Homework 5

Colin Murphy Nate Holland Greg Dehmlow

7 April 2014

Single Layer Neural Network

1. First let's start by deriving the stochastic gradient descent for the Mean squared error. At a high level the stochastic gradient descent is basically taking the weight and then adding or subtracting a small epsilon times the gradient at that point. So we start with:

$J = \frac{1}{2}\sum_{k=1}^{n_{out}}(t_k - y_k)^2$

Then at a high level we get:

$w_{new} = w_{old} - \alpha\frac{\sigma J}{\sigma w}$

$$w_{new} = w_{old} - \alpha \begin{bmatrix} \frac{\sigma J}{\sigma w_{11}} & \dots & \frac{\sigma J}{\sigma w_{1n}} \\ \dots & \frac{\sigma J}{\sigma w_{ij}} & \dots \\ \frac{\sigma J}{\sigma w_{m1}} & \dots & \frac{\sigma J}{\sigma w_{mn}} \end{bmatrix}$$

Then we know that:

$y_k = \sigma(\sum w_{jk}x_j + b_k) = \frac{1}{1+exp(-(s_k+b_k))^2}$

For notations purposes let $S_k = \sum w_{jk}x_j$. Then:

$\frac{\sigma y_k}{\sigma S_k} = \frac{-exp(-(S_k+b_k))}{(1+exp(-(S_k+b_k)))^2}$

Now since we are only dealing with one layer we get:

$\frac{\sigma J}{\sigma w_{ij}} = \frac{\sigma J}{\sigma S_j}\cdot\frac{\sigma S_j}{\sigma w_{ij}} = \delta_i X_i$

$\frac{\sigma J}{\sigma S_j} = \frac{\sigma J}{\sigma y_j}\cdot\frac{\sigma y_j}{\sigma S_j} = -(y_j - t_j)\cdot\frac{-exp(-(S_k+b_k))}{(1+exp(-(S_k+b_k)))^2}$

$\frac{\sigma J}{\sigma w_{ij}} = \frac{-exp(-(S_k+b_k))}{(1+exp(-(S_k+b_k)))^2}\cdot(y_j - t_j)\cdot x_i$

Now we select a random w to start computing on and we compute $\frac{\sigma J}{\sigma w}$ for a given (x_i, y_i) or set of data points. Then we use $\frac{\sigma J}{\sigma w}$ to perform the gradient descent update. Then if we want to include a bias term b_j we get:

$\frac{\sigma J}{\sigma b_j} = \frac{\sigma J}{\sigma y_j}\cdot\frac{\sigma y_j}{\sigma b_j} = \frac{exp(-(S_k+b_k))}{(1+exp(-(S_k+b_k)))^2}\cdot(y_j - t_j)$

2. Now let us derive the stochastic gradient descent of the cross entropy error. The equation we start out with is:

$J = -\sum_{k=1}^{n_{out}} [t_k ln(y_k) + (1 - t_k) ln(1 - y_k)]$

Again like in the previous derivation, since we are dealing with only a single layer we get:

$\frac{\sigma J}{\sigma w_{ij}} = \frac{\sigma J}{\sigma y_j} \cdot \frac{\sigma y_j}{\sigma S_j} \cdot \frac{\sigma S_j}{\sigma w_{ij}}$

We also know:

$y_k = \sigma(s_k, b_k) = \frac{1}{1+exp(-(s_k+b_k))}$

$\frac{\sigma J}{\sigma w_{ij}} = -(\frac{t_k}{y_k} - \frac{1-t_k}{1-y_k}) \cdot \frac{exp(-(s_k+b_k))}{(1+exp(-(s_k+b_k)))^2} \cdot x_i$

$\frac{\sigma J}{\sigma w_{ij}} = (\frac{y_k-t_k}{y_k(1-y_k)}) \cdot \frac{exp(-(s_k+b_k))}{(1+exp(-(s_k+b_k)))^2} \cdot x_i$

Then we use this to compute the gradient and perform the update step like on the previous page. Then for the bias we get:

$\frac{\sigma J}{\sigma b_j} = \frac{\sigma J}{\sigma y_j} \cdot \frac{\sigma y_j}{\sigma b_j} = (\frac{y_k-t_k}{y_k(1-y_k)}) \cdot \frac{exp(-(s_k+b_k))}{(1+exp(-(s_k+b_k)))^2}$

Again you compute $\frac{\sigma J}{\sigma b_j}$ for a point and then do the gradient step. For mini batches you simply average $\frac{\sigma J}{\sigma w}$ for each data point in the minibatch.

Appendix