Connectionist Neurons and Multi Layer Perceptrons

Terminology

a) As 3-layered networks with linear transfer functions are equivalent to a single connectionist neuron (p.13), non-linear transfer functions are required achieve the computational capabilities to approximate more complex functions or model more complex behaviour. Basically, we end up with only linear decision boundaries, which might not be sufficient to seperate our data points effectively.

b) The bias serves as a threshold for the activity signal (p.12). Modifying the threshold moves the decision boundary while preserving it's slope.

In figure 8 of the script (p.8) we see a working example with a bias. If we didn't have a bias, the decision boundary would pass through the origin, making it impossible to reliably distinguish apple-datapoints from orange-datapoints, regardless of it's slope.

c) Point and edge filters (Sobel filters) are specific linear weights (w) for the input vector x, used for point and edge detection in images.

d) The stochastic function has a negative slope at the inflection point and is not deterministic.

Finding Parameters of a Connectionist Neuron

a) the solutions for this task are stored in the file '2-2.ipynb'

Learning paradigms

The code for task 3 is stored in '23.py'

a) Multilayer perceptrons can have several outputs y_j, while connectionist neurons only have one output. The number of outputs defines the maximum amount of classes, that can be distinguished. To distinguish between two classes, a MLP not only needs two outputs, it also needs to apply two different activation functions. To guarantee that for a MLP with only one input x, the MLP needs at least one hidden layer.

b) Picture 1 shows 50 simulations of a MLP with ten hidden layers which uses the input-output function

$$\sum_{i=1}^{n_{hid}} w_i \, tanh(a_i(x - b_i)) \tag{1}$$

. The weights w_i were randomly varied between 0 and 1, a_i was randomly varied between 0 and 2 and b_i between -2 and 2.

c) Now, the maximum value for a_i was reduced from 2 to 0.5. This parameter amplifies the argument of the hyperbolic tangent function. Higher values of a_i lead to a more saturated curve. Since we applied smaller values in this case, the curves appear more linear/ we can see smaller extracts of the hyperbolic tangent function (fig. 2).

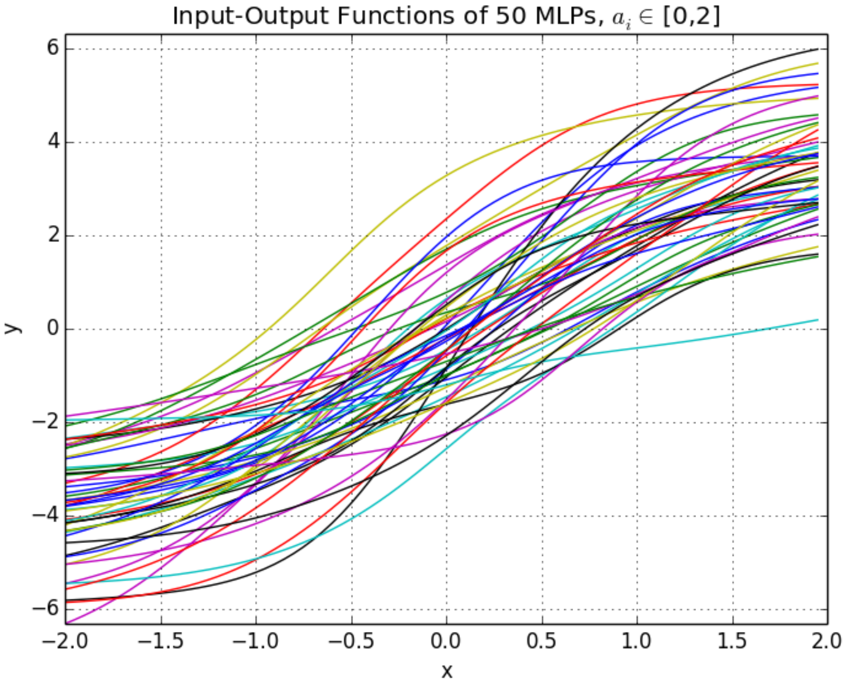

Figure 1: Simulation of 2.3 b)

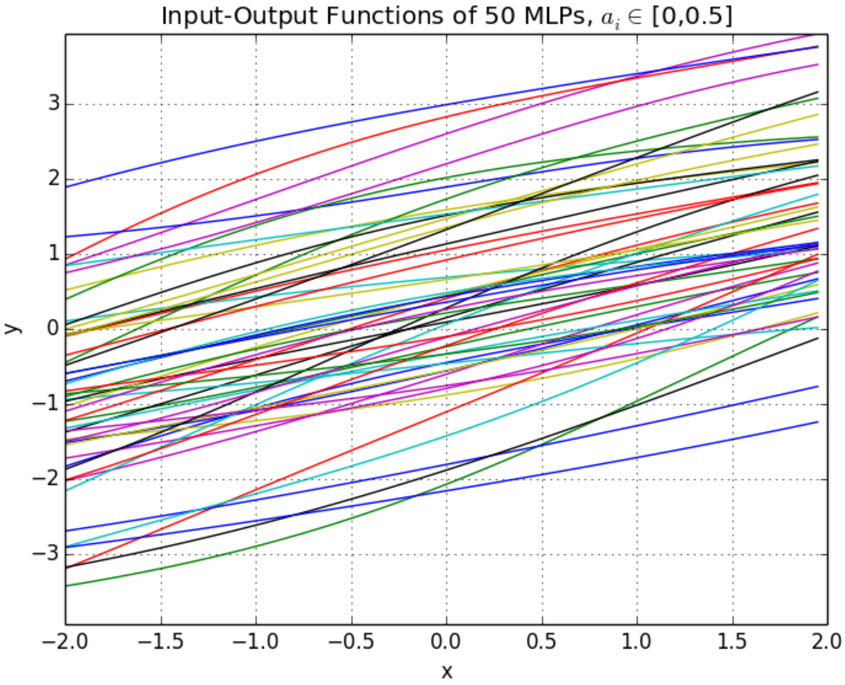

Figure 2: Simulation of 2.3 c)