

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

Set number: 21

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 classification:

1. Add new neuron activation function: leaky Relu.
2. Add new functions for data augmentation with the use of the random noise signal. The function should change inputs by adding or subtracting small random value of the input vector. The random value should be different for each sample and each element of the input vector. The random value should be generated with use standard normal distribution and scaled with additional parameter "rand_scale" (rand_scale*numpy.random.randn()). Add new argument "rand_scale" to training function, which specify a scaling factor for data augmentation.
3. Change training function to mini-batch training function. Add new argument "batch_size" to training function, which specify the batch size. If the batch size is equal to 1 than training will be using the stochastic gradient descent. If the batch size is equal to size of training dataset than training will be using the full batch training.

Task 2

Evaluate classification accuracy of neural network: 1 input layer, 2 hidden layers with 12 units and leaky Relu activation, 1 output layer. Select the correct activation function for the output layer, loss function and other training parameters. The training parameters:

- epochs = 1000,
- learning rate = 0.001,
- batch size = 8,
- rand_scale = 0.1.

Training dataset:

```
import numpy as np
n = 40

X1_1 = 2 + 4 * np.random.rand(n, 1)
X1_2 = 1 + 4 * np.random.rand(n, 1)
class1 = np.concatenate((X1_1, X1_2), axis=1)
Y1 = np.ones(n)

X0_1 = 4 + 4 * np.random.rand(n, 1)
X0_2 = 3 + 4 * np.random.rand(n, 1)
class0 = np.concatenate((X0_1, X0_2), axis=1)
Y0 = np.zeros(n)

train_x = np.concatenate((class1, class0))
train_y = np.concatenate((Y1, Y0))
```