UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 544NA Pattern Recognition

Homework 5

Fall 2013

Assigned: Tuesday, October 8, 2013 Due: Thursday, October 17, 2013

Reading: NNPR Chapter 4

Problem 5.1

We saw that neural network architectures with *depth* 2 can approximate any (continuous) function to arbitrary precision, so why should one bother with anything deeper (aka more layers)? Let us draw a parallel to boolean circuits, in which any function can be computed as a depth-2 circuit with the sum of products representation (OR of ANDs). In this exercise, we consider the parity function and its various implementations. Given a binary vector $x = (x_1, ..., x_N)$, where $x_i \epsilon \{0, 1\}$, $parity(x)$ is defined by

$$parity(x) = \left(\sum_{i=1}^{N} x_i \right) mod\ 2$$

That is, $parity(x) = 1$ if and only if the input vector x has an odd number of ones. We further saw how each of the following boolean functions – {AND, OR, NOT, XOR} – can be constructed using a neural network. For the following settings, either *draw* or *describe your approach in words* with sufficient details for me to replicate your design. Justify your selection of the logic gates and how many of each you would use (if you need to), the number of weight parameters (including biases) in an equivalent neural network implementation, and the *depth* of the architecture – i.e., how many *layers* do you need in your equivalent neural network implementation (e.g. the *depth* of an AND gate or an OR gate is 1, whereas the *depth* of an XOR gate is 2). Your solutions need not be unique. We loosely use $O(.)$ to indicate *order of*. When we say, for example, that the number of weights is $O(N^2)$, we mean that the total number of weights is $aN^2 + bN + smaller\ terms$ for $a \neq 0$ (otherwise it would've been $O(N)$).

(a) An architecture in which (no more than) N gates are used.

(b) An architecture with depth 2.

(c) An architecture with depth $O(\log_2 N)$.

(d) An architecture with depth 3 and with $O(N^2)$ weights [hint: try to directly implement the definition given above]

(e) An architecture with $O(N)$ weights for which the *depth* is minimized [hint: relax your solution to part (d)]

(f) What are the advantages and disadvantages of a shallow architecture? And of a deep architecture? In practical settings, we would also need to *learn* these weights, so please provide a precise (and rigorous) discussion of the tradeoffs between the number of parameters, computation, and optimality.

(g) Which of these architectures would you implement, and why?

Matlab Exercises

Problem 5.2

Let us continue to use the diabetes dataset from Homework 4. Again, partition the dataset into a **training set** that contains **80 %** of the samples and a **test set** that contains the remaining **20 %**. Implement the backpropagation algorithm (**note:** you are expected to write your own functions, scripts, etc.) with the *tanh* activation function to train and test a feedforward neural network with

(a) 8 input nodes and h hidden nodes, where $h \epsilon \{2, 4, 8, 16\}$.

(b) Comment on the effect of h on classification error. Intuitively describe what is happening when the number of hidden nodes is greater than the number of input nodes.

(c) Compare the classification error, as well as the advantages and disadvantages of multiple layers, with your results from Homework 4.