Deep Learning Lab
Assignment 1

Pietro Miotti

October 27, 2021

1 Polynominal Regression

1.1 Ex 1)

Please find the requested code from line 11 to 24 (function named *create_dataset*) of the submitted file *main.py*.

1.2 Ex 2)

In order to create the training and validation set I called 2 times the *create_dataset* function implemented for the previous point considering different seeds as requested. Code lines 49-50

1.3 Ex 3)

Please check the Fig:1 for the scatter plot of the training and validation set generated for the previous point. Please note the code for doing this plot is also reported in the *main.py* (from line 54 to line 63) but it is commented to avoid confusion, if you would like to re-generate the plot by yourself please uncomment it.

1.4 Ex 4)

Considering the equation of a general linear line in 1D:

$$y = wx + b \tag{1}$$

where w is the angular coefficient of the line and **b** is the bias, which represents more practically where the line intersect the y-axis.
Considering the formula of a Linear Neural Network (namely torch.nn.Linear):

$$\mathbf{Y} = \mathbf{W}\mathbf{X} + \mathbf{b} \tag{2}$$

as in Equaton (1) the **b** term represents the additive bias and hence the term which does not multiply the data.

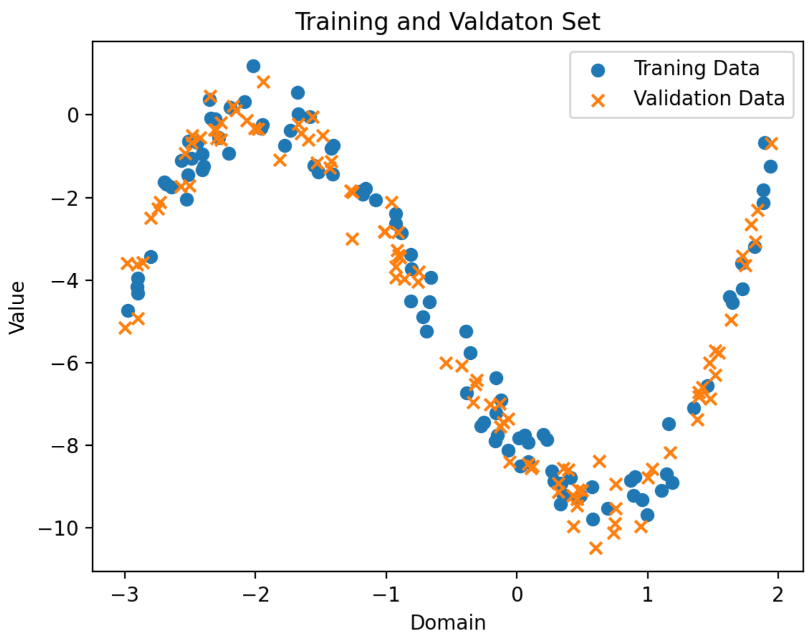


Figure 1: Training and validation points

In our case, since we are dealing with a polynomial we don't need the bias since it is already implicitly considered as the weight that multiplies the x^0 term.

1.5 Ex 5)

Please consider the code reported in the *main.py* file in order to see how I implemented the regression.
My estimate of the real $w^* = [-8, -4, 2, 1]$ is $w^{approx} = [-7.8378, -4.0569, 1.9672, 1.0036]$
In order to print the values of the model parameter please uncomment lines from 101 to 103 in *main.py*.

1.6 Ex 6)

Suitable Hyperparameters:

- **learning rate:** 0.01
- **number of iterations:** 1000

1.7 Ex 7)

Please consider the plot fig:2 for training and validation losses as function of the gradient descent iterations.

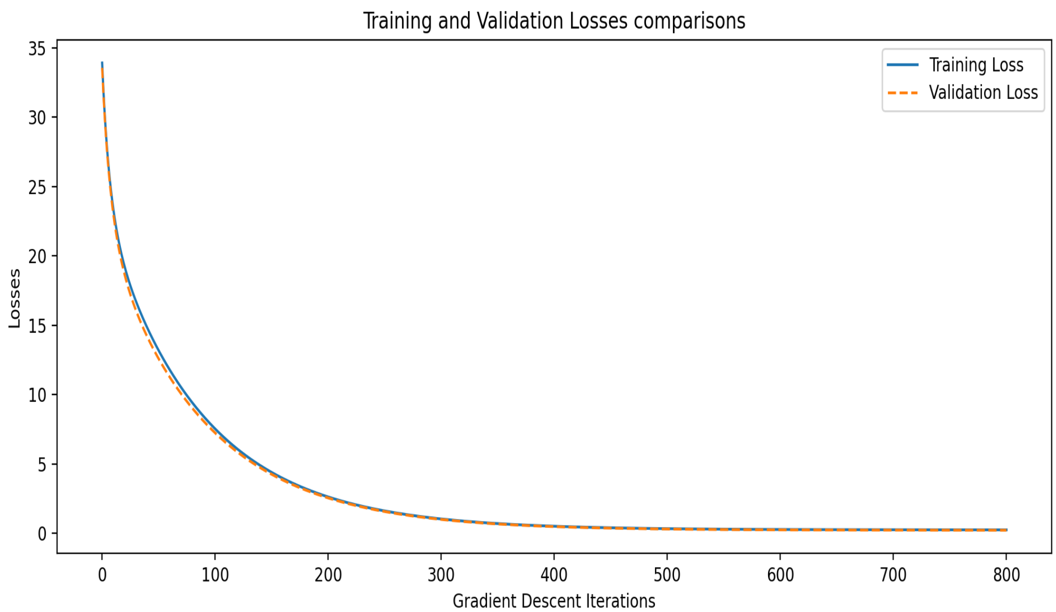


Figure 2: Training and validation points

1.8 Ex 8)

Please consider fig: 3 to see the different plots.

1.9 Ex 9)

By looking at the plots in Fig:4 we can see that by reducing the number of training points we loose in generalization over the validation data, thus in this case the error on the validation data increases when the number of training points decreases. With 50 training points the regression works fine but with 100 training points the validation error is lower. Instead by considering just 10 or 5 training points the model is overfitting since it works well on training data but does not generalize well on validation data. Please notice that for doing this plot I have changed manually the parameters of the model each time, in order to reproduce the results it is necessary to make the edits manually.

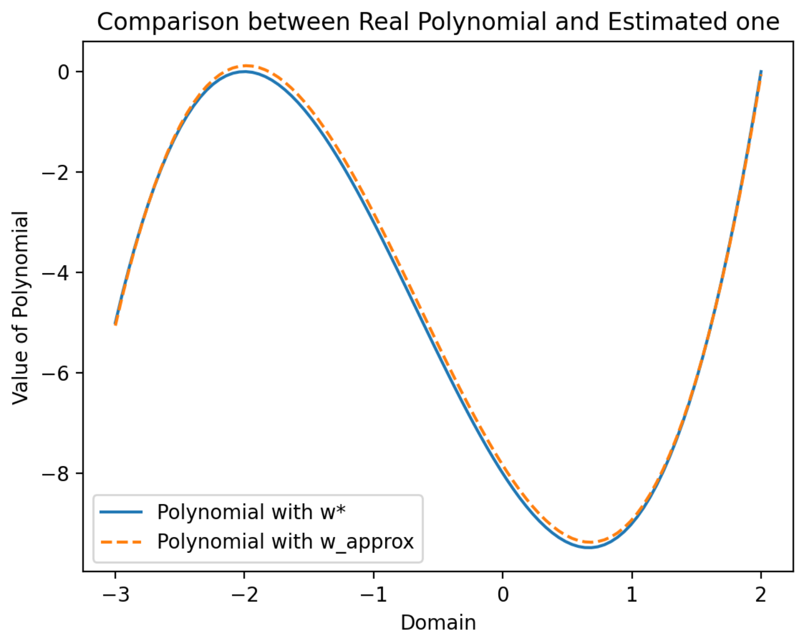


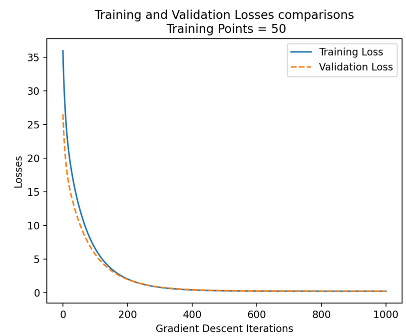
Figure 3: Real Polynomial vs Approximated Polynomial

1.10 Ex 10)

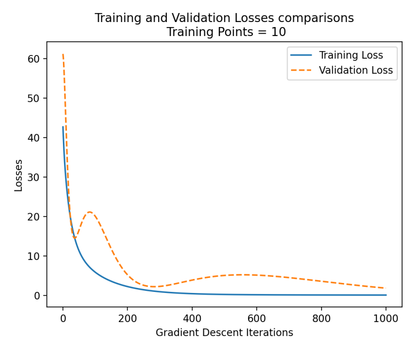
By looking at the plots in Fig:5 we can see that by adding more noise into the training data, the loss on training data increase and hence the model does not manage to find patterns in the data, thus in this case the model is underfitting the data. With σ=2 the model still manage to find a good pattern in the data, but not for the other values of σ.
Please notice that for doing this plot I have changed manually the parameters of the model each time, in order to reproduce the results it is necessary to make the edits manually.

1.11 Ex 11)

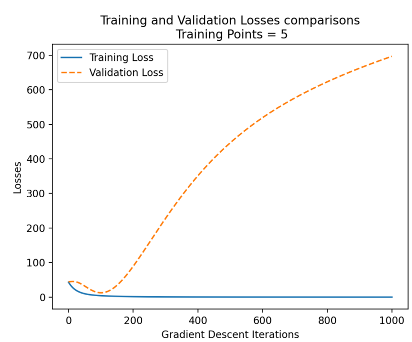
It is possible to see from fig:6 and fig:7 that increasing the order of the model doesn't solve the overfitting problem.
Please notice that for doing this plot I have changed manually the parameters of the model each time, in order to reproduce the results it is necessary to make the edits manually.



(a)

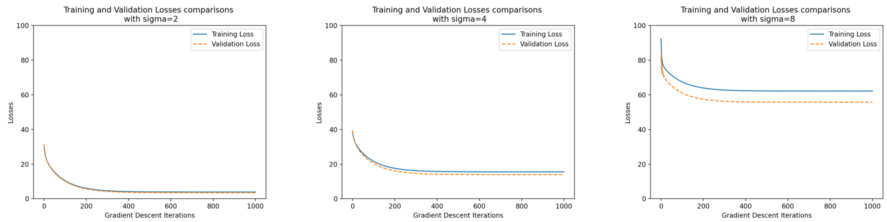


(b)



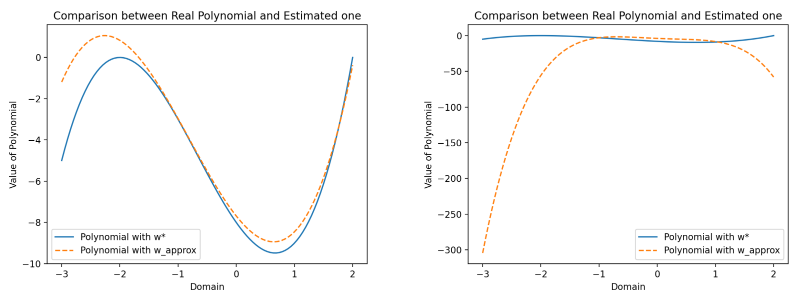
(c)

Figure 4: (a) 50 training Points (b) 10 training points (c) 5 training points



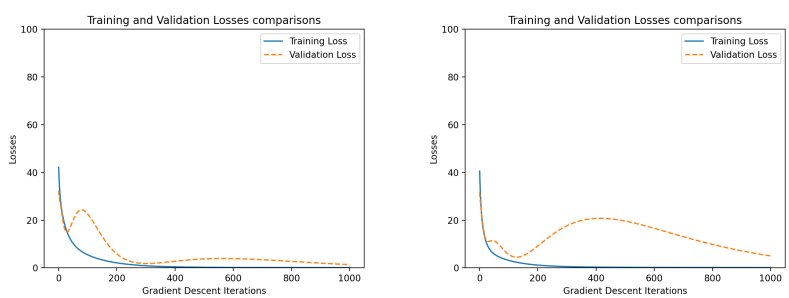
(a) (b) (c)

Figure 5: (a) σ=2 (b) σ=4 (c) σ=8



(a) (b)

Figure 6: a) Model with degree 3 b) Model with degree 4



(a) (b)

Figure 7: a) Loss model with degree 3 b) Loss model with degree 4

2 Questions

2.1 What is the difference between a local and a global minimum?

A local minimum is a point where the function is lower than at all the neighboring points. A global minimum instead is the point where the function obtains its absolute lowest value w.r.t all the other points in which that function is defined.

2.2 What does it mean if your model is overfitting/underfitting

A model is underfitting when is not able to obtain sufficiently low error value on training data. A model is overfitting when the gap between training error and validation(or testing) error is too large, thus the model does not generalize on unseen data.

2.3 How can you mitigate underfitting/overfitting?

In order to mitigate underfitting we need to make our model more complex and adding number of unknown parameters in order to find a more complex structure in data. In order to mitigate overfitting instead a possible way would be to reduce the complexity of the model in order to improve generalization, or (as we have seen in this assignment) by getting more training data.