

Rachel Nock

CSE 3521

Lab 3 Report

July 28, 2017

The objective of this project was to generate training data for a perceptron to learn the difference between two different shapes. The program must learn the shapes from any orientation or position. To go about this I first generated all the possible configurations of each shape. Then I built a perceptron program that inputs each shape configuration and assigns a weight based on which category the shape belongs to.

My design included a Shape class which included the image, label and name. The image is a 2D array that represents the shape with 1's where the shape is and 0's where the shape is not. The label is either a 0 or 1. A 1 for shape 1 and 0 for shape 2. The name is a char that represents which configuration the shape is in. 'o' = original configuration, 'r' = right configuration, 'l' = left configuration and 'd' = upsidedown configuration. The activation function I used was the thresholding function. The way the data was generated isn't very special because it's just hard coding all possible configurations of the 2 shapes.

I didn't use any configurations of the shape with a change in size because it would have disfigured the original shape. For example, making the U-shape smaller would have completely closed the inside gap of the shape.

When I increased the learning rate from 0.5 to 0.75, the shape 1 weights increased. The shape 2 weights increased as well. When I increased the learning rate

from 0.75 to 1, the shape 1 weights increased. The shape 2 weights increased as well.

When I decreased the learning rate from 0.5 to 0.25, the shape 1 weights decreased.

The shape 2 weights decreased as well. The optimal learning rate was 1. Shape 1 had a final weight of whatever the initial weight was and shape 2 had a weight different from the initial weight.