Technical Report: Multidimensional, Downsampled Convolution for Autoencoders

Ian Goodfellow

August 9, 2010

Abstract

This technical report describes discrete convolution with a multidimensional kernel. Convolution implements matrix multiplication by a sparse matrix with several elements constrained to be equal to each other. To implement a convolutional autoencoder, the gradients of this operation, the transpose of this operation, and the gradients of the transpose are all needed. When using standard convolution, each of these supplementary operations can be described as a convolution on slightly modified arguments. When the output is implicitly downsampled by moving the kernel in more than one pixel at each step, we must define two new operations in order to compute all of the necessary values.

1 Definitions

Let L be our loss function, W our weights defining the kernel, d a vector of strides, H our hidden units, and V our visible units. H_{cij}indexes position c (an N-dimensional index) within feature map i for example j. V is of the same format as H. W_{cij} indexes the weight at position c within the kernel, connecting visible channel i to hidden channel j.

Convolution with downsampling is performed (assuming W is pre-flipped) by

$$H_{cij} = \sum_{k,m} W_{kmi} V_{d \circ c + k, m, j}$$

(Where $\circ$ is elementwise product)

In the following sections, I derive all of the necessary operations to use this operation in an autoencoder. You may want to skip directly to the summary of results, section 7.

2 Basic gradient

The gradient of the loss function with respect to the weights is given by

$$\frac{\partial L}{\partial W_{cij}} = \sum_{k,m,n} \frac{\partial L}{\partial H_{kmn}} \frac{\partial H_{kmn}}{\partial W_{cij}}$$

$$= \sum_{k,m,n} \frac{\partial L}{\partial H_{kmn}} \frac{\partial \sum_{p,q} W_{pqm} V_{d \circ k+p,q,n}}{\partial W_{cij}}$$

$$= \sum_{k,n} \frac{\partial L}{\partial H_{kjn}} \frac{\partial \sum_{p,q} W_{pqj} V_{d \circ k+p,q,n}}{\partial W_{cij}}$$

$$= \sum_{k,n} \frac{\partial L}{\partial H_{kjn}} V_{d \circ k+c,i,n}$$

$$\frac{\partial L}{\partial W_{cij}} = \sum_{k,m} \frac{\partial L}{\partial H_{kjm}} V_{d \circ k+c,i,m}$$

With a few dimshuffles, the gradient can be computed as a convolution provided that d is all 1s. However, if d is not 1 for any element, then we have a problem because during forward prop, the index into the output is multiplied by the stride, while during computation of the gradient, the index into the *kernel* is multiplied by the stride.

3 Transpose

We can think of strided convolution as multiplication by a matrix M. Let h be H reshaped into a vector and v be V reshaped into a vector. Then

$h = Mv$

Let $hr(c, i, j)$ be a reshaping function that maps indices in H to indices in h. Let $vr(c, i, j)$ be the same for V and v.

Then

$$h_{hr(c,i,j)} = \sum_{k,m} W_{kmi} v_{vr(d \circ c+k,m,j)}$$

Thus $M_{hr(c,i,j),vr(d \circ c+k,m,j)} = W_{kmi}$ where all the relevant indices take on appropriate values, and 0 elsewhere.

Suppose we want to calculate R, a tensor in the same shape as V, such that $R_{cij} = r_{vr(c,i,j)}$ and $r = M^T h$.

$$r_a = \sum M_{ab} h_b$$

$$r_a = \sum_{c,i,j,k,m|vr(d \circ c+k,m,j)=a} W_{kmi} H_{cij}$$

$$R_{q,m,j} = \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}$$

To sum over the correct set of values for c and k, the we will need a modulus operator or saved information from a previous iteration of a for loop, unless $d = \vec{1}$. So this is not a convolution in the large stride case.

In the case where $d = \vec{1}$, we have

$$R_{q,m,j} = \sum_p \sum_i W_{w-p,m,i} H_{q-w+p,i,j}$$

where $w = W.shape - \vec{1}$.

Changing some variable names, we get

$$R_{c,i,j} = \sum_{k,m} W_{w-k,i,m} H_{c-w+k,m,j}$$

Recall that a stride 1 convolution is given by:

$$H_{cij} = \sum_{k,m} W_{kmi} V_{c+k,m,j}$$

So our transpose may be calculated by padding $d - \vec{1}$ zeros to H (each dimension of the vector gives the number of zeros to pad to each dimension of the hidden unit tensor), flipping all the spatial dimensions of the kernel, and exchanging the input channel and output channel dimensions of the kernel.

4 New notation

I'm going to make up some new notation now, since our operation isn't really convolution (downsampling is built into the operation, we don't flip the kernel, etc). From here on out, I will write

$$H_{cij} = \sum_{k,m} W_{kmi} V_{c \circ d+k,m,j}$$

as

$$H = W @_d V$$

and

$$R_{q,m,j} = \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}$$

as

$$R = W @_d^T H$$

and

$$\frac{\partial L(H = W @_d V)}{\partial W_{cij}} = \sum_{k,m} \frac{\partial L}{\partial H_{kjm}} V_{d \circ k+c,i,m}$$

as

$$\nabla_W L(H = W @_d V) = (\nabla_H L) \#_d V$$

5 Autoencoder gradients

To make an autoencoder, we'll need to be able to compute

$$R = g_v(b_v + W_r @_d^T g_h(b_h + W_e @_d V))$$

This means we're also going to need to be able to take the gradient of $L(W @^T H)$ with respect to both W (so we know how to update the encoding weights) and H, so we'll be able to propagate gradients back to the encoding layer. Finally, when we stack the autoencoders into a convolution MLP, we'll need to be able to propagate gradients back from one layer to another, so we must also find the gradient of $L(W @ V)$ with respect to V.

5.1 Gradients of the loss applied to the transpose

5.1.1 With respect to the weights

$$R_{qmj} = \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}$$

so

$$\frac{\partial L}{\partial W_{xyz}} = \sum_{q,m,j} \frac{\partial L}{\partial R_{qmj}} \frac{\partial R_{qmj}}{\partial W_{xyz}}$$

$$= \sum_{q,m,j} \frac{\partial L}{\partial R_{qmj}} \frac{\partial \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}}{\partial W_{xyz}}$$

$$= \sum_{q,j} \frac{\partial L}{\partial R_{qyj}} \frac{\partial \sum_{c|d \circ c+x=q} W_{xyz} H_{czj}}{\partial W_{xyz}}$$

$$= \sum_{q,j} \frac{\partial L}{\partial R_{qyj}} \sum_{c|d \circ c+x=q} H_{czj}$$

$$= \sum_{c,j} \frac{\partial L}{\partial R_{d \circ c+x,y,j}} H_{czj}$$

Changing some variable names, we get

$$\frac{\partial L}{\partial W_{cij}} = \sum_{k,m} H_{kjm} \frac{\partial L}{\partial R_{d \circ k+c,i,m}}$$

Recall that the gradient of $L(W@V)$ with respect to the kernel is:

$$\frac{\partial L}{\partial W_{cij}} = \sum_{k,m} \frac{\partial L}{\partial H_{kjm}} V_{d \circ k+c,i,m}$$

This has the same form as the gradient we just derived, ie both use the new #operation. Thus we can write that the gradient of $L(W@^T H)$ with respect to the kernel is given by

$$\nabla_W L(R = W@_d^T H) = H \#_d \nabla_R L$$

5.1.2 With respect to the inputs

$$R_{qmj} = \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}$$

so

$$\frac{\partial L}{\partial H_{xyz}} = \sum_{q,m,j} \frac{\partial L}{\partial R_{qmj}} \frac{\partial R_{qmj}}{\partial H_{xyz}}$$

$$= \sum_{q,m,j} \frac{\partial L}{\partial R_{qmj}} \frac{\partial \sum_{c,k|d \circ c+k=q} \sum_i W_{kmi} H_{cij}}{\partial H_{xyz}}$$

$$= \sum_{q,m} \frac{\partial L}{\partial R_{qmz}} \frac{\partial \sum_{k|d \circ x+k=q} W_{kmy} H_{xyz}}{\partial H_{xyz}}$$

$$= \sum_{q,m} \frac{\partial L}{\partial R_{qmz}} \sum_{k|d \circ x + k = q} W_{kmy}$$

$$= \sum_{k,m} \frac{\partial L}{\partial R_{d \circ x + k,m,z}} W_{kmy}$$

Changing some variable names, we get

$$\frac{\partial L}{\partial H_{cij}} = \sum_{k,m} W_{kmi} \frac{\partial L}{\partial R_{d \circ c + k,m,j}}$$

Remember that

$$H_{cij} = \sum_{k,m} W_{kmi} V_{d \circ c + k,m,j}$$

so we can write

$$\nabla_H L(R = W @_d^T H) = W @_d \nabla_R L$$

5.2 Gradient of the loss applied to @, with respect to the inputs

The above is sufficient to make a single layer autoencoder. To stack on top of it, we also need to compute:

$$\frac{\partial L(H = W @_d V)}{\partial V_{xyz}} = \sum_{cij} \frac{\partial L}{\partial H_{cij}} \frac{\partial H_{cij}}{\partial V_{xyz}}$$

$$= \sum_{cij} \frac{\partial L}{\partial H_{cij}} \frac{\partial \sum_{k,m} W_{kmi} V_{d \circ c + k,m,j}}{\partial V_{xyz}}$$

$$= \sum_{ci} \frac{\partial L}{\partial H_{ciz}} \frac{\partial \sum_{k|d \circ c + k = x} W_{kyi} V_{xyz}}{\partial V_{xyz}}$$

$$= \sum_{ci} \frac{\partial L}{\partial H_{ciz}} \sum_{k|d \circ c + k = x} W_{kyi}$$

$$= \sum_{c,k|d \circ c + k = x} \sum_{i} \frac{\partial L}{\partial H_{ciz}} W_{kyi}$$

Changing variable names around, we get

$$\frac{\partial L(H = W @_d V)}{\partial V_{qmj}} = \sum_{c,k|d \circ c + k = q} \sum_{i} W_{kmi} \frac{\partial L}{\partial H_{cij}}$$

$$\nabla_V L(H = W @_d V) = W @_d^T \nabla_H L$$

6 The rest of the gradients

We now know enough to make a stacked autoencoder. However, there are still some gradients that may be taken and it would be nice if our ops supported all of them. The @ op's gradient can be expessed in terms of $@^T$and #, and the $@^T$op's gradient can be expressed in terms of @ and #. Thus if we add a gradient method to the # op, our ops will be infinitely differentiable for all combinations of variables.

6.1 # with respect to kernel

Let $A = B\#_d C$. Then

$$\frac{\partial L(A)}{\partial B_{xyz}} = \sum_{c,i,j} \frac{\partial L(A)}{\partial A_{cij}} \frac{\partial A_{cij}}{\partial B_{xyz}}$$

$$= \sum_{c,i,j} \frac{\partial L(A)}{\partial A_{cij}} \frac{\partial \sum_{k,m} B_{kjm} C_{d \circ k+c,i,m}}{\partial B_{xyz}}$$

$$= \sum_{c,i} \frac{\partial L(A)}{\partial A_{ciy}} \frac{\partial B_{xyz} C_{d \circ x+c,i,z}}{\partial B_{xyz}}$$

$$= \sum_{c,i} \frac{\partial L(A)}{\partial A_{ciy}} C_{d \circ x+c,i,z}$$

Renaming variables, we get

$$\frac{\partial L(A)}{\partial B_{cij}} = \sum_{k,m} \frac{\partial L(A)}{\partial A_{kmi}} C_{d \circ c+k,m,j}$$

So

$$\nabla_B L(A = B\#_d C) = (\nabla_A L)@_d C$$

6.2 # with respect to input

Let $A = B\#_d C$. Then

$$\frac{\partial L(A)}{\partial C_{xyz}} = \sum_{c,i,j} \frac{\partial L(A)}{\partial A_{cij}} \frac{\partial A_{cij}}{\partial C_{xyz}}$$

$$= \sum_{c,i,j} \frac{\partial L(A)}{\partial A_{cij}} \frac{\partial \sum_{k,m} B_{kjm} C_{d \circ k+c,i,m}}{\partial C_{xyz}}$$

$$= \sum_{c,k|d\circ k+c=x} \sum_j \frac{\partial L(A)}{\partial A_{cyj}} \frac{\partial B_{kjz} C_{xyz}}{\partial C_{xyz}}$$

$$= \sum_{c,k|d\circ k+c=x} \sum_j \frac{\partial L(A)}{\partial A_{cyj}} B_{kjz}$$

Renaming variables, we get

$$\frac{\partial L(A)}{\partial C_{qmj}} = \sum_{c,k|d\circ c+k=q} \sum_i \frac{\partial L(A)}{\partial A_{kmi}} B_{cij}$$

$$\nabla_C L(A = B\#_d C) = (\nabla_A L)@_d^T B$$

7 Summary

We have defined these operations:

Strided convolution:

$$H = W@_d V \Rightarrow H_{cij} = \sum_{k,m} W_{kmi} V_{c\circ d+k,m,j}$$

Transpose of strided convolution:

$$R = W@_d^T H \Rightarrow R_{qmj} = \sum_{c,k|d\circ c+k=q} \sum_i W_{kmi} H_{cij}$$

Gradient of strided convolution with respect to weights:

$$A = B\#_d C \Rightarrow A_{cij} = \sum_{k,m} B_{kjm} C_{d\circ k+c,i,m}$$

We have observed that, if and only if if and only if $d = \vec{1}$. $@^T$and $\#$ may both be expressed in terms of $@$ by modifying their arguments with the operations of zero padding, exchanging tensor dimensions, or mirror imaging tensor dimensions. More specifically, $@_{\vec{1}}^T$and $\#_{\vec{1}}^T$ may be expressed in this way in terms of $@_{\vec{1}}$.

We have shown that the following identities are sufficient to compute any derivative any of the three operations:

$$\nabla_W L(H = W@_d V) = (\nabla_H L)\#_d V$$

$$\nabla_V L(H = W@_d V) = W@_d^T \nabla_H L$$

$$\nabla_W L(R = W @_d^T H) = H \#_d \nabla_R L$$

$$\nabla_H L(R = W @_d^T H) = W @_d \nabla_R L$$

$$\nabla_B L(A = B \#_d C) = (\nabla_A L) @_d C$$

$$\nabla_C L(A = B \#_d C) = (\nabla_A L) @_d^T B$$