UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

COS711

ARTIFICIAL NEURAL NETWORKS

BACKPROPAGTION SOLUTION

Name and Surname	Student Number	Signature
L.M. Burgess	18015001	Liam

By submitting this assignment I confirm that I have read and am aware of the University of Pretoria's policy on academic dishonesty and plagiarism and I declare that the work submitted in this assignment is my own as delimited by the mentioned policies. I explicitly declare that no parts of this assignment have been copied from current or previous students' work or any other sources (including the internet), whether copyrighted or not. I understand that I will be subjected to disciplinary actions should it be found that the work I submit here does not comply with the said policies.

1 Introduction

Assignment 1 for COS711 to derive back-propgation weight update functions for an arbitrary set of error and activation functions.

2 Rationale

The algorithm derives the back-propagtion weight update functions for the weights from the output layer to the input layer.

3 Algorithm to follow

The terms of the following equation must be found per set of weights between layers:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial W_{ij}}$$

with net_j (b being the bias neuron):

$$net_j = \sum w_{ij} y_j + b$$

with the derivative for the weights:

$$\frac{\partial net_j}{\partial w_{ij}} = \sum y_j$$

with the derivative for the bias:

$$\frac{\partial net_j}{\partial b_j} = 1$$

3.1 Activation Functions and Derivatives

The network consists of three layers with activation functions and their derivatives:

1. Input layer:
   - $$f(x) = x$$
   - $$f'(x) = 1$$
2. Hidden layer:
   - $$f(x) = e^{-x^2} - \frac{1}{2}x$$
   - $$f'(x) = -2xe^{-x^2} - \frac{1}{2}$$
3. Output layer:
   - $$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
   - $$\begin{aligned} f'(x) &= \frac{e^x + e^{-x}}{e^x + e^{-x}} - (\frac{e^x - e^{-x}}{e^x + e^{-x}})^2 \\ &= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2} \\ &= 1 - f(x)^2 \end{aligned}$$

3.2 Objective function

The objective function and it's derivative

$$E(y_j) = \frac{1}{2}(1 - y_j t_j) + \frac{1}{2}\sqrt{(1 - y_j t_j)^2 + \epsilon}$$

$$\begin{aligned} E'(y_j) &= -\frac{t_j}{2} + 2.\frac{1}{2}. - \frac{1}{2}. - t_j.\frac{1 - y_j t_j}{\sqrt{(1 - y_j t_j)^2 + \epsilon}} \\ &= -\frac{t_j}{2} + \frac{t_j}{2}.\frac{1 - y_j t_j}{\sqrt{(1 - y_j t_j)^2 + \epsilon}} \end{aligned}$$

4 Deriving Output Layer Update Equation (Layer = 3)

1.
$$\frac{\partial net_j}{\partial w_{ij}} = y_j$$

2.
$$\begin{aligned}\frac{\partial y_j}{\partial net_j} &= 1 - f(net_j)^2 \\ &= 1 - y_j^2\end{aligned}$$

3.
$$\frac{\partial E}{\partial y_j} = -\frac{t_j}{2} - \frac{-t_j}{2}.\frac{1 - y_j t_j}{\sqrt{(1 - y_j t_j)^2 + \epsilon}}$$

4.1 Weight Update Rule

$$\begin{aligned}&\frac{\partial E}{\partial w_{ij}} \\ &= \frac{\partial E}{\partial y_j}.\frac{\partial y_j}{\partial net_j}.\frac{\partial net_j}{\partial w_{ij}} \\ &= (-\frac{t_j}{2} - \frac{-t_j}{2}.\frac{1 - y_j t_j}{\sqrt{(1 - y_j t_j)^2 + \epsilon}})(1 - y_j^2)(y_j) \\ &= \delta_j y_j\end{aligned}$$

Thus the weight update rule is:

$$\Delta W = -\eta(-\frac{t_j}{2} - \frac{-t_j}{2}.\frac{1 - y_j t_j}{\sqrt{(1 - y_j t_j)^2 + \epsilon}})(1 - y_j^2)(y_j)$$

with eta some constant term.

5 Deriving Hidden Layer Update Equation (Layer = 2)

1. $$\frac{\partial net_j}{\partial w_{ij}} = y_j$$

2. $$\frac{\partial y_j}{\partial net_j} = -2 * net(e^{-net^2}) - \frac{1}{2}$$
$$= -2 * y_j(e^{-y_j{}^2}) - \frac{1}{2}$$

3. $$\frac{\partial E}{\partial y_j} = \sum_{l \in L} \delta_l w_{jl}$$
$$= \sum_{l \in L} w_{jl}(-\frac{t_l}{2} - \frac{-t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l)$$

5.1 Weight change equation

$$\frac{\partial E}{\partial w_{ij}}$$
$$= \frac{\partial E}{\partial y_j}.\frac{\partial y_j}{\partial net_j}.\frac{\partial net_j}{\partial w_{ij}}$$
$$= (\sum_{l \in L} w_{jl}(-\frac{t_l}{2} + \frac{t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l))(-2 * y_j(e^{-y_j{}^2}) - \frac{1}{2})(y_j)$$

Thus the weight update rule is:

$$\Delta W = -\eta(\sum_{l \in L} w_{jl}(-\frac{t_l}{2} + \frac{t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l))(-2 * y_j(e^{-y_j{}^2}) - \frac{1}{2})(y_j)$$

with eta some constant term.

6 Bias Update in Hidden Layer

1.
$$\frac{\partial net_j}{\partial b_j} = 1$$

2.
$$\frac{\partial y_j}{\partial net_j} = -2 * net(e^{-net^2}) - \frac{1}{2}$$
$$= -2 * y_j(e^{-y_j{}^2}) - \frac{1}{2}$$

3.
$$\frac{\partial E}{\partial y_j} = \sum_{l \in L} \delta_l w_{jl}$$
$$= \sum_{l \in L}(-\frac{t_l}{2} - \frac{-t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l)$$

6.1 Weight change equation (for bias b, Layer = 2)

$$\frac{\partial E}{\partial b_j}$$
$$= \frac{\partial E}{\partial y_j}.\frac{\partial y_j}{\partial net_j}.\frac{\partial net_j}{\partial b_j}$$
$$= (\sum_{l \in L}(-\frac{t_l}{2} + \frac{t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l))(-2 * y_j(e^{-y_j{}^2}) - \frac{1}{2})$$

Thus the weight update rule is:

$$\Delta W_b = -\eta(\sum_{l \in L}(-\frac{t_l}{2} + \frac{t_l}{2}.\frac{1 - y_l t_l}{\sqrt{(1 - y_l t_l)^2 + \epsilon}})(1 - y_l^2)(y_l))(-2 * y_j(e^{-y_j{}^2}) - \frac{1}{2})$$

with eta some constant term.

It can thus be seen that the bias update rule differs from the normal update for the hidden layer.