Artificial Neural Networks

Prof. Dr. Sen Cheng

Nov 18, 2019

Problem Set 7: Artifical Neural Networks

Tutors: Vinita Samarasinghe (samarasinghe@ini.rub.de), Thomas Walther (thomas.walther@rub.de)

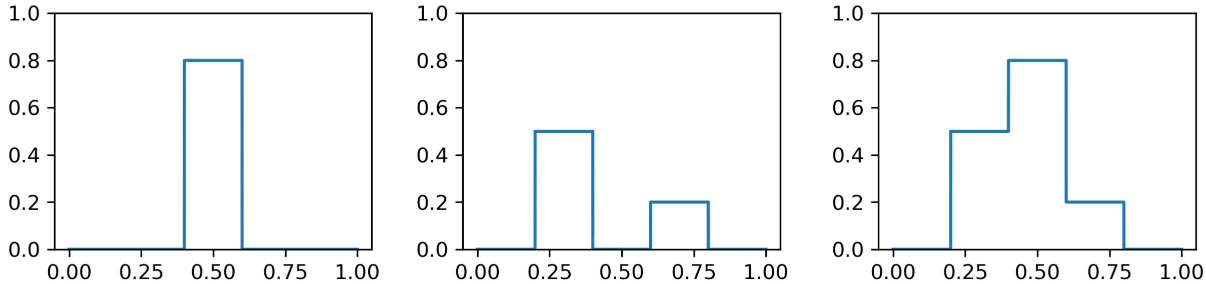
1. The universal approximation theorem states that any function can be approximated by a neural network with one hidden layer.

$$f(x) = \sum_{i=1}^{N} \nu_i \phi\left(w_i^T x + b_i\right) \tag{1}$$

   Implement this network in a Python function using only elementary programming operations. For the activation function $\phi(\cdot)$, use the sigmoid function $\sigma(\cdot)$. For the latter, use the `expit` function from `scipy.special`.

2. Using the previously implemented function $f(x)$, manually set the parameters ν_i, w_i, b_i, N in your program to replicate the output of $f(x)$ shown in the Figures 1a, 1b, and 1c.

   Useful applet that visualizes the impact of the network parameters: `http://neuralnetworksanddeeplearning.com/chap4.html`



(a) 0.4 0.6

(b) 0.2 0.4 0.6 0.8

(c)

Figure 1: Sample outputs of $f(x)$.

3. Given $g(x) = \sin(2\pi x)$ on the domain $[0;1]$.

   use 10 bars

   (a) Approximate $g(x)$ with $f(x)$ using N=10, by computing ν_i, w_i, b_i in a program. Plot the functions $g(x)$ and $f(x)$ in a single figure.

   (b) Compute the residual error $|f(x) - g(x)|$ using elementary programming operations. Repeat the approximation for several larger values of N. Plot the residual error against N.