The Backpropagation Algorithm

Wen Sun

1 Problem setup

We consider a fully connected feedfoward neural network. For simplicity, we assume all layers have the same width, i.e., all layers have K many neurons and the input x is in K dimension as well. Given a training input $x \in \mathbb{R}^K$, the network is defined inductively by the following procedure:

$$z^0 = x, \tag{1}$$

$$\forall t = 0, \dots T-1 : u^{t+1} = W^{[t+1]} z^t, z^{t+1} = \sigma(u^{t+1}), \tag{2}$$

$$\hat{y} = \alpha^\top z^T \tag{3}$$

where $W^{[i]} \in \mathbb{R}^{K \times K}$, and $\sigma(\cdot)$ denotes a nonlinear transformation, e.g., ReLU, and for any vector u, $\sigma(u)$ denotes elementwise operation, i.e., it applies σ to every element in the vector u.

Our goal is to compute the derivatives of $\hat{y}$ with respect to any parameters in $W^{[i]}$ and α.

1.1 Notations

Since we will derive everything using matrix / vector format, we need to define some notations regarding matrix/vector operations. We will abuse notation a bit and denote $\partial \hat{y} / \partial z^t \in \mathbb{R}^K$ as the vector where the i-th element is $\partial \hat{y} / \partial z^t_i$, i.e.,

$$\partial \hat{y} / \partial z^t = \begin{bmatrix} \partial \hat{y} / \partial z^t_1 \\ \dots \\ \partial \hat{y} / \partial z^t_K \end{bmatrix}$$

Similarly we will denote $\partial \hat{y} / \partial u^t$ as the vector where the i-th element is $\partial \hat{y} / \partial u^t_i$. For a matrix $A \in \mathbb{R}^{d_1 \times d_2}$, we denote $\partial \hat{y} / \partial A \in \mathbb{R}^{d_1 \times d_2}$ where the (i,j)-th entry of $\partial \hat{y} / \partial A$ is $\partial \hat{y} / \partial A[i,j]$, i.e.,

$$\partial \hat{y} / \partial A = \begin{bmatrix} \partial \hat{y} / \partial A[1,1], & \dots, & \partial \hat{y} / \partial A[1,d_2] \\ \partial \hat{y} / \partial A[2,1], & \dots, & \partial \hat{y} / \partial A[2,d_2] \\ \dots & \dots & \dots \\ \partial \hat{y} / \partial A[d_1,1], & \dots, & \partial \hat{y} / \partial A[d_1,d_2] \end{bmatrix}$$

For a vector $x \in \mathbb{R}^d$, and a nonlinear transformation σ (i.e., ReLU), we denote $\sigma(x) \in \mathbb{R}^d$ as the outcome vector of element-wise application of σ on x, i.e., $\sigma(x) = [\sigma(x_1), \dots, \sigma(x_d)]^\top \in \mathbb{R}^d$. Similarly, we denote $\sigma'(x)$ as the outcome vector of element-wise application of σ' on x, i.e., $\sigma'(x) = [\sigma'(x_1), \dots, \sigma'(x_d)]^\top \in \mathbb{R}^d$. Note that when σ is ReLU (i.e., $\sigma(a) = \max\{a, 0\}$), we have $\sigma'(a)$ being defined as

$$\forall a \in \mathbb{R} : \ \sigma'(a) = \begin{cases} 1 & a \geq 0 \\ 0 & \text{else} \end{cases}$$

Finally, given two vectors $x \in \mathbb{R}^d, x' \in \mathbb{R}^d$, we denote $x \circ x'$ as the element-wise product of x, x', i.e., $x \circ x' = [x_1 x'_1, \dots, x_d x'_d]^\top \in \mathbb{R}^d$.

2 The Backpropagation Algorithm

Consider a input $x \in \mathbb{R}^d$. The algorithm consists of two passes, a forward pass and a backward pass.

2.1 Forward pass

The forward pass is described exactly in the procedure of Eq. 1, 2, 3, which starts from $z^0 = x$, and computes $u^1, z^1, u^2, z^2 \dots u^T, z^T, \hat{y}$ forward in time. The forward pass computes these vectors and stores these values for the use in the following backward pass.

2.2 Backward pass

Overall, the backward pass computes gradients $\partial \hat{y}/\partial z^t$ backward in time from T to $t = 1$. The first question we need to address is that why we care about computing $\partial \hat{y}/\partial z^t$? Aren't we just interested in computing the partial derivatives of $\hat{y}$ with respect to edge parameters in $W^{[i]}$? The following claim shows that it suffices to compute $\partial \hat{y}/\partial z^t, \forall t$, since with $\partial \hat{y}/\partial z^t, \forall t$, computing $\partial \hat{y}/\partial W^{[t]}, \forall t$ will be very easy.

Claim 1. *To compute $\partial \hat{y}/\partial W^{[t]}$ for all t, it suffices to compute $\partial \hat{y}/\partial z^t, \forall t$.*

Proof. Without loss of generality, we can just focus on a particular layer t. Assume that we are given $\partial \hat{y}/\partial z^t$. We show that we can easily compute $\partial \hat{y}/\partial W^{[t]}$. First note that $z^t = \sigma(u^t)$, hence via chain rule, we have:

$$\partial \hat{y}/\partial u_i^t = \frac{\partial \hat{y}}{\partial z_i^t}\frac{\partial z_i^t}{\partial u_i^t} = \frac{\partial \hat{y}}{\partial z_i^t}\sigma'(u_i^t)$$

Using vector format, we can write this compactly as:

$$\partial \hat{y}/\partial u^t = \frac{\partial \hat{y}}{\partial z^t} \circ \sigma'(u^t),$$

where recall $x \circ y$ represents element-wise product of two vectors, and $\sigma'(x)$ is also an element-wise operation which applies σ' to every element in the vector x.

With $\partial \hat{y}/\partial u^t$, we can compute $\partial \hat{y}/\partial W^{[t]}$. Since $u^t = W^{[t]} z^{t-1}$, we have:

$$\partial \hat{y}/\partial W^{[t]}[i,j] = \frac{\partial \hat{y}}{\partial u_i^t}\frac{\partial u_i^t}{\partial W^{[t]}[i,j]} = \frac{\partial \hat{y}}{\partial u_i^t} \cdot z_j^{t-1},$$

where the first equality comes from the fact that $W^{[t]}[i,j]$ only affects u_i^t, and the second equality comes from the fact that $\frac{\partial u_i^t}{\partial W^{[t]}[i,j]} = z_j^{t-1}$. Writing this in the matrix format, we get:

$$\partial \hat{y}/\partial W^{[t]} = \frac{\partial \hat{y}}{\partial u^t}\left(z^{t-1}\right)^\top \in \mathbb{R}^{K \times K}$$

Repeating this for every layer t concludes the proof, since we have shown that it is easy to compute $\partial \hat{y}/\partial W^{[t]}$ — the gradients that we want, using $\partial \hat{y}/\partial z^t$ for all t. □

With the above claim, now we can describe the backward pass which computes $\partial \hat{y}/\partial z^t$ backward in time t. Let us describe the base case first.

Base case Consider T-th layer. It is easy to compute $\partial \hat{y}/\partial z^T$. Since $\hat{y} = \sum_{i=1}^K \alpha_i z_i^T$, we have $\partial \hat{y}/\partial z^T = \alpha \in \mathbb{R}^K$.

Induction step. Assume that we have computed $\partial \hat{y}/\partial z^t$. Now we are going to show that we can compute $\partial \hat{y}/\partial z^{t-1}$ using $\partial \hat{y}/\partial z^t$.

Recall that in forward pass, we compute z^{t-1}, followed by u^t, followed by z^t, i.e., $z^{t-1} \to u^t \to z^t$. So by chain rule, let's first compute $\partial \hat{y}/\partial u^t$ using $\partial \hat{y}/\partial z^t$. This step is easy and we have down that in the proof of our claim above. Using the element-wise operation that we mentioned in the proof of the claim, we have:

$$\partial \hat{y}/\partial u^t = \frac{\partial \hat{y}}{\partial z^t} \circ \sigma'(u^t) \in \mathbb{R}^K.$$

Now we can compute $\partial \hat{y}/\partial z^{t-1}$ using $\partial \hat{y}/\partial u^t$. Recall the relationship between z^{t-1} and u^t, it is $u^t = W^{[t]} z^{t-1}$. Let us denote the i-th column of $W^{[t]}$ as $W^{[t]}[:, i]$, we have $u^t = \sum_{i=1}^K W^{[t]}[:, i] z_i^{t-1}$. So via chain rule, for $\partial \hat{y}/\partial z_i^{t-1}$, we have:

$$\forall i: \quad \partial \hat{y}/\partial z_i^{t-1} = \sum_{j=1}^K \frac{\partial \hat{y}}{\partial u_j^t} \frac{\partial u_j^t}{z_i^{t-1}} = \sum_{j=1}^K \frac{\partial \hat{y}}{\partial u_j^t} W^{[t]}[j, i] = \langle \frac{\partial \hat{y}}{\partial u^t}, W^{[t]}[:, i] \rangle = \left\langle \frac{\partial \hat{y}}{\partial z^t} \circ \sigma'(u^t), W^{[t]}[:, i] \right\rangle$$

Express $\partial \hat{y}/\partial z^{t-1}$ in the vector-matrix format, we have:

$$\partial \hat{y}/\partial z^{t-1} = \left(W^{[t]} \right)^\top \left(\frac{\partial \hat{y}}{\partial z^t} \circ \sigma'(u^t) \right).$$

This completes the induction step.

Put everything together with the claim 1, this gives us the following backward procedure:

$$\begin{aligned}
&\partial \hat{y}/\partial z^T = \alpha, \\
&\forall t = T \to 1: \\
&\qquad \partial \hat{y}/\partial u^t = \frac{\partial \hat{y}}{\partial z^t} \circ \sigma'(u^t) \\
&\qquad \partial \hat{y}/\partial z^{t-1} = \left(W^{[t]} \right)^\top \frac{\partial \hat{y}}{\partial u^t}, \\
&\qquad \partial \hat{y}/\partial W^{[t]} = \frac{\partial \hat{y}}{\partial u^t} \left(z^{t-1} \right)^\top.
\end{aligned}$$

So, at the end, we get $\partial \hat{y}/\partial W^{[t]}$ for $t = 1, \dots T$.

Question 1 Convince yourself that the above forward pass and the backward pass all have computation scaling linearly with respect to the size of the graph (i.e., number of edges and number of nodes in the neural network).

Question 2 Does the above generalizes to more general feedforward fully connected NN such as the one with different width at different layers? Does the above idea generalize to any directed acyclic graph?.

Further Reading http://www.offconvex.org/2016/12/20/backprop/