

Activity Task 2: Neural Networks for Classification and Identification, 2021

Set number: 6

Instruction: Write solution in Python to given problem. Task 1.1: 0.2 point, Task 1.2: 0.2 point, Task 1.3: 0.3 point, Task 2: 0.3 point. Deadline: 09.12.2021

Task 1

Create multi-layer neural network training procedure for data regression:

1. Create a new function for saving and loading trained model to file with 3 arguments. The first argument will be "file" with file name/path. The second argument will be "operation" with option 'save' for saving a network model to file and 'load' to load network model from file. The third optional argument of new function will be "model", where we send neural network model dictionary. The function will return model of neural network.
2. Update network training function to return loss function history for each iteration and each epochs. You can use any structure of loss history. Create a new function "history_plot", which plot changes of loss function based on loss history in two subplots: first plot of loss function in each epoch (total loss) and second plot of loss function in each iteration in each epoch. Add necessary description of the charts (labels, axis description, titles).
3. Add and integrate early stopping function in the training process. The early stop function allows to stop training process and return best network model, if the difference in loss function is small enough in few epochs or loss function is increasing value in few epochs. The function parameter will be "accuracy" and "patience". The accuracy control the difference between two successive epochs to stop training process. The patience parameter control number of epochs with no improvements to stop training process. The function should return the best neural network model, which means it returns the model with the lowest loss function value in the patient range. Add new arguments "accuracy" and "patience" to the training function to control early stop criterion.

Task 2

Evaluate loss function changes of neural network: 1 input layer, 2 hidden layers with 8 units and hyperbolic tangent, 1 output layer. Select the correct activation function for the output layer, loss function and other training parameters. The training parameters:

- epochs = 10000,
- learning rate = 0.001,
- early stopping: accuracy = 0.01, patience = 5 epochs

Training dataset:

```
import numpy as np
n = 20

X = np.linspace(-5, 5, n).reshape(-1, 1)
data_noise = np.random.normal(0, 0.2, n).reshape(-1, 1)
xaxis = np.sin(2 * X)
yaxis = np.cos(X)
x1, x2 = np.meshgrid(xaxis, yaxis)
y = x1 * x2 + data_noise + 7

train_x = [np.reshape(x1, (x1.size)), np.reshape(x2, (x2.size))]
train_y = np.reshape(y, (y.size))
```