CS 189 Introduction to Machine Learning
Fall 2017 DIS7

1 Backpropagation

In this discussion, we will explore the chain rule of differentiation, and provide some algorithmic motivation for the backpropagation algorithm. Those of you who have taken CS170 may recognize a particular style of algorithmic thinking that underlies the computation of gradients.

Let us begin by working with simple functions of two variables.

(a) Define the functions $f(x) = x^2$ and $g(x) = x$, and $h(x,y) = x^2 + y^2$. Compute the derivative of $\ell(x) = h(f(x), g(x))$ with respect to x.

(b) Chain rule of multiple variables: Assume that you have a function given by $f(x_1, x_2, \dots, x_n)$, and that $x_i = g_i(w)$ for a scalar variable w. How would you compute $\frac{\partial f}{\partial w}$?

The function graph of this computation is given below:

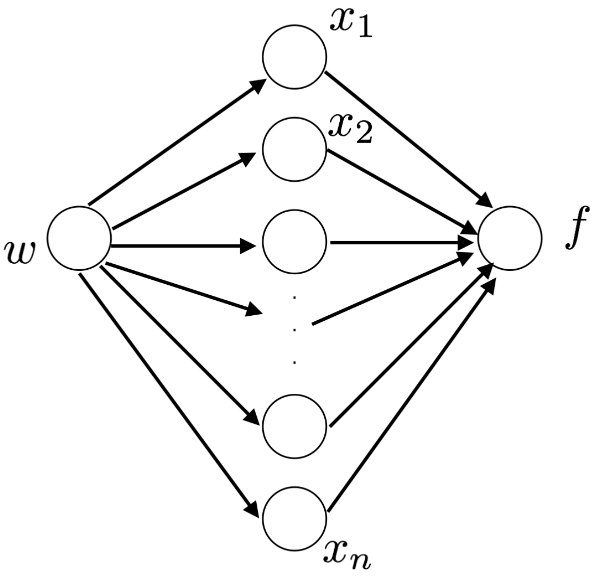

Figure 1: Example function computation graph

(c) Let $w_1, w_2, \dots, w_n \in \mathbb{R}^d$, and we refer to these variables together as W. We also have $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. Consider the function

$$f(W,x,y) = \left(y - \sum_{i=1}^{n} \phi(w_i^\top x + b_i) \right)^2 .$$

Write out the function computation graph (also sometimes referred to as a pictorial representation of the network). This is a directed graph of decomposed function computations, with the function at one end (which we will call the sink), and the variables W, x, y at the other end (which we will call the sources).

(d) What is the interpretation of taking a partial derivative of the output with respect to a particular node of this function graph? How can we use these partial derivatives to “move downhill”?

(e) Assume that the function graph is given to you as input, and you wish to determine the partial derivatives of the sink with respect to the sources. Assume that it is a directed acyclic graph (DAG). Show how one would perform this by computing the partial derivatives $\frac{\partial v_j}{\partial v_i}$ for every pair of nodes in the network where v_j is farther away from the sources than v_i. How much time (number of operations) does this take?

(f) Let us now use a different method to save computation. Starting from the output node (sink), recursively compute derivatives for nodes at distance $1, 2, \ldots$ from the sink. This is called the dynamic programming approach, and corresponds to the backpropagation algorithm. How long does this take?

(g) Discuss how gradient descent would work on the function $f(W, x, y)$ if we use backpropagation as a subroutine to compute gradients with respect to the parameters W (with x and y given).

(h) How might you make the backpropagation steps above more efficient by using vector operations?

2 Derivatives of simple functions

Compute the derivatives of the following simple functions used as non-linearities in neural networks.

(a) $\sigma(x) = \frac{1}{1+e^{-x}}$

(b) $\mathsf{ReLu}(x) = \max(x, 0)$

(c) $\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

(d) Leaky ReLu: $f(x) = \max(x, -0.1x)$