Vector Derivative For Back Propagation

Yongqiang Huang

December 2019

1 Jacobian and gradient

In this article we go through the basic vector calculus we have needed *so far* for computing back propagation for neural networks. First we need to familiarize ourselves with the concept of Jacobian, which is a container that holds derivative of each output variable with respect to each input variable. To be specific, let a function be

$$y = f(x) \tag{1}$$

where $f : \mathbb{R}^N \to \mathbb{R}^M$, then the Jacobian matrix is defined as

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_N} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_M}{\partial x_1} & \frac{\partial y_M}{\partial x_2} & \cdots & \frac{\partial y_M}{\partial x_N} \end{bmatrix} \tag{2}$$

In this case, the dimension of J is $M \times N$. If $f : \mathbb{R}^N \to \mathbb{R}$, then J has dimension $1 \times N$: only one y corresponding to N x_i's. If $f : \mathbb{R} \to \mathbb{R}^M$, then J has dimension $M \times 1$. Basically

$$J_{i,j} = \frac{\partial f(x)_i}{\partial x_j} \tag{3}$$

Gradient is one level down from Jacobian. We usually consider gradient $\nabla_x f(x)$ as the derivative of a function $f : \mathbb{R}^N \to \mathbb{R}$. The gradient matches in dimension the input x. If x has dimension $N \times 1$, $\nabla_x f(x)$ has dimension $N \times 1$. If x has dimension $N \times M$, $\nabla_x f(x)$ has dimension $N \times M$. In neural networks, we define a loss function L which is a scalar, and we compute the gradient of L with respect to the weight matrices, $\frac{\partial L}{\partial W}$. Usually W is a matrix, say of dimension $M \times N$, and the dimension of the gradient would be $M \times N$.

2 $y = Wx$

Let's look at a simple neural network

$$y = Wx \tag{4}$$

for which the dimensions are

$$y : M \times 1 \tag{5}$$

$$x : N \times 1 \tag{6}$$

$$W : M \times N \tag{7}$$

Suppose some loss function L is defined which is a function of y and somehow you have computed the gradient with respect to y:

$$\nabla_y L = \frac{\partial L}{\partial y} = \delta \tag{8}$$

Note that here we use $\nabla_y L$ and $\frac{\partial L}{\partial y}$ interchangably, which may not be very rigorous. The dimension of $\frac{\partial L}{\partial y}$ should be the same as that of y: $M \times 1$. Now here are two questions: given δ, what is $\frac{\partial L}{\partial x}$ and what is $\frac{\partial L}{\partial W}$? We use chain rule to solve them. First we look at $\frac{\partial L}{\partial x}$ which is computed by

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x}, \tag{9}$$

and what is $\frac{\partial y}{\partial x}$? $\frac{\partial y}{\partial x}$ is a Jacobian matrix. To know what its elements are, we need to take a closer look at how y is calculated from x:

$$y_i = \sum_{k=1}^{N} W_{i,k} x_k \tag{10}$$

What you want to notice that row i in y, or actually y_i only has relations with row i in W, and has nothing to do with any other row in W. Thus,

$$\frac{\partial y_i}{\partial x_j} = W_{i,j} \tag{11}$$

Which written in vector form is

$$\frac{\partial y}{\partial x} = W \tag{12}$$

Notice that we always *first* look at the element of a gradient and *then* recover its vector form. The vector form is neat, but we may not need it for *every* step of the chain rule. We do need it for certain steps, i.e. for the inbound and outbound derivative for any operation. Now let's go back to the gradient with respect to x: $\frac{\partial L}{\partial x}$. We already know the shape of $\frac{\partial L}{\partial x}$, which is $N \times 1$. To determine its entirety, we really only need its elements:

$$\frac{\partial L}{\partial x_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k}\frac{\partial y_k}{\partial x_i} \tag{13}$$

Pay attention to this step. L is a function of $\{y_1, y_2, \ldots, y_M\}$, so the gradient with respect to x_i must go through every y_k. Let's get back to the formula

$$\frac{\partial L}{\partial x_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial x_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} W_{k,i} \tag{14}$$

which is an inner product between δ and the i-th column of W. Its vector form (we actually need the vector form this time) is:

$$\frac{\partial L}{\partial x} = W^\top \frac{\partial L}{\partial y} = W^\top \delta \tag{15}$$

If we look at the dimension, we get: $(N \times M) \times (M \times 1) = N \times 1$ which is compatible with $\frac{\partial L}{\partial x}$. Now let's compute $\frac{\partial L}{\partial W}$, which is a little bit more difficult:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial W} \tag{16}$$

$\frac{\partial y}{\partial W}$ is, believe it or not, a Jacobian, which has a dimension of $M \times (M \times N)$. This is the case when we do *not* want to recover its vector form: it's a tensor, and if we simply expand it, we might make mistakes easily. Actually, we don't need to expand it, we just need to know its elements: $\frac{\partial y}{\partial W_{i,j}}$. Again:

$$y_i = \sum_{j=1}^{N} W_{i,j} x_j \tag{17}$$

and therefore

$$\frac{\partial y_k}{\partial W_{i,j}} = \mathbf{1}(k = i) x_j \tag{18}$$

Thus

$$\begin{aligned} \frac{\partial L}{\partial W_{i,j}} &= \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial W_{i,j}} && (19) \\ &= \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \mathbf{1}(k = i) x_j && (20) \\ &= \frac{\partial L}{\partial y_i} x_j && (21) \end{aligned}$$

The restored vector form is

$$\frac{\partial L}{\partial W} = \delta x^\top \tag{22}$$

Let's check the dimension: $(M \times 1) \times (1 \times N) = M \times N$, compatible with W.

3 $y = xW$

We can consider the same network but formulate it a little bit differently:

$$y = xW \tag{23}$$

for which the dimensions are:

$$y : 1 \times M \tag{24}$$
$$x : 1 \times N \tag{25}$$
$$W : N \times M \tag{26}$$

All we do is transpose the input x really, and W and y accordingly. Assume, again, that you have computed $\frac{\partial L}{\partial y} = \delta$ which now has dimension $1 \times M$. Again, we want to compute $\frac{\partial L}{\partial x}$, $\frac{\partial L}{\partial W}$, and we will follow almost the same procedure as we do for x being a column vector. We need two immediate partial derivatives: $\frac{\partial y_k}{\partial x_i}$ and $\frac{\partial y_k}{\partial W_{i,j}}$. Again, we need to know how y is computed from x:

$$y_k = \sum_{i=1}^{N} x_i W_{i,k} \tag{27}$$

Thus

$$\frac{\partial y_k}{\partial x_i} = W_{i,k} \tag{28}$$

which restored to vector form is

$$\frac{\partial y}{\partial x} = W^\top \tag{29}$$

and

$$\frac{\partial L}{\partial x_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial x_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} W_{i,k} \tag{30}$$

which restored to vector form is

$$\frac{\partial L}{\partial x} = \delta W^\top \tag{31}$$

Let's check the dimension: $(1 \times M) \times (M \times N) = 1 \times N$, compatible with x.

Next we compute $\frac{\partial L}{\partial W}$. First,

$$\frac{\partial y_k}{\partial W_{i,j}} = \mathbf{1}(k = j) x_i \tag{32}$$

Then

$$\frac{\partial L}{\partial W_{i,j}} = \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial W_{i,j}} \tag{33}$$

$$= \sum_{k=1}^{M} \frac{\partial L}{\partial y_k} \mathbf{1}(k=j) x_i \tag{34}$$

$$= \frac{\partial L}{\partial y_j} x_i \tag{35}$$

which restored to vector form is

$$\frac{\partial L}{\partial W} = x^\top \delta \tag{36}$$

The dimension is $(N \times 1) \times (1 \times M) = N \times M$, which is compatible with W.

4 $\mathbf{Y = XW}$

In a realistic scenario, when x is of dimension $1 \times N$, it represents a single sample that has N dimensions. If, instead of a single sample, we now have multiple samples, then our X (upper case now) would have dimension $n \times N$, where each row represents a sample, and there are n samples in total. In this case, the output Y (also upper case) would have dimension $n * M$ accordingly.

Samples in X are independent of one another, and therefore we could view X and Y as simply collections of samples:

$$X = [x_1, x_2, \ldots, x_n]^T \tag{37}$$

$$Y = [y_1, y_2, \ldots, y_n]^T \tag{38}$$

where x_i is of shape $1 \times N$, and y_j is of shape $1 \times M$, respectively, as in the previous section.

Samples are processed independently of each other, and so are their corresponding gradients. When we obtain $\frac{\partial L}{\partial Y}$, we really are obtaining a batch of n separate derivatives, each of which having the same shape $1 \times M$, that is

$$\frac{\partial L}{\partial Y} = \left[\frac{\partial L}{\partial y_1}, \frac{\partial L}{\partial y_2}, \ldots, \frac{\partial L}{\partial y_n}\right]^\top = \Delta = [\delta_1, \delta_2, \ldots, \delta_n]^\top \tag{39}$$

We already know from the previous section that, in the one-sample case,

$$\frac{\partial L}{\partial x} = \delta W^\top, \tag{40}$$

and naturally, in the batch case, since each sample works independently, we should have

$$\frac{\partial L}{\partial x_i} = \delta_i W^\top, \tag{41}$$

If we follow the dimension of X here, which is $X = [x_1, x_2, \dots, x_n]^T$, then we should also have

$$\begin{aligned}
\frac{\partial L}{\partial X} &= \left[\frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \dots, \frac{\partial L}{\partial x_n}\right]^\top && (42)\\
&= [\delta_1 W^\top, \delta_2 W^\top, \dots, \delta_n W^\top]^\top && (43)\\
&= [\delta_1, \delta_2, \dots, \delta_n]^\top W^\top && (44)\\
&= \Delta W^\top && (45)
\end{aligned}$$

The dimension checks out: $(n \times M) \times (M \times N) = n \times N$.

$\frac{\partial L}{\partial W}$ does not extend from the previous section as naturally. It is tempting to simply take the one-sample case:

$$\frac{\partial L}{\partial W} = x^\top \delta \tag{46}$$

and extend it to the batch case:

$$\frac{\partial L}{\partial W} = X^\top \Delta, \tag{47}$$

and check the dimension which does work. However, we have not justified making that extension. What should $\frac{\partial L}{\partial W}$ be when there are multiple samples?

Since samples are processed separately, We are still justified to look at each sample, for which

$$\frac{\partial L}{\partial W_i} = x_i^\top \delta_i, \quad i = [1, 2, \dots, n] \tag{48}$$

we add a subscript i to W to indicate that it corresponds to the i-th sample. Thus, we actually have n separate derivatives of L with respect to W, each corresponding to one sample: $\{\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial W_2}, \dots, \frac{\partial L}{\partial W_n}\}$. How do we obtain the ultimate $\frac{\partial L}{\partial W}$ using $\{\frac{\partial L}{\partial W_i}\}$?

We need to define a reduction operation to be applied to $\{\frac{\partial L}{\partial W_i}\}$. Do we want to add them together, or do we want to compute their average? (I think) it depends on how we compute the loss. If we compute the loss by summing the loss from all samples, then we want to average $\{\frac{\partial L}{\partial W_i}\}$ to get $\frac{\partial L}{\partial W}$. Alternatively, if we compute loss by averaging, then we should compute $\frac{\partial L}{\partial W}$ using summation.

If we assume summation, then,

$$\begin{aligned}
\frac{\partial L}{\partial W} &= \sum_i \frac{\partial L}{\partial W_i} && (49)\\
&= \sum_i x_i^\top \delta_i && (50)\\
&= [x_1^\top, x_2^\top, \dots, x_n^\top] \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix} && (51)\\
&= X^\top \Delta && (52)
\end{aligned}$$

Surprisingly, we arrive at the same conclusion as what we get previously, but this time, our result is justified.

If we assume average, we simply need to divide the above result by n.