Neuroinformatics laboratory 2

Name: Radek Bartyzal
Email: rbartyzal1@gmail.com
Code of Feed Forward NN: https://github.com/BartyzalRadek/neuroinformatics-course/blob/master/FFNN.ipynb

Polygon dataset

Network:
1st layer: 10 sigmoid neurons
2nd layer: 5 sigmoid neurons
3rd layer: 2 softmax neurons
batch_size = 10
dataset = 30x30 = 900 points

Training Algorithm: **AdaDelta**

After 1000 epochs:

After 2000 epochs:

After 3000 epochs:

After 4000 epochs:

Training Algorithm: **SGD Learning Rate = 0.01**

After 1000 epochs:

After 2000 epochs:

After 3000 epochs:

Training Algorithm: **SGD Learning Rate = 0.05**

After 1000 epochs:

After 2000 epochs:

After 3000 epochs:

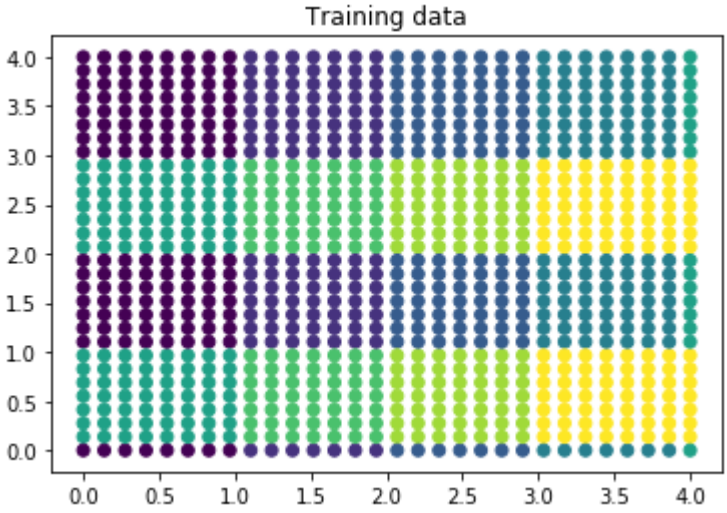
Training Algorithm: **SGD Learning Rate = 0.1**

After 1000 epochs:

After 2000 epochs:

After 3000 epochs:

Multi-class dataset



Network:
1st layer: 10 sigmoid neurons
2nd layer: 5 sigmoid neurons
3rd layer: 2 softmax neurons
batch_size = 10
dataset = 30x30 = 900 points

Training Algorithm: SGD Learning Rate = 0.1

After 500 epochs:

After 1500 epochs:

After 2500 epochs:

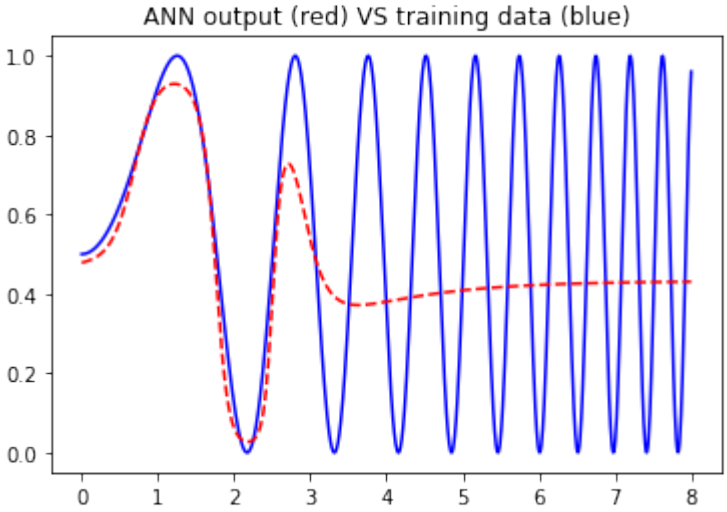
1-D function dataset

Network:

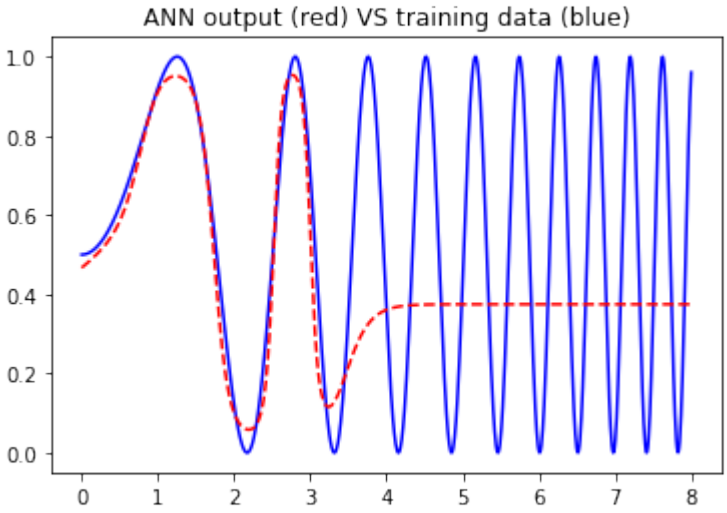
1st layer: 10 sigmoid neurons
3rd layer: 1 sigmoid neuron
batch_size = 10
dataset = 30x30 = 900 points

Training Algorithm: SGD Learning Rate = 1.0

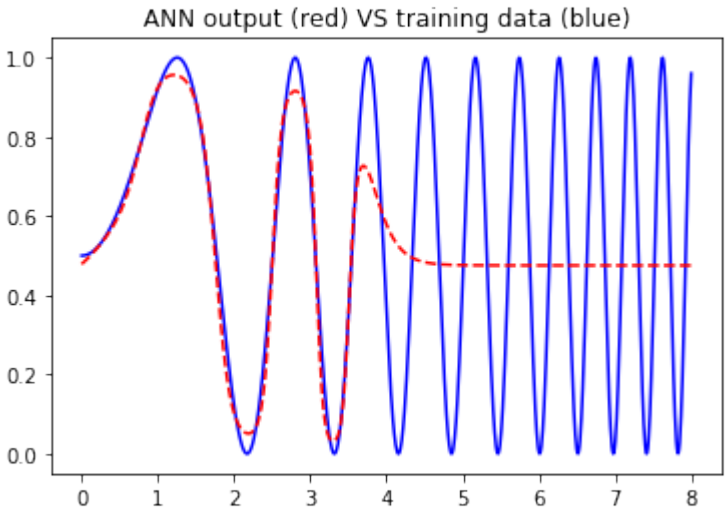
After 1000 epochs:



After 2000 epochs:

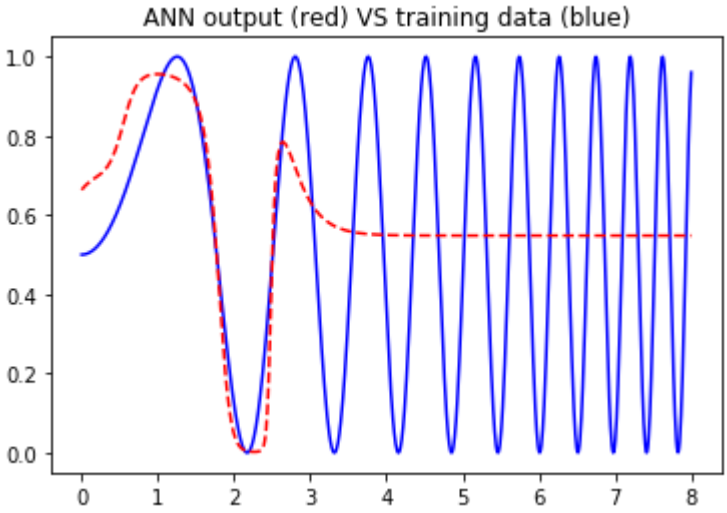


After 4000 epochs:

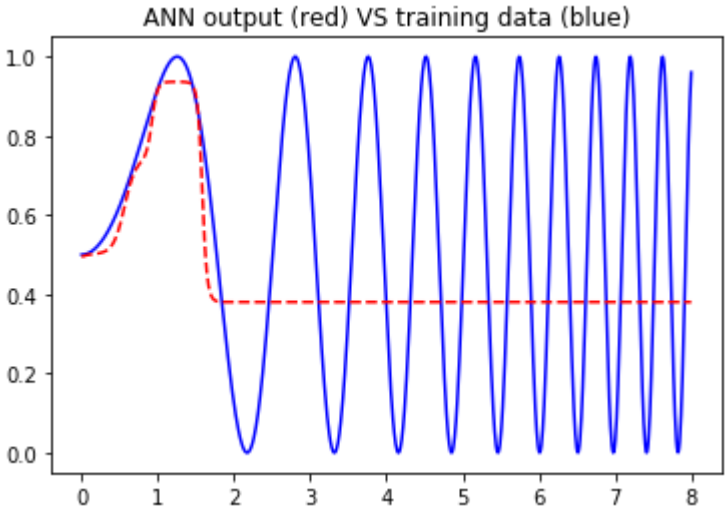


Training Algorithm: Adam

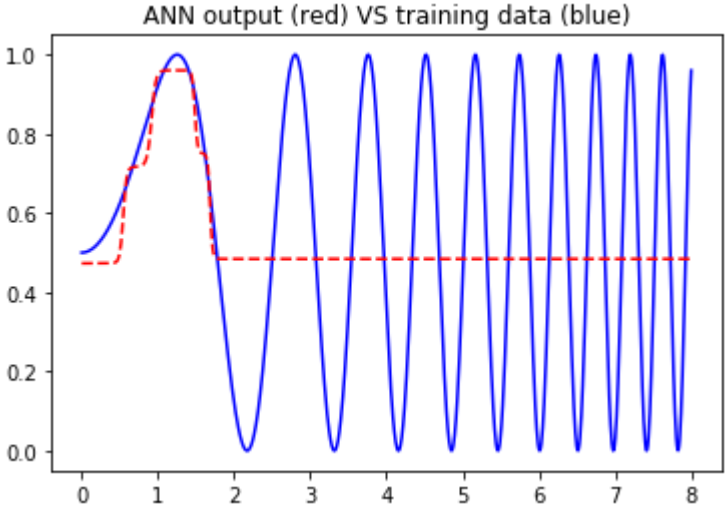
After 1000 epochs at learning rate = 0.1



After 1000 epochs at learning rate = 0.2

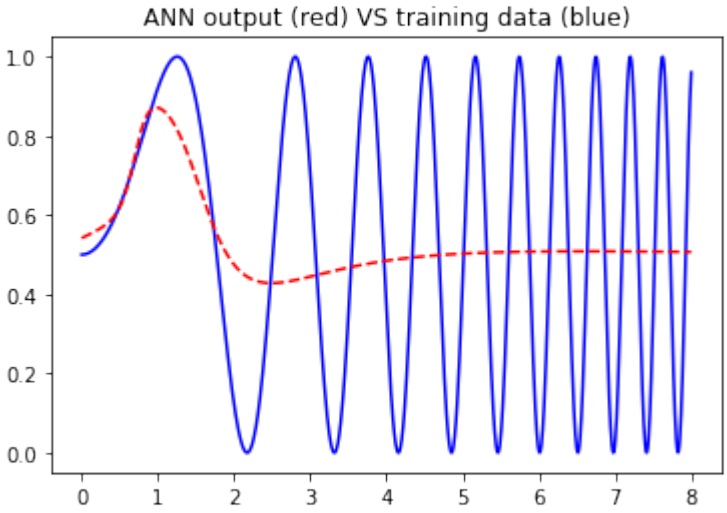


After 1000 epochs at learning rate = 0.3

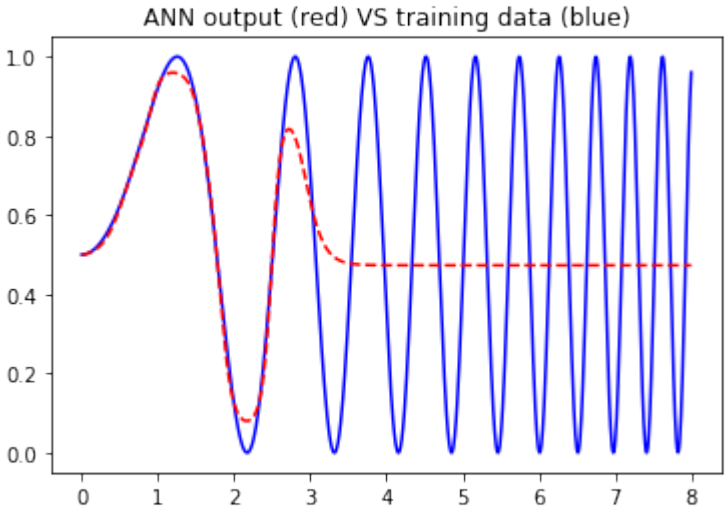


Training Algorithm: Adagrad

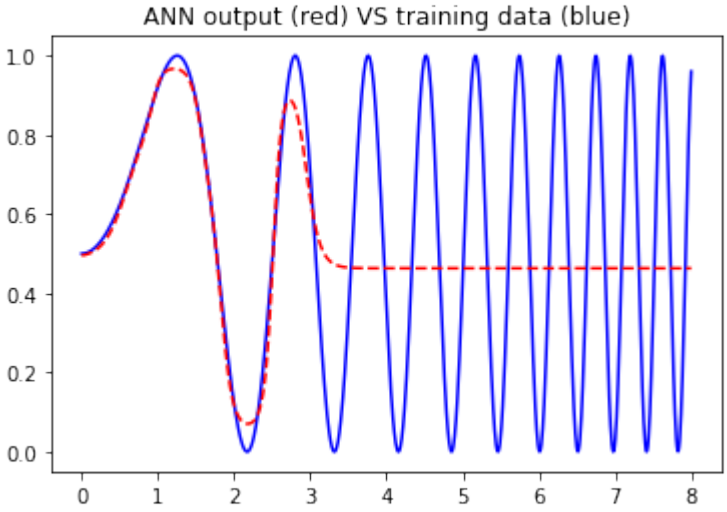
After 1000 epochs at learning rate = 0.1



After 1000 epochs at learning rate = 1.0



After 2000 epochs at learning rate = 1.0



Optimization algorithms comparison:

Computational complexity of SGD, Adam, Adagrad is approximately the same (using TensorFlow 1.3 implementation).

The best option for this simple task is SGD with a high learning rate.