Backpropagation for a Linear Layer

Justin Johnson

April 19, 2017

In these notes we will explicitly derive the equations to use when backpropagating through a linear layer, using minibatches.

During the forward pass, the linear layer takes an input X of shape $N \times D$ and a weight matrix W of shape $D \times M$, and computes an output $Y = XW$ of shape $N \times M$ by computing the matrix product of the two inputs. To make things even more concrete, we will consider the case $N = 2$, $D = 2$, $M = 3$.

We can then write out the forward pass in terms of the elements of the inputs:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} \qquad W = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{pmatrix} \tag{1}$$

$$Y = XW \tag{2}$$

$$= \begin{pmatrix} x_{1,1}w_{1,1} + x_{1,2}w_{2,1} & x_{1,1}w_{1,2} + x_{1,2}w_{2,2} & x_{1,1}w_{1,3} + x_{1,2}w_{2,3} \\ x_{2,1}w_{1,1} + x_{2,2}w_{2,1} & x_{2,1}w_{1,2} + x_{2,2}w_{2,2} & x_{2,1}w_{1,3} + x_{2,2}w_{2,3} \end{pmatrix} \tag{3}$$

After the forward pass, we assume that the output will be used in other parts of the network, and will eventually be used to compute a scalar loss L.

During the backward pass through the linear layer, we assume that the derivative $\frac{\partial L}{\partial Y}$ has already been computed. For example if the linear layer is part of a linear classifier, then the matrix Y gives class scores; these scores are fed to a loss function (such as the softmax or multiclass SVM loss) which computes the scalar loss L and derivative $\frac{\partial L}{\partial Y}$ of the loss with respect to the scores.

Since L is a scalar and Y is a matrix of shape $N \times M$, the gradient $\frac{\partial L}{\partial Y}$ will be a matrix with the same shape as Y, where each element of $\frac{\partial L}{\partial Y}$ gives the derivative of the loss L with respect to one element of Y:

$$\frac{\partial L}{\partial Y} = \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \tag{4}$$

During the backward pass our goal is to use $\frac{\partial L}{\partial Y}$ in order to compute $\frac{\partial L}{\partial X}$ and $\frac{\partial L}{\partial W}$. Again, since L is a scalar we know that $\frac{\partial L}{\partial X}$ must have the same shape as X ($N \times D$) and $\frac{\partial L}{\partial W}$ must have the same shape as W ($D \times M$).

By the chain rule, we know that:

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial X} \qquad\qquad \frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial W} \tag{5}$$

The terms $\frac{\partial Y}{\partial X}$ and $\frac{\partial Y}{\partial W}$ in Equation 5 are *Jacobian matrices* containing the partial derivative of each element of Y with respect to each element of the inputs X and W.

However we do not want to form the Jacobian matrices $\frac{\partial Y}{\partial X}$ and $\frac{\partial Y}{\partial W}$ explicitly, because they will be very large. In a typical neural network we might have $N = 64$ and $M = D = 4096$; then $\frac{\partial Y}{\partial X}$ consists of $64 \cdot 4096 \cdot 64 \cdot 4096$ scalar values; this is more than 68 billion numbers; using 32-bit floating point, this Jacobian matrix will take 256 GB of memory to store. Therefore it is completely hopeless to try and explicitly store and manipulate the Jacobian matrix.

However it turns out that for most common neural network layers, we can derive expressions that compute the product $\frac{\partial Y}{\partial X}\frac{\partial L}{\partial Y}$ *without explicitly forming the Jacobian* $\frac{\partial Y}{\partial X}$. Even better, we can typically derive this expression without even computing an explicit expression for the Jacobian $\frac{\partial Y}{\partial X}$; in many cases we can work out a small case on paper and then infer the general formula.

Let's see how this works out for our specific case of $N = 2$, $D = 2$, $M = 3$. We first tackle $\frac{\partial L}{\partial X}$. Again, we know that $\frac{\partial L}{\partial X}$ must have the same shape as X:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} \implies \frac{\partial L}{\partial X} = \begin{pmatrix} \frac{\partial L}{\partial x_{1,1}} & \frac{\partial L}{\partial x_{1,2}} \\ \frac{\partial L}{\partial x_{2,1}} & \frac{\partial L}{\partial x_{2,2}} \end{pmatrix} \tag{6}$$

We can proceed one element of a time. First we will compute $\frac{\partial L}{\partial x_{1,1}}$. By the chain rule, we know that

$$\frac{\partial L}{\partial x_{1,1}} = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{\partial L}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial x_{1,1}} = \frac{\partial L}{\partial Y} \cdot \frac{\partial Y}{\partial x_{1,1}} \tag{7}$$

In the above equation L and $x_{1,1}$ are scalars so $\frac{\partial L}{\partial x_{1,1}}$ is also a scalar. If we view Y not as a matrix but as a collection of intermediate scalar variables, then we can use the chain rule to write $\frac{\partial L}{\partial x_{1,1}}$ solely in terms of scalar derivatives.

To avoid working with sums, it is convenient to collect all terms $\frac{\partial L}{\partial y_{i,j}}$ into a single matrix $\frac{\partial L}{\partial Y}$; here L is a scalar and Y is a matrix, so $\frac{\partial L}{\partial Y}$ has the same shape as Y $(N \times M)$, where each element of $\frac{\partial L}{\partial Y}$ gives the derivative of L with respect to one element of Y. We similarly collect all terms $\frac{\partial y_{i,j}}{\partial x_{1,1}}$ into a single matrix $\frac{\partial Y}{\partial x_{1,1}}$; since Y is a matrix and $x_{1,1}$ is a scalar, $\frac{\partial Y}{\partial x_{1,1}}$ is a matrix with the same shape as Y $(N \times M)$.

Since $\frac{\partial L}{\partial x_{1,1}}$ is a scalar, we know that the product of $\frac{\partial L}{\partial Y}$ and $\frac{\partial Y}{\partial x_{1,1}}$ must be a scalar; by inspecting the expression using only scalar derivatives, it is clear that in this context the product of $\frac{\partial L}{\partial Y}$ and $\frac{\partial Y}{\partial x_{1,1}}$ must be a dot product.

In the backward pass we are already given $\frac{\partial L}{\partial Y}$, so we only need to compute $\frac{\partial L}{\partial x_{1,1}}$; we can easily compute this from Equation 3:

$$\frac{\partial Y}{\partial x_{1,1}} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{8}$$

Now combining Equations 6, 7, and 8 gives:

$$\frac{\partial L}{\partial x_{1,1}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{1,1}} \tag{9}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

$$= \frac{\partial L}{\partial y_{1,1}} w_{1,1} + \frac{\partial L}{\partial y_{1,2}} w_{1,2} + \frac{\partial L}{\partial y_{1,3}} w_{1,3} \tag{11}$$

We can now repeat the process to compute the other entries of $\frac{\partial L}{\partial X}$, one element at a time:

$$\frac{\partial L}{\partial x_{1,2}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{1,2}} \tag{12}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} w_{2,1} & w_{2,2} & w_{2,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{13}$$

$$= \frac{\partial L}{\partial y_{1,1}} w_{2,1} + \frac{\partial L}{\partial y_{1,2}} w_{2,2} + \frac{\partial L}{\partial y_{1,3}} w_{2,3} \tag{14}$$

$$\frac{\partial L}{\partial x_{2,1}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{2,1}} \tag{15}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 \\ w_{1,1} & w_{1,2} & w_{1,3} \end{pmatrix} \tag{16}$$

$$= \frac{\partial L}{\partial y_{2,1}} w_{1,1} + \frac{\partial L}{\partial y_{2,2}} w_{1,2} + \frac{\partial L}{\partial y_{2,3}} w_{1,3} \tag{17}$$

$$\frac{\partial L}{\partial x_{2,2}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{2,2}} \tag{18}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 \\ w_{2,1} & w_{2,2} & w_{2,3} \end{pmatrix} \tag{19}$$

$$= \frac{\partial L}{\partial y_{2,1}} w_{2,1} + \frac{\partial L}{\partial y_{2,2}} w_{2,2} + \frac{\partial L}{\partial y_{2,3}} w_{2,3} \tag{20}$$

$$\tag{21}$$

Finally we can combine Equations 9, 14, 17, and 20 to give a single expression for $\frac{\partial L}{\partial X}$ in terms of W and $\frac{\partial L}{\partial Y}$:

$$\frac{\partial L}{\partial X} = \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} w_{1,1} + \frac{\partial L}{\partial y_{1,2}} w_{1,2} + \frac{\partial L}{\partial y_{1,3}} w_{1,3} & \frac{\partial L}{\partial y_{1,1}} w_{2,1} + \frac{\partial L}{\partial y_{1,2}} w_{2,2} + \frac{\partial L}{\partial y_{1,3}} w_{2,3} \\ \frac{\partial L}{\partial y_{2,1}} w_{1,1} + \frac{\partial L}{\partial y_{2,2}} w_{1,2} + \frac{\partial L}{\partial y_{2,3}} w_{1,3} & \frac{\partial L}{\partial y_{2,1}} w_{2,1} + \frac{\partial L}{\partial y_{2,2}} w_{2,2} + \frac{\partial L}{\partial y_{2,3}} w_{2,3} \end{pmatrix} \tag{22}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \begin{pmatrix} w_{1,1} & w_{2,1} \\ w_{1,2} & w_{2,2} \\ w_{1,3} & w_{2,3} \end{pmatrix} \tag{23}$$

$$= \boxed{\frac{\partial L}{\partial Y} W^T} \tag{24}$$

In Equation 24, recall that $\frac{\partial L}{\partial Y}$ is a matrix of shape $N \times M$ and W is a matrix of shape $D \times M$; thus $\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} W^T$ has shape $N \times D$, which is the same shape as X.

We derived Equation 24 in the specific case of $N = D = 2$,$M = 3$, but it holds for any values of N, D, and M. This equation allows us to efficiently compute $\frac{\partial L}{\partial X}$ using $\frac{\partial L}{\partial Y}$ and W, without explicitly forming the Jacobian $\frac{\partial Y}{\partial X}$.

Using the same strategy of thinking about components one at a time, you can derive a similarly simple equation to compute $\frac{\partial L}{\partial W}$ without explicitly forming the Jacobian $\frac{\partial Y}{\partial W}$:

$$\boxed{\frac{\partial L}{\partial W} = X^T \frac{\partial L}{\partial Y}} \tag{25}$$

In this equation $\frac{\partial L}{\partial W}$ must have the same shape as W ($D \times M$); on the right hand side X is a matrix of shape $N \times D$ and $\frac{\partial L}{\partial Y}$ is a matrix of shape $N \times M$, so the matrix-matrix product on the right will produce a matrix of shape $D \times M$.

This strategy of thinking one element at a time can help you to derive equations for backpropagation for a layer even when the inputs and outputs to the layer are tensors of arbitrary shape; this can be particularly valuable for example when deriving backpropagation for a convolutional layer.