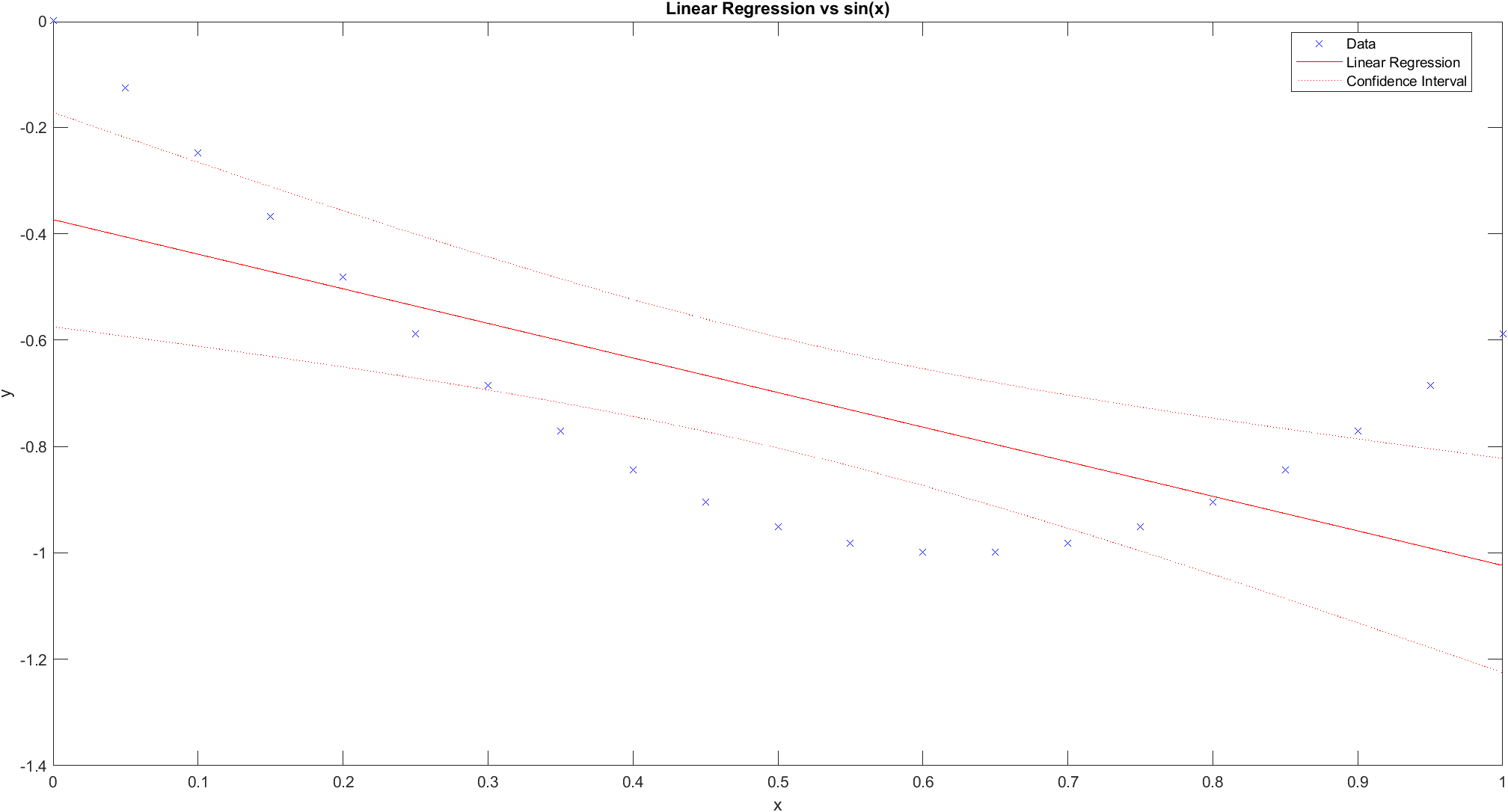
**1.1 THE PERCEPTRON AND BEYOND**

**Ex. 1.1.1** We can perform a linear regression as a one-layer perceptron with id () activation function:

, which is trained minimizing MSE.

The main link between models is linearity. Linear regression is only appropriate for modeling correlated random variables. While perceptron is appropriate for the classification of linearly separable random variables. And if random variables are linearly separable, means that data in the same class are correlated.

**Ex. 1.1.2**



**Figure 1** Linear regression vs sin(x)

Linear model can’t be used for modelling . Since linear models are appropriate only for correlated data points. This can be related to underfitting since the model is too simple for the data.

**Ex. 1.1.3**

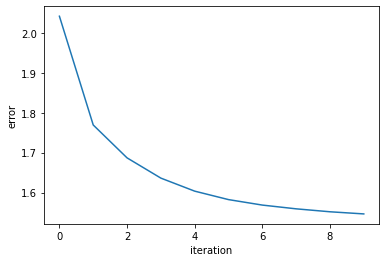
MSE of neural network trained according to the code is ., MSE of linear regression trained on the previous data is . The trained neural network much better approximates .

**1.2 BACKPROPAGATION IN FEEDFORWARD MULTI-LAYER NETWORKS**

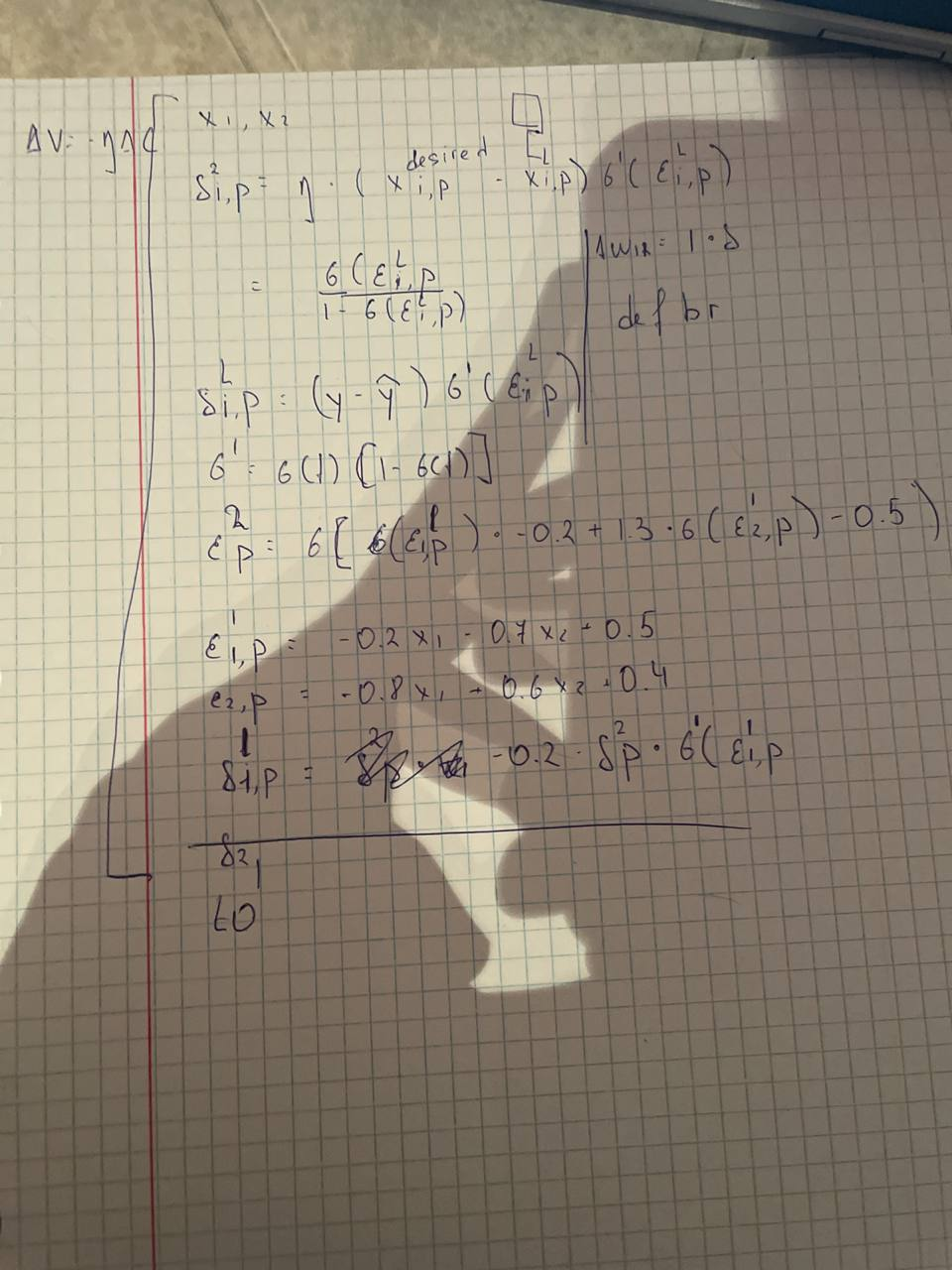
**Ex. 1.2.1** Formulas ((2.24) presented in lectures are implemented in [this](https://github.com/dsman1823/DMNN_code/blob/main/2.1.ipynb) code. After one step of offline gradient descent, the following results where obtained.

(Δw111, Δw112 , Δw121, Δw222, Δw211,Δw212) = (0.994, -0.284, -11.781, 3.369, -1.062, -1.455)

(Δb11 , Δb12, Δb21) = (0.354, -4.202, -2.646)



**Figure 1.** Error reducing after 10 iterations

****

