Deep Neural Network - Backpropagation with ReLU

user1157751

I'm having some difficulty in deriving back propagation with ReLU, and I did some work, but I'm not sure if I'm on the right track.

ReLU についての誤差逆伝搬法を微分するのに私は少し手間取っている。多少やってみたのだが、正しい方法でできているのか不安だ。

Cost Function: $\frac{1}{2}(y-\hat{y})^2$ where y is the real value, and $\hat{y}$ is a predicted value.

Also assume that $x > 0$ always.

y を実数とし、$\hat{y}$ を予測された値として、$\frac{1}{2}(y-\hat{y})^2$ がコスト関数だ。

また、常に $x > 0$ だと前提している。

1 Layer ReLU, where the weight at the 1st layer is w_1

1 層の ReLU。その第 1 層の重みは w_1 である。

$$\frac{dC}{dw_1} = \frac{dC}{dR}\frac{dR}{dw_1}$$

$$\frac{dC}{w_1} = (y - ReLU(w_1 x))(x)$$

Neil Slater

Working definitions of ReLU function and its derivative:

ReLu 関数とその導関数の動作する定義は次のものだ：

$$ReLU(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases}$$

$$\frac{d}{dx}ReLU(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{otherwise} \end{cases}$$

The derivative is the unit step function.

この導関数は、単位階段関数だ。

This does ignore a problem at $x = 0$, where the gradient is not strictly defined, but that is not a

これは、$x = 0$ で勾配が厳密に定義されていないという問題を無視する。しかし、ニューラルネット

practical concern for neural networks.

With the above formula, the derivative at 0 is 1, but you could equally treat it as 0, or 0.5 with no real impact to neural network performance.

ワークにおいてはそれは実際上は問題ではない。

上の公式によると、0 における微分係数は 1 だ。しかし 0 や 0.5 としても、ニューラルネットワークのパフォーマンスにはどうという影響はなく同じだ。

Simplified network 単純化されたネットワーク

With those definitions, let's take a look at your example networks.

これらの定義に則り、あなたが例示したネットワークらを見ていこう。

You are running regression with cost function $C = \frac{1}{2}(y - \hat{y})^2$.

You have defined R as the output of the artificial neuron, but you have not defined an input value.

I'll add that for completeness - call it z, add some indexing by layer, and I prefer lower-case for the vectors and uppercase for matrices, so $r^{(1)}$ output of the first layer, $z^{(1)}$ for its input and $W^{(0)}$ for the weight connecting the neuron to its input x (in a larger network, that might connect to a deeper r value instead).

I have also adjusted the index number for the weight matrix – why that is will become clearer for the larger network.

NB I am ignoring having more than newuron in each layer for now.

あなたは、$C = \frac{1}{2}(y - \hat{y})^2$ を費用関数として回帰を行っている。

あなたは R を、人工的ニューロンの出力として定義した。しかしまだ入力値を定義していない。

完全にするために私はそれを追加し、z と呼ぼう。層によって添字をつけよう。また私は、ベクトルについて小文字を、行列について大文字を使いたいから、 第 1 層の出力を $r^{(1)}$ と呼び、それへの入力を $z^{(1)}$ と呼び、