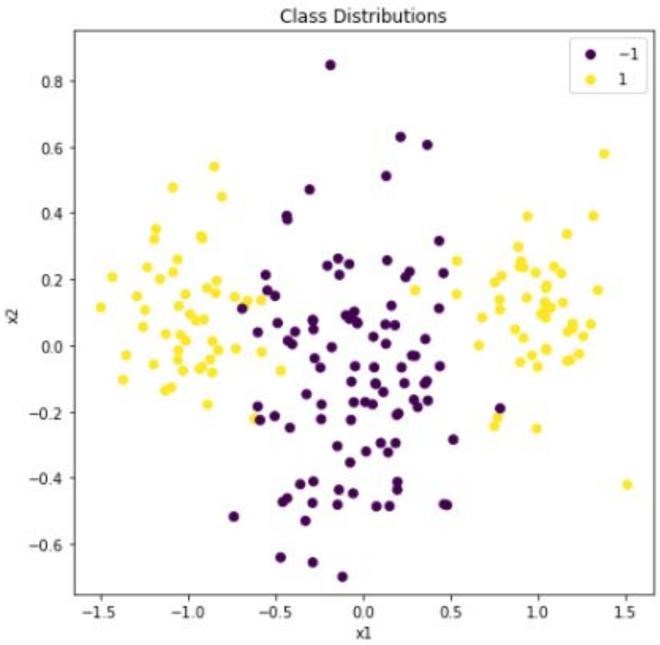
Lab 1b
Artificial Neural Networks and Deep Architectures

2 February 2023

Group 12 - Isabella Rositi, Gustav Thorén and Nicolas Wittmann

Classification with a two-layer perceptron

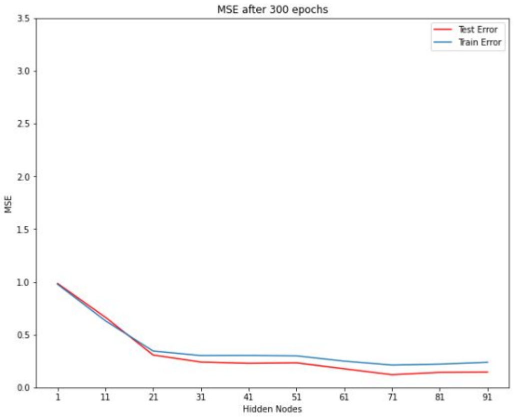


Non-Linearly Separable Data

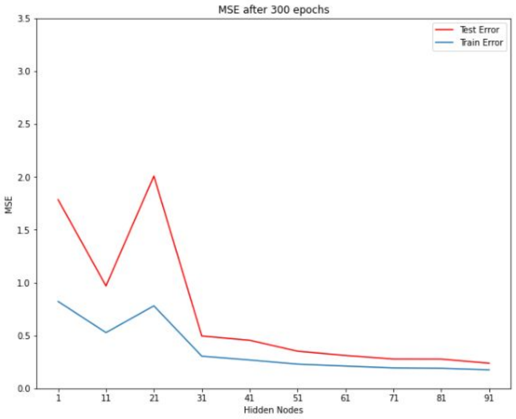
2 classes drawn from three different Gaussian distributions.

	Class A (1)	**Class B (-1)**
Mean	50% (1, 0.1), 50% (-1, 0.1)	(0, -0.1)
Standard Deviation	0.2	0.3

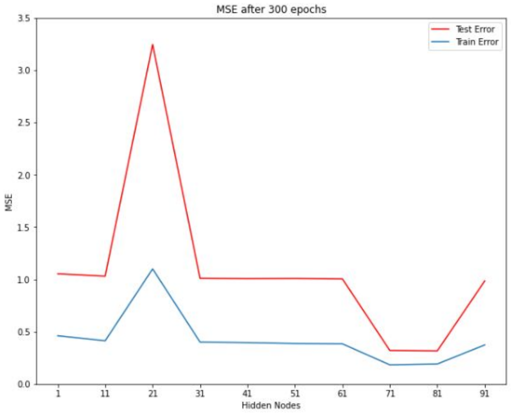
Train - Validation Error vs Hidden Nodes



25% each class



50% Class A



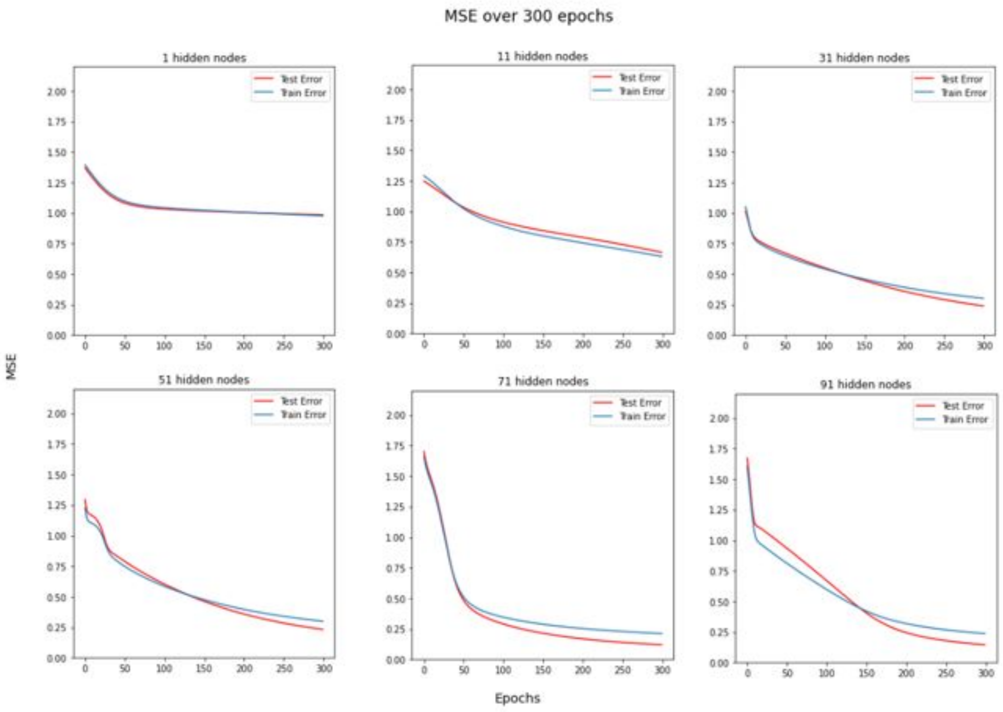
20% low and 80% high Class A

Learning Rate	0.001
Alpha	0.25
Epochs	300

Error decreases as number of hidden neurons increases

For higher numbers the error starts increasing again

Learning Curve over 300 Epochs



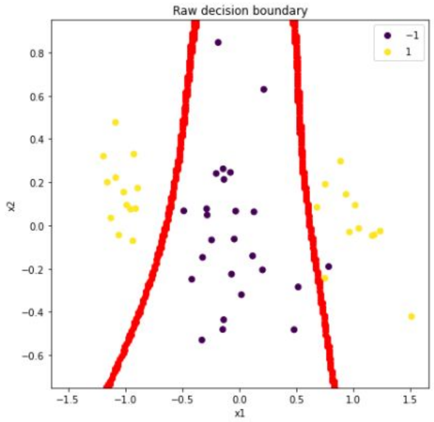
25% from each class

Error decreases as number of hidden neurons increases

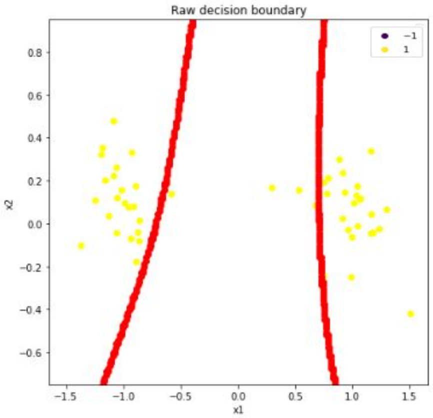
For higher numbers the error starts increasing again

MSE for 71 hidden nodes: 0.12

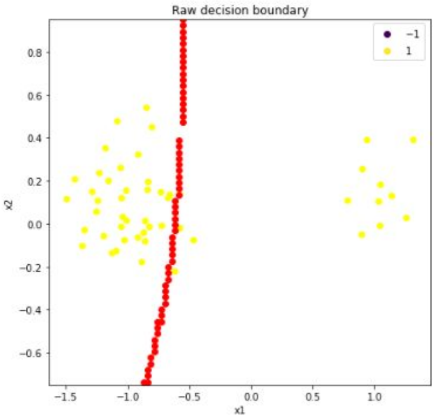
Raw Decision Boundaries



25% each class



50% Class A



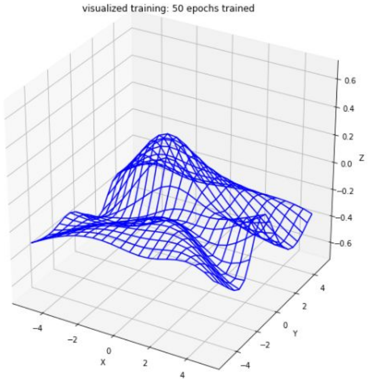
20% low and 80% high Class A

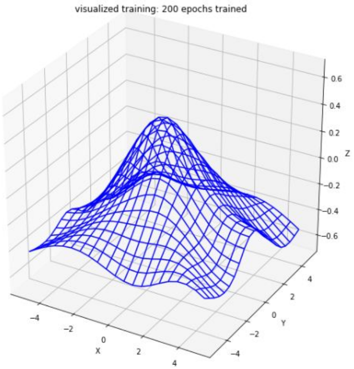
Learning Rate	0.001
Alpha	0.25
Epochs	300

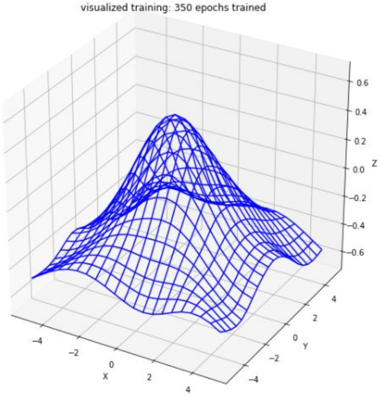
Almost perfectly classified in the test set with both classes

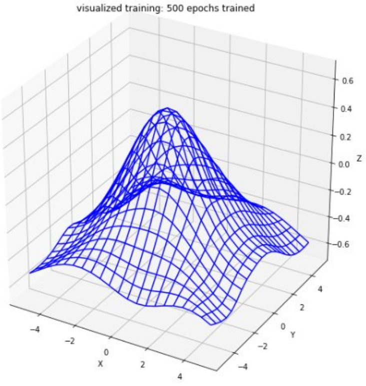
More errors in the unbalanced sets, especially in the last

Function Approximation

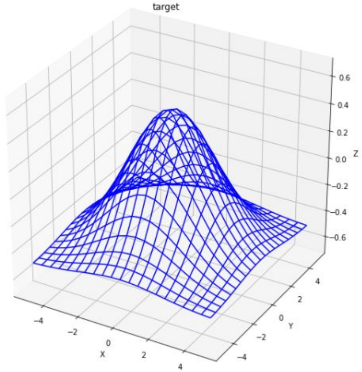






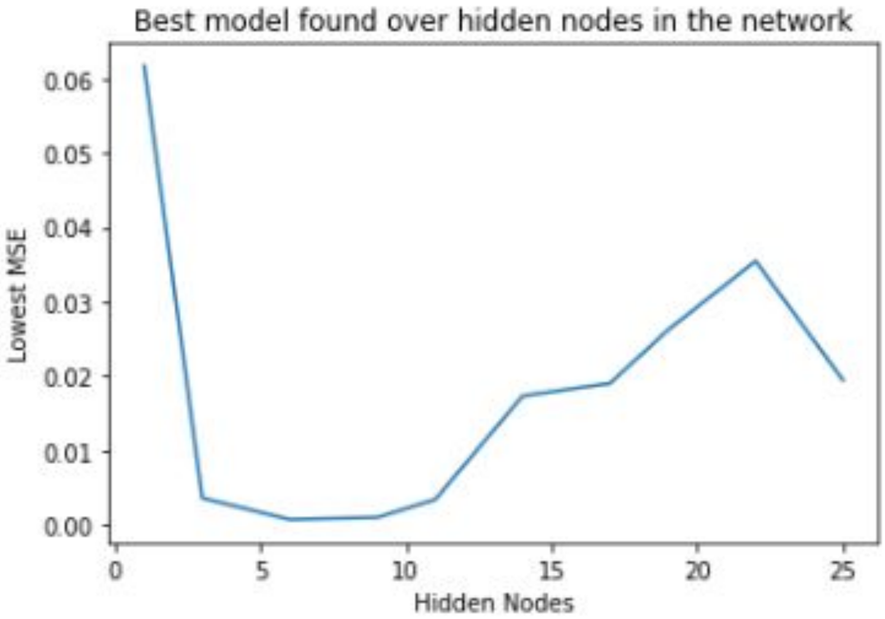


Gaussian



Target Function

Training with Different Hidden Nodes

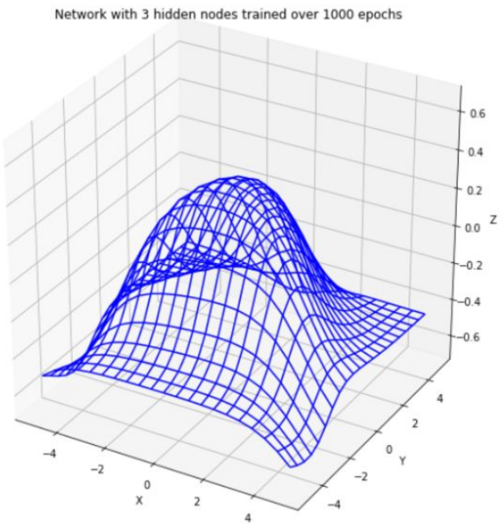


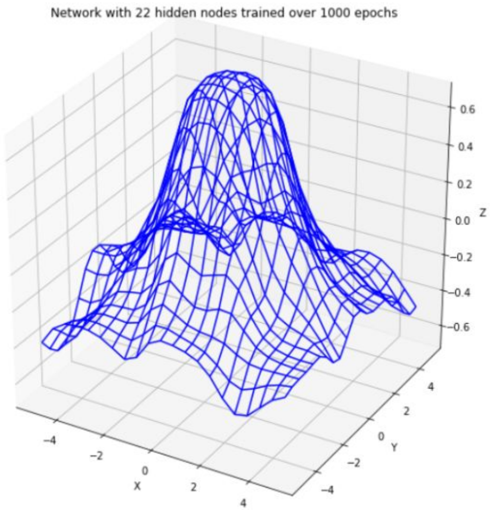
MSE vs Hidden Nodes

Lowest error for **6 hidden nodes**

The error seems decreasing after 20 hidden nodes, but mostly due to a good initialization of weights

Training with Different Hidden Nodes



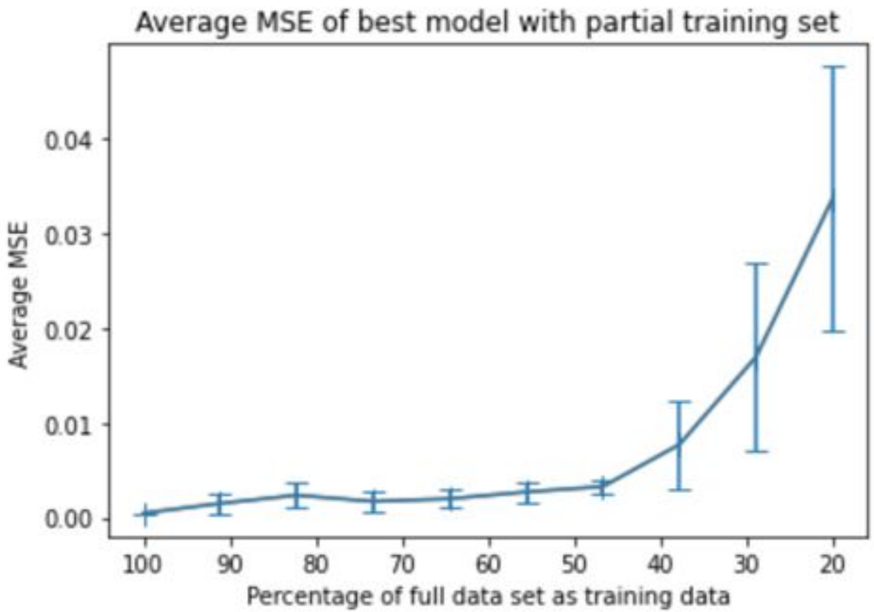


Learning Rate	0.01
Alpha	0.1
Epochs	1000

Too few hidden nodes don’t capture the function

Too many hidden nodes bring the model to get stuck in many local minima

Model Generalization



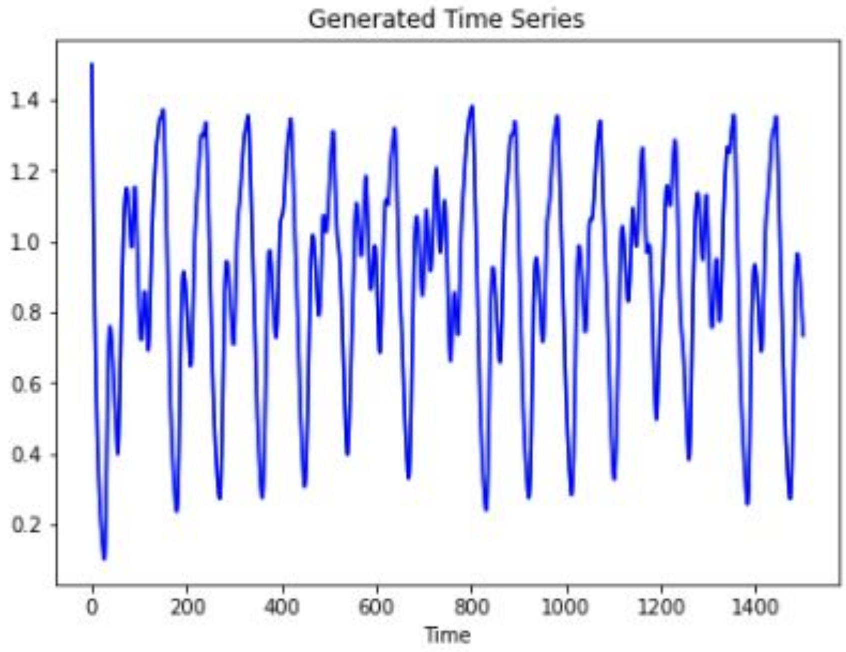
Best model

Hidden Nodes	6
Epochs	500

Consistent performance until at least 40% of the data is used for training

Less than 40% of data there are not enough to approximate the function

Time Series Prediction

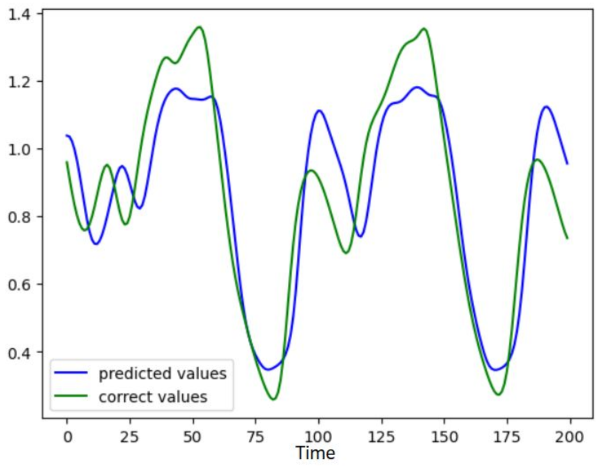


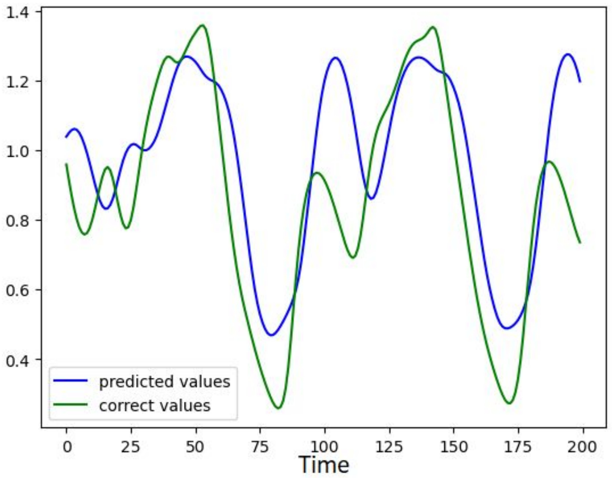
Time Series

Generated 1500 data points

Only 1200 will be used [301:1500]

Best and Worst Architectures

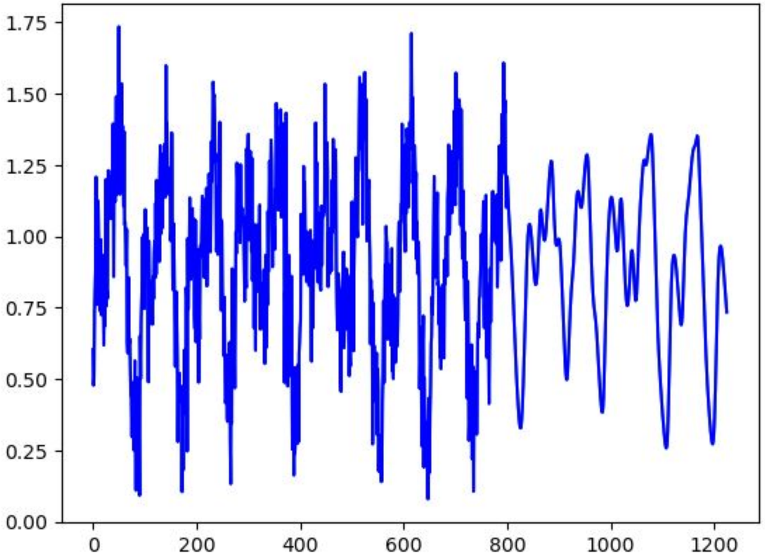




	Hidden Nodes	MSE	Variance
Best	(4,2)	0.018	1.6*10e-5
Worst	(3,6)	0.042	3.6*10e-4

Even worst architecture can still capture the overall shape of the data

Time Series Prediction with Noisy Training Data



Noisy Data

	$\lambda = 10^{-6}$		$\lambda = 10^{-4}$	
Hidden Nodes	σ =**0.05**	σ =**0.15**	σ = **0.05**	σ = **0.15**
3	0.010	0.022	0.016	0.026
6	0.014	0.018	0.017	0.020
9	0.020	0.030	0.018	0.022

Increasing the regularization parameter increases the performance of complex models

It also seems better to improve it when there is more noise

Final remarks

- Models very susceptible to randomization if few data points
- Few hidden nodes don’t capture complexity, too many hidden nodes get stuck in local minima
- Increasing the regularisation parameter can be a solution to deal with more complex model or more noisy data.

Thank you!