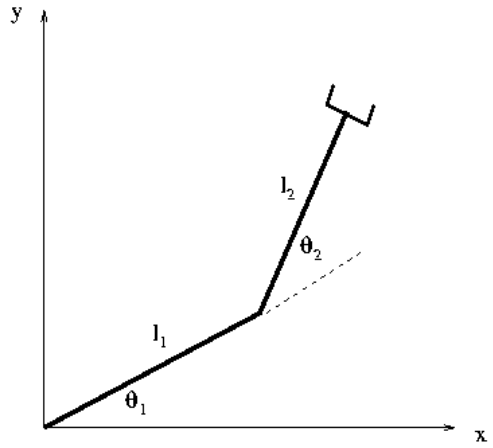
Feedforward Neural Network Exercise



Introduction

A two-link planar robot arm places its end effector at coordinates (x, y) according to the forward kinematics equations:

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

For this exercise, we will use an arm with link lengths $l_1 = 5$ and $l_2 = 3$.

Software

You may use Matlab's Neural Network toolbox or any neural network simulator of your choice. For simplicity, I recommend the BasicProp simulator (`https://basicprop.wordpress.com/`). Note that if you use the BasicProp simulator, you will need to normalize your output values so that they fall in the range of $[0, 1]$.

Questions

1. Train a multilayer feedforward neural network with 5 hidden layer units to learn the inverse kinematic transformation, i.e., given the coordinates of the end effector (x, y) as input, the network should output the corresponding joint angles (θ_1, θ_2).

2. When trained with 10 (approximately) uniformly distributed training vectors, how many epochs are required before the network converges? How does this change when the number of training vectors increases?

3. Use a set of validation data to determine how well the network generalizes to previously unseen examples. Does the addition of more hidden-layer units improve the performance of the network? What about reducing the number of these units?