Details on neural network derivative computations for HW2

Justin Domke

October 11, 2017

1 Intro

These are notes intended for students in machine learning 589 on how to implement a simple version of autodiff / backprop for a fully-connected network with one hidden layer. In particular, you will need to impement the boxed equations.

2 Predictor

In this section we are interested in the neural network

$$f(x) = c + V\sigma(b + Wx).$$

Let's carefully define each object:

- p is the number of inputs
- M is the number of hidden units
- K is the number of outputs. (This is the number of classes if we are doing classification.)
- $x \in \mathbb{R}^p$ is the input vector
- $W \in \mathbb{R}^{M \times p}$ is the first weight matrix
- $V \in \mathbb{R}^{K \times M}$ is the second weight matrix
- $b \in \mathbb{R}^M$ is the first "bias vector"
- $c \in \mathbb{R}^K$ is the second "bias vector"
- $\sigma(s) = \tanh(s)$ is the non-linearity, which should be thought of as acting "elementwise"..

Notice that $f(x) : \mathbb{R}^p \to \mathbb{R}^K$ is a function that maps from $\mathbb{R}^p$ to $\mathbb{R}^K$.

3 Loss function and goal of optimization

As a loss-function, we will use the multivariate logistic loss

$$L(y, f) = -f_y + \log \sum_{k=1}^{K} f_k.$$

To fit the network, we want to minimize the average loss over a dataset, namely

$$\sum_{i=1}^{N} L(y_i, f(x_i)).$$

In order to do this, we will need to compute partial derivatives of $L(y, f(x))$ with respect to the four parameters, namely W, V, b, and c. It will be convenient to remember derivatives with respect to some intermediate quantities as well.

4 Derivatives

In all these derivatives, it is convenient to use the notation

$$h = \sigma(b + Wx).$$

4.1 Derivative with respect to f.

To start with, we calculate the derivative of the loss with respect to the output, i.e. $dL/df \in \mathbb{R}^K$. This can be shown to be

$$\boxed{\frac{dL(y, f(x))}{df} = -\hat{e}_y + g(f(x))}$$

where $g(T) : \mathbb{R}^K \to \mathbb{R}^k$ is the "softmax" function $g(T)_i = \exp(T_i) \Big/ \sum_{k=1}^{K} \exp(T_k)$.

4.2 Derivative with respect to c

Next, we can immediately see that

$$\boxed{\frac{dL}{dc} = \frac{dL}{df}.}$$

4.3 Derivative with respect to V

Next, the gradient of the loss with respect to V is

$$\boxed{\frac{dL}{dV} = \frac{dL}{df} h^T.}$$

Note that $\frac{dL}{df} \in \mathbb{R}^K$ and $h \in \mathbb{R}^M$, and so $\frac{dL}{df} h^T \in \mathbb{R}^{K \times M}$ as desired.

4.4 Derivative with respect to hidden units

Recall that $h = \sigma(b + Wx)$. It is not hard to show that

$$\frac{dL}{dh} = V^T \frac{dL}{df}.$$

Then, if we define $s = b + Wx$. we can see that

$$\frac{dL}{ds} = \sigma'(s) \odot \frac{dL}{dh},$$

where $\odot$ denotes the elementwise product.

4.5 Derivative with respect to b and W

From this we can compute that

$$\begin{aligned}\frac{dL}{db} &= \frac{dL}{ds} \\ &= \sigma'(s) \odot \frac{dL}{h}\end{aligned}$$

And so, finally,

$$\boxed{\frac{dL}{db} = \sigma'(b + Wx) \odot \left(V^T \frac{dL}{df}\right)}.$$

Using similar same logic that we used above to get the derivative with respect to V we can get the derivative with respect to W as

$$\boxed{\frac{dL}{dW} = \sigma'(b + Wx) \odot (V^T \frac{dL}{df}) x^T.}$$

5 Optimization

This homework asks you to use gradient descent "with momentum".

Regular gradient descent can be written as the following iteration:

$$\theta^{r+1} \leftarrow \theta^r - \gamma \nabla R(\theta^r).$$

Where $R(\theta) = \sum_{i=1}^{N} L(y_i, f(x_i))$ is the "risk" on the dataset, and γ is a step-size.

A small modification can often make convergence much faster. We will maintain a quantity ϕ^r which is sort of a "smoothed" version of the gradient. The iteration is then

$$\begin{aligned}\phi^{r+1} &\leftarrow (1 - \alpha)\phi^r + \alpha \nabla R(\theta^r) \\ \theta^{r+1} &\leftarrow \theta^r - \gamma \phi^{r+1}.\end{aligned}$$

This is sometimes called the "heavy-ball" method, or "gradient descent with momentum". Note that all gradients you compute will be on the full dataset. (You do not do "stochastic gradient descent". (Don't worry if you don't know what that means.))