PTML 7: 03/06/2022

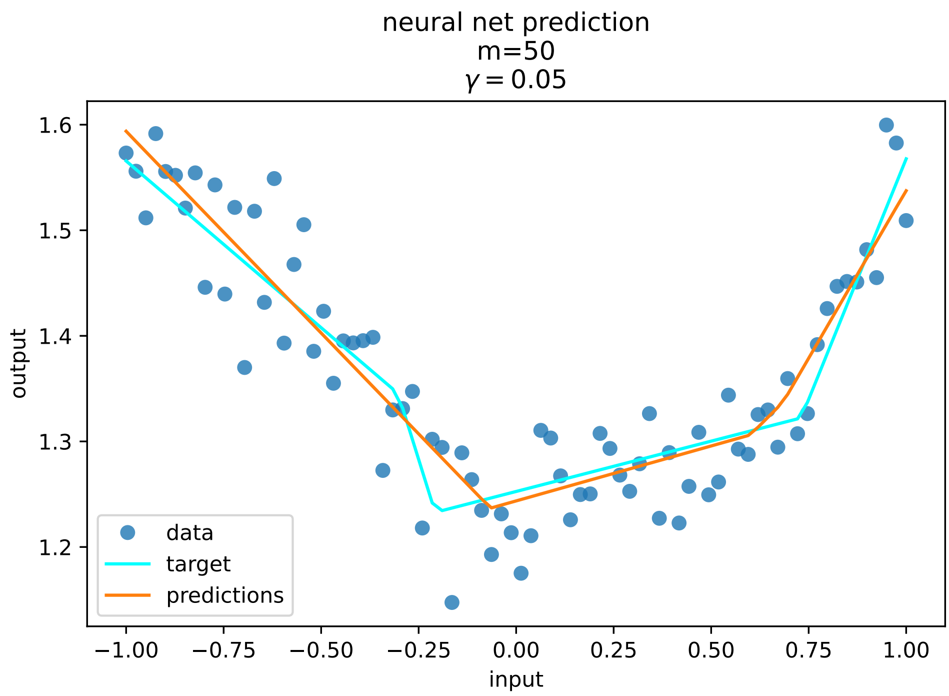


TABLE DES MATIÈRES

1 PTML 6

Many additional explanations and details, and a reference, have been added to the instructions of the previous TP. You can keep working on this one and experiment with the algorithm.

2 SIMPLICITY BIAS OF NEURAL NETWORKS

With some neural networks, it is unlikely to overfit the data. We will illustrate this with neurons that have ReLU activations. In this exercise, you can create new files taking blocks from the previous session.

To have some visual setting, we will set

— $\mathcal{X} = \mathbb{R}$

— $\mathcal{Y} = \mathbb{R}$

2.1 Target function generation

Exercice 1 : Generate a target function with a neural network that has $m = 5$ hidden layer with the same architecture has in TP6, but with a one dimensional input and ReLU activations. Generate a dataset by adding some noise to the outputs of this target function. See figure 1 for an example function.

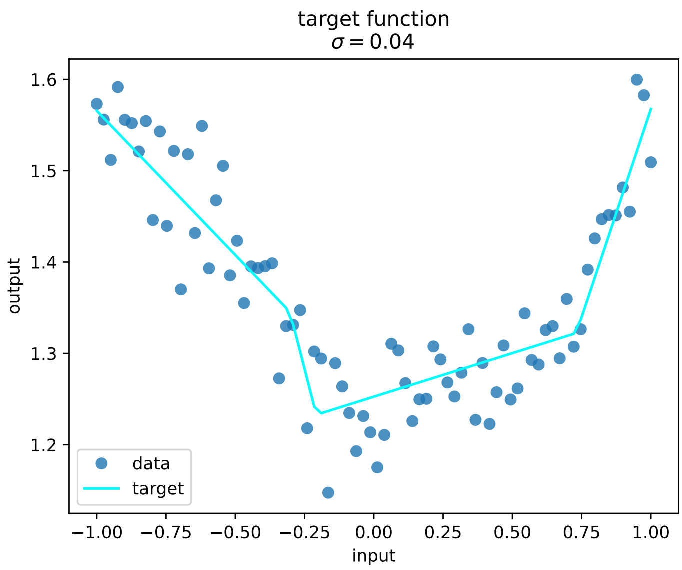


FIGURE 1 – Example target function and dataset

2.2 Network learning

Exercice 2 : Train a neural network with a varying number m of hidden layer, as follows :

2.2.1 *Initialization*

— θ is initialized uniformly in $[-\frac{1}{\sqrt{m}}, -\frac{1}{\sqrt{m}}]^{m+1}$

— Each column of w_h, that belongs to $\mathbb{R}^2$, is initialized on the sphere of radius $\frac{1}{\sqrt{m}}$.

2.2.2 *Activation function*

For the derivative of ReLU, you can use the heaviside function.
https://numpy.org/doc/stable/reference/generated/numpy.heaviside.html
https://numpy.org/doc/stable/reference/generated/numpy.maximum.html

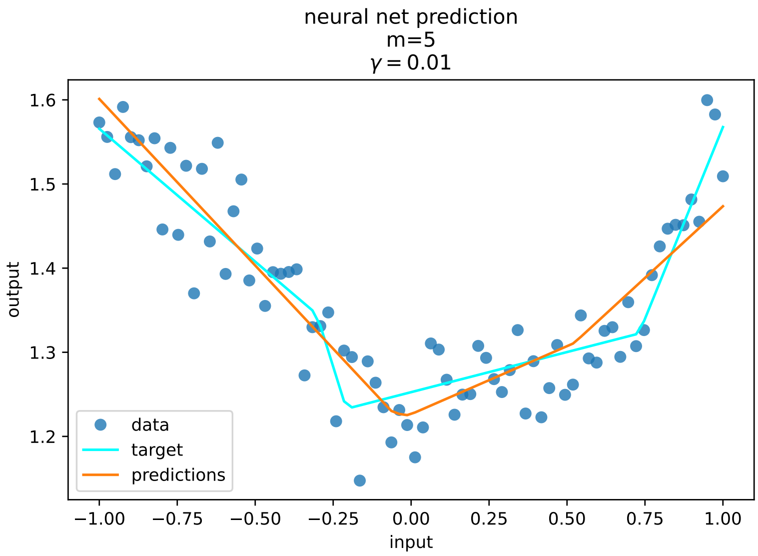
2.2.3 *Learning rate*

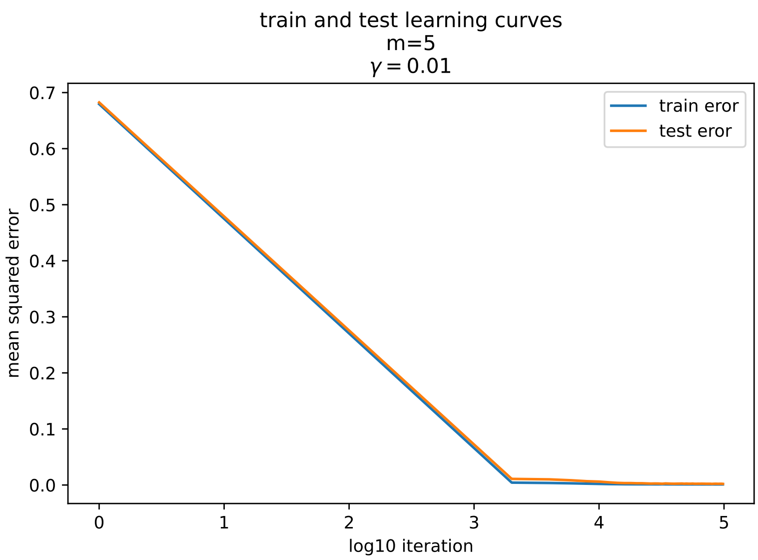
You will probably need to use small values of γ, in order to observe learning and the simplicity bias (experiment with this parameter too).

2.3 Results

Here are some example results. As the learning algorithm is stochastic, you might observe different outputs. The learning rate γ has a direct influence on the convergence.

In figure 2, we see that although the neural networks has a larger number of parameters than necessary in order to represent the target function (it has a high capacity), no overfitting has occurred. However, in figure 4, the network has not been able to approximate the target function.





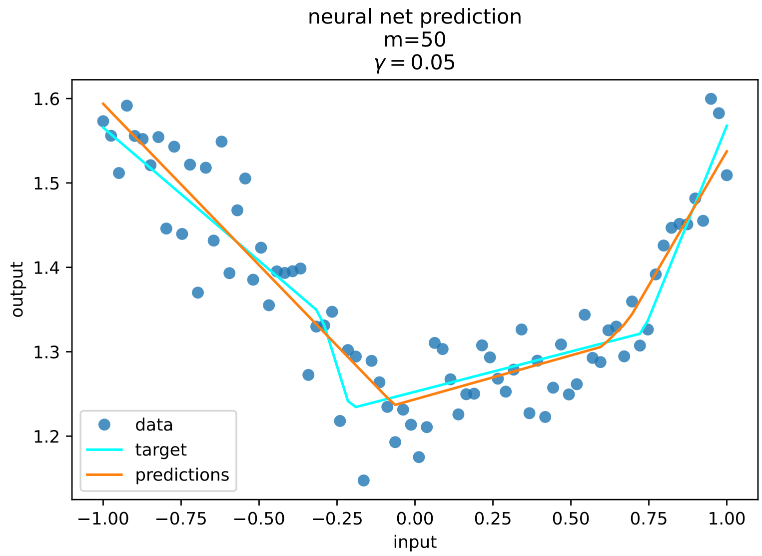
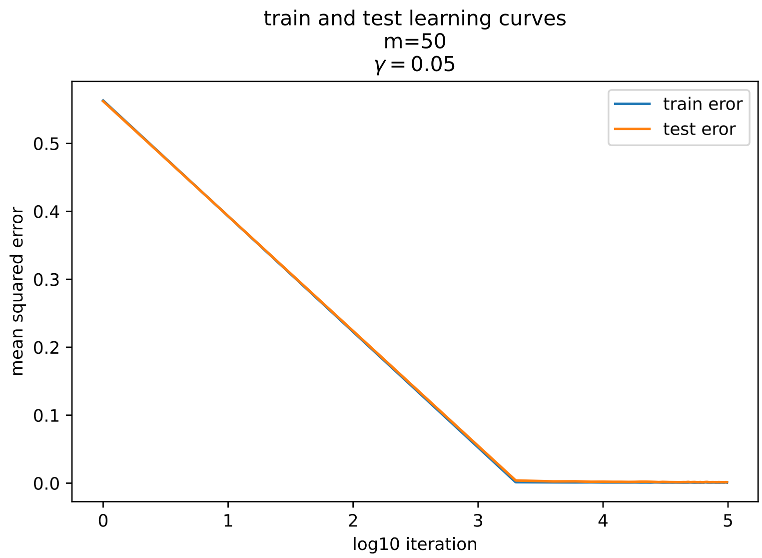


FIGURE 2 – Although the network has a high capacity, it does not overfit the data.



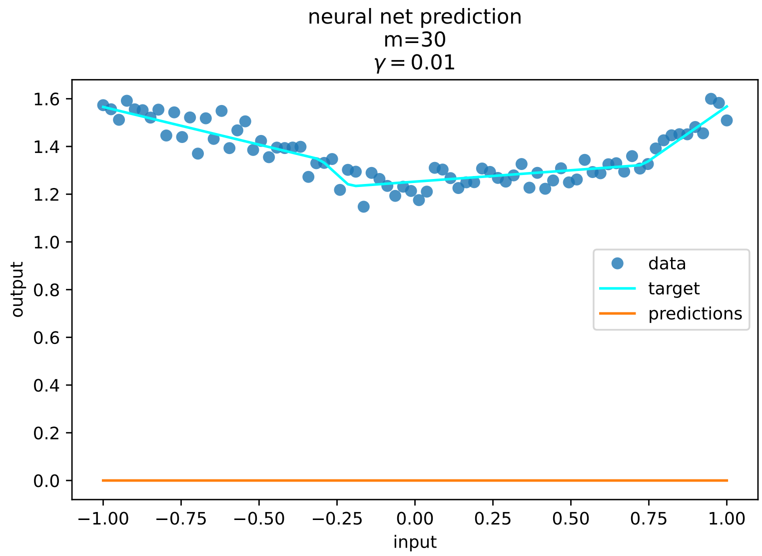


FIGURE 3 – No training has occurred in this simulation

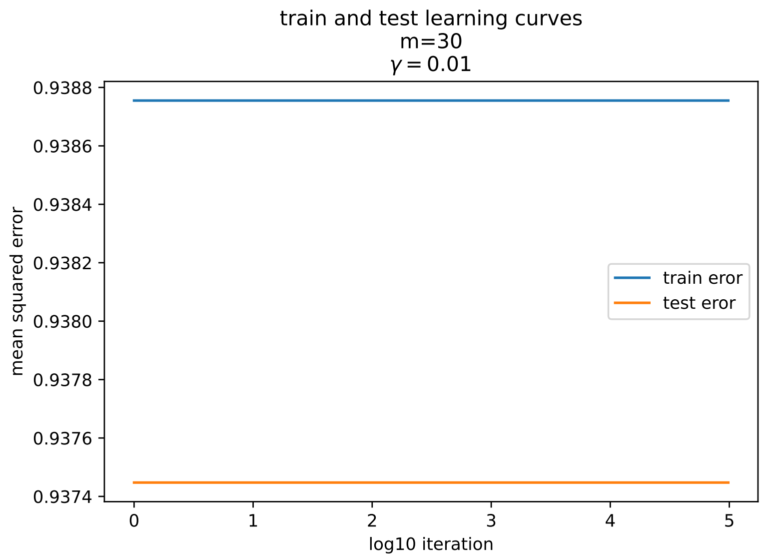


FIGURE 4 – No training has occurred in this simulation

2.4 Conclusion

These neurons tend to not overfit, although some of them have a number of parameters way larger than of the minimal space containing the target function.

See more in this post : https://francisbach.com/quest-for-adaptivity/