Deep Learning Class
Homework I

Yaroslava Lochman June 2019

Here is the neural network's forward pass in scalar and vector form and backward pass in scalar form, layer-by-layer. The backward pass is recommended to check out in the reverse order from the end to the beginning as it was written this way. The backward pass is done for a single example (no batch), for a batch the resulting gradient w.r.t. a specific parameter would be just a mean gradient w.r.t. this parameter inside a batch. Implementations of functions **convweight2rows** and **im2col** in a fully vector form are in the experimental part. Elementary functions like **sum**, **max**, **view**, **permute**, **stack** are the same as implemented in PyTorch. The axes / dimensions for these functions are calculated starting from 0. Matrix multiplications for tensors with more than 2 dimensions are like PyTorch's **matmul**.

Input: Tensor $\mathbf{x}$ of size $[N_{batch} \times C_{in} \times S_{in} \times S_{in}] = [64 \times 1 \times 28 \times 28]$ where 64 is the batch size, 1 is the number of input channels (1 since grayscale), 28 is the input image's size.

1. **Convolutional layer**

   Learnable parameters (520 in total):

   - weights $\mathbf{w}^{(conv)}$ of size $[C_{out} \times C_{in} \times K \times K] = [20 \times 1 \times 5 \times 5]$
   - bias $\mathbf{b}^{(conv)}$ of size $[C_{out}] = [20]$

   Output: tensor $\mathbf{z}^{(conv)}$ of size $[N_{batch} \times C_{out} \times S_{out} \times S_{out}] = [64 \times 20 \times 24 \times 24]$, here $S_{out} = S_{in} - K + 1 = 28 - 5 + 1 = 24$.

   a **Forward Pass (scalar form):**

   $$z^{(conv)}_{n,c_{out},m,l} = \sum_{i=1}^{5}\sum_{j=1}^{5} x_{n,1,m+i-1,l+j-1} w^{(conv)}_{c_{out},1,i,j} + b^{(conv)}_{c_{out}}$$

   for $n = \overline{1,64}$, $c_{out} = \overline{1,20}$, $m = \overline{1,24}$, $l = \overline{1,24}$

   b **Forward Pass (vector form):**
   $\mathbf{w}^{(conv,rows)} = \text{convweight2rows}(\mathbf{w}^{(conv)})$ of size $[C_{out} \times (C_{in} \cdot K \cdot K)] = [20 \times 25]$
   $\mathbf{x}^{(cols)} = \text{im2col}(\mathbf{x}, \text{K} = 5, \text{S} = 1)$ of size $[N_{batch} \times (C_{in} \cdot K \cdot K) \times (S_{out} \cdot S_{out})] = [64 \times 25 \times 576]$
   $\hat{\mathbf{z}}^{(conv)} = \mathbf{w}^{(conv,rows)}\mathbf{x}^{(cols)} + \mathbf{b}^{(conv)}$ of size $[N_{batch} \times C_{out} \times S_{out} \cdot S_{out}] = [64 \times 20 \times 576]$
   $\mathbf{z}^{(conv)} = \text{view}(\hat{\mathbf{z}}^{(conv)}, (64, 20, 24, 24))$ of size $[N_{batch} \times C_{out} \times S_{out} \times S_{out}] = [64 \times 20 \times 24 \times 24]$

   c **Forward Pass (vector form with parallelization on channel level):**
   $\mathbf{x}^{(rows)} = \text{permute}(\mathbf{x}^{(cols)}, \text{ dims} = (0, 2, 1))$ of size $[N_{batch} \times (S_{out} \cdot S_{out}) \times (C_{in} \cdot K \cdot K)] = [64 \times 576 \times 25]$
   For the channel c:
   A vector $\mathbf{w}_c^{(conv,rows)}$ of size $C_{in} \cdot K \cdot K = 25$ represents c^{th} flattened kernel.
   The flattened convolution results on the c^{th} channel are:
   $\hat{\mathbf{z}}_c^{(conv)} = \mathbf{x}^{(rows)}\mathbf{w}_c^{(conv,rows)} + \mathbf{b}^{(conv)}$ – a tensor of size $[N_{batch} \times S_{out} \cdot S_{out}] = [64 \times 576]$

Reshaping it results in:
$\mathbf{z}_c^{(conv)} = \text{view}(\hat{\mathbf{z}}_c^{(conv)},\ (64, 24, 24))$ of size $[N_{batch} \times S_{out} \times S_{out}] = [64 \times 24 \times 24]$

After convolving for each output channel independently we can stack the results along a new channel dimension:

$$\mathbf{z}^{(conv)} = \text{stack}(\{\mathbf{z}_c^{(conv)}\}_{c=1}^{20},\ \text{dim} = 1)$$

$\mathbf{z}^{(conv)}$ has size $[N_{batch} \times C_{out} \times S_{out} \times S_{out}] = [64 \times 20 \times 24 \times 24]$

d **Backward Pass (scalar form):**

$$\frac{\partial z^{(conv)}_{c_{out},p,s}}{\partial w^{(conv)}_{b_{out},c_{in},q,r}} = \begin{cases} x_{c_{in},p+q-1,s+r-1} & \text{if } b_{out} = c_{out} \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial z^{(conv)}_{c_{out},p,s}}{\partial b^{(conv)}_{b_{out}}} = \begin{cases} 1 & \text{if } b_{out} = c_{out} \\ 0 & \text{otherwise} \end{cases}$$

for $c_{out} = \overline{1,20},\ p = \overline{1,24},\ s = \overline{1,24},\ b_{out} = \overline{1,20},\ c_{in} = 1,\ q = \overline{1,5},\ r = \overline{1,5}$

So (here is the last step of gradient derivation):

$$\frac{\partial L}{\partial w^{(conv)}_{c_{out},c_{in},q,r}} = \sum_{p=1}^{24}\sum_{s=1}^{24} \delta^{(pool)}_{c_{out},p,s} x_{c_{in},p+q-1,s+r-1} =$$

$$= \sum_{p=1}^{24}\sum_{s=1}^{24}\sum_{i=1}^{500}\sum_{j=1}^{10} (y_j - t_j) w^{(fc2)}_{j,i} \mathbf{1}_+(z_i^{(fc1)}) w^{(fc1)}_{i,ind(c_{out},p,s)} f(c_{out}, p, s) x_{c_{in},p+q-1,s+r-1}$$

$$\frac{\partial L}{\partial b^{(conv)}_{c_{out}}} = \sum_{p=1}^{24}\sum_{s=1}^{24} \delta^{(pool)}_{c_{out},p,s} =$$

$$= \sum_{p=1}^{24}\sum_{s=1}^{24}\sum_{i=1}^{500}\sum_{j=1}^{10} (y_j - t_j) w^{(fc2)}_{j,i} \mathbf{1}_+(z_i^{(fc1)}) w^{(fc1)}_{i,ind(c_{out},p,s)} f(c_{out}, p, s)$$

2. **Max-pooling layer**

Learnable parameters: –

Output: tensor $\mathbf{z}^{(pool)}$ of size $[N_{batch} \times C_{out} \times S_{out}/2 \times S_{out}/2] = [64 \times 20 \times 12 \times 12]$

a **Forward Pass (scalar form):**

$$z^{(pool)}_{n,c_{out},i,j} = \max\left(z^{(conv)}_{n,c_{out},2i-1,2j-1}, z^{(conv)}_{n,c_{out},2i-1,2j}, z^{(conv)}_{n,c_{out},2i,2j-1}, z^{(conv)}_{n,c_{out},2i,2j}\right)$$

for $n = \overline{1,64},\ c_{out} = \overline{1,20},\ i = \overline{1,12},\ j = \overline{1,12}$

b **Forward Pass (vector form):**

$\mathbf{z}^{(conv,cols)} = \text{im2col}(\mathbf{z}^{(conv)},\ \text{K} = 2,\ \text{S} = 2)$ of size $[N_{batch} \times C_{out} \times (2 \cdot 2) \times (S_{out}/2 \cdot S_{out}/2)]$ $= [64 \times 20 \times 4 \times 144]$

$\hat{\mathbf{z}}^{(pool)} = \max(\mathbf{z}^{(conv,cols)},\ \text{axis} = 2)$ of size $[N_{batch} \times C_{out} \times S_{out}/2 \cdot S_{out}/2] = [64 \times 20 \times 144]$

$\mathbf{z}^{(pool)} = \text{reshape}(\hat{\mathbf{z}}^{(pool)})$ of size $[N_{batch} \times C_{out} \times S_{out}/2 \times S_{out}/2] = [64 \times 20 \times 12 \times 12]$

c **Backward Pass (scalar form):**

$$\frac{\partial z^{(pool)}_{d_{out},m,l}}{\partial z^{(conv)}_{c_{out},p,s}} = \begin{cases} 1 & \text{if } \{c_{out},p,s\} = \arg\max\left(z^{(conv)}_{d_{out},2m-1,2l-1}, z^{(conv)}_{d_{out},2m-1,2l}, z^{(conv)}_{d_{out},2m,2l-1}, z^{(conv)}_{d_{out},2m,2l}\right) \\ 0 & \text{otherwise (noteworthy that it is always 0 for } c_{out} \neq d_{out}) \end{cases}$$

for $d_{out} = \overline{1,20}$, $m = \overline{1,12}$, $l = \overline{1,12}$, $c_{out} = \overline{1,20}$, $p = \overline{1,24}$, $s = \overline{1,24}$

We can see that:

$$\sum_{d_{out}=1}^{20} \frac{\partial z^{(pool)}_{d_{out},m,l}}{\partial z^{(conv)}_{c_{out},p,s}} \delta^{(pool)}_{d_{out},m,l} = \frac{\partial z^{(pool)}_{c_{out},m,l}}{\partial z^{(conv)}_{c_{out},p,s}} \delta^{(pool)}_{c_{out},m,l}$$

We can also see that:

$$\sum_{m=1}^{12}\sum_{l=1}^{12} \frac{\partial z^{(pool)}_{c_{out},m,l}}{\partial z^{(conv)}_{c_{out},p,s}} \delta^{(pool)}_{c_{out},m,l} = \frac{\partial z^{(pool)}_{c_{out},(p+1)//2,(s+1)//2}}{\partial z^{(conv)}_{c_{out},p,s}} \delta^{(pool)}_{c_{out},(p+1)//2,(s+1)//2}$$

where '//' means truncated division . Let's denote:

$$f(c_{out},p,s) = \frac{\partial z^{(pool)}_{c_{out},(p+1)//2,(s+1)//2}}{\partial z^{(conv)}_{c_{out},p,s}}$$

and:

$$ind(c_{out},p,s) = (c_{out}-1)\cdot 144 + ((p+1)//2 - 1)\cdot 12 + (s+1)//2$$

then finally we have:

$$\delta^{(conv)}_{c_{out},p,s} = \frac{\partial L}{\partial z^{(conv)}_{c_{out},p,s}} = \sum_{i=1}^{500}\sum_{j=1}^{10} (y_j - t_j) w^{(fc2)}_{j,i} \mathbf{1}_+(z^{(fc1)}_i) w_{i,ind(c_{out},p,s)} f(c_{out},p,s)$$

3. **Reshape (flatten) layer**

Learnable parameters: –

Output: tensor $\mathbf{z}^{(flat)}$ of size $[N_{batch} \times D] = [64 \times 2880]$, here $D = C_{out} \cdot S_{out}/2 \cdot S_{out}/2 = 20 \cdot 12 \cdot 12 = 2880$

a **Forward Pass:**

$$z^{(flat)}_{n,j} = z^{(pool)}_{n,c_{out},m,l}\ , \quad j = (c_{out}-1)\cdot 144 + (m-1)\cdot 12 + l$$

for $n = \overline{1,64}$, $c_{out} = \overline{1,20}$, $m = \overline{1,12}$, $l = \overline{1,12}$

b **Backward Pass:**

$$\frac{\partial z^{(flat)}_k}{\partial z^{(pool)}_{d_{out},m,l}} = \begin{cases} 1 & \text{if } k = (d_{out}-1)\cdot 144 + (m-1)\cdot 12 + l \\ 0 & \text{otherwise} \end{cases}$$

for $k = \overline{1,2880}$, $d_{out} = \overline{1,20}$, $m = \overline{1,12}$, $l = \overline{1,12}$ and:

$$\delta^{(pool)}_{d_{out},m,l} = \frac{\partial L}{\partial z^{(pool)}_{d_{out},m,l}} = \sum_{i=1}^{500}\sum_{j=1}^{10} (y_j - t_j) w^{(fc2)}_{j,i} \mathbf{1}_+(z^{(fc1)}_i) w^{(fc1)}_{i,(d_{out}-1)\cdot 144+(m-1)\cdot 12+l}$$

4. **Fully-connected layer 1**

   Learnable parameters (1440500 in total):

   - weights $\mathbf{w}^{(fc1)}$ of size $[P_1 \times D] = [500 \times 2880]$
   - bias $\mathbf{b}^{(fc1)}$ of size $[P_1] = [500]$

   Output: tensor $\mathbf{z}^{(fc1)}$ of size $[N_{batch} \times P_1] = [64 \times 500]$

   a **Forward Pass (scalar form):**

$$z_{n,j}^{(fc1)} = \sum_{i=1}^{2880} w_{j,i}^{(fc1)} \cdot z_{n,i}^{(flat)} + b_j^{(fc1)}$$

   for $n = \overline{1,64}$, $j = \overline{1,500}$

   b **Forward Pass (vector form):**

$$\mathbf{z}^{(fc1)} = \mathbf{z}^{(flat)}(\mathbf{w}^{(fc1)})^\top + \mathbf{b}^{(fc1)}$$

   c **Backward Pass (scalar form):**

$$\frac{\partial z_i^{(fc1)}}{\partial z_k^{(flat)}} = w_{i,k}^{(fc1)}$$

   for $i = \overline{1,500}$, $k = \overline{1,2880}$ and:

$$\delta_k^{(flat)} = \frac{\partial L}{\partial z_k^{(flat)}} = \sum_{i=1}^{500} \sum_{j=1}^{10} (y_j - t_j) w_{j,i}^{(fc2)} \mathbf{1}_+(z_i^{(fc1)}) w_{i,k}^{(fc1)}$$

5. **ReLU activation**

   Learnable parameters: –

   Output: tensor $\mathbf{z}^{(relu)}$ of size $[N_{batch} \times P_1] = [64 \times 500]$

   a **Forward Pass (scalar form):**

$$z_{n,j}^{(relu)} = \max\left(z_{n,j}^{(fc1)}, 0\right)$$

   for $n = \overline{1,64}$, $j = \overline{1,500}$

   b **Forward Pass (vector form):**

$$\mathbf{z}^{(relu)} = \max\left(\mathbf{z}^{(fc1)}, 0\right)$$

   c **Backward Pass (scalar form):**

$$\frac{\partial z_j^{(relu)}}{\partial z_i^{(fc1)}} = \begin{cases} \mathbf{1}_+(z_i^{(fc1)}) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad \text{where} \quad \mathbf{1}_+(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

   for $j = \overline{1,500}$, $i = \overline{1,500}$ and:

$$\delta_i^{(fc1)} = \frac{\partial L}{\partial z_i^{(fc1)}} = \sum_{j=1}^{10} (y_j - t_j) w_{j,i}^{(fc2)} \mathbf{1}_+(z_i^{(fc1)})$$

6. **Fully-connected layer 2**

   Learnable parameters (5010 in total):

   - weights $\mathbf{w}^{(fc2)}$ of size $[P_2 \times P_1] = [10 \times 500]$
   - bias $\mathbf{b}^{(fc2)}$ of size $[P_2] = [10]$

   Output: tensor $\mathbf{z}^{(fc2)}$ of size $[N_{batch} \times P_2] = [64 \times 10]$

   a **Forward Pass (scalar form):**

   $$z_{n,j}^{(fc2)} = \sum_{i=1}^{500} w_{j,i}^{(fc2)} \cdot z_{n,i}^{(relu)} + b_j^{(fc2)}$$

   for $n = \overline{1, 64}$, $j = \overline{1, 10}$

   b **Forward Pass (vector form):**

   $$\mathbf{z}^{(fc2)} = \mathbf{z}^{(relu)} (\mathbf{w}^{(fc2)})^\top + \mathbf{b}^{(fc2)}$$

   c **Backward Pass (scalar form):**

   $$\frac{\partial z_j^{(fc2)}}{\partial z_i^{(relu)}} = w_{j,i}^{(fc2)}$$

   for $j = \overline{1, 10}$, $i = \overline{1, 500}$ and:

   $$\delta_i^{(relu)} = \frac{\partial L}{\partial z_i^{(relu)}} = \sum_{j=1}^{10} (y_j - t_j) \frac{\partial z_j^{(fc2)}}{\partial z_i^{(relu)}} = \sum_{j=1}^{10} (y_j - t_j) w_{j,i}^{(fc2)}$$

7. **Softmax activation**

   Learnable parameters: –

   Output: tensor $\mathbf{y}$ of size $[N_{batch} \times P_2] = [64 \times 10]$

   a **Forward Pass (scalar form):**

   $$y_{n,j} = \frac{\exp(z_{n,j}^{(fc2)})}{\sum_{i=1}^{10} \exp(z_{n,i}^{(fc2)})}$$

   for $n = \overline{1, 64}$, $j = \overline{1, 10}$

   b **Forward Pass (vector form):**

   $$\mathbf{y} = \frac{\exp(\mathbf{z}^{(fc2)})}{\text{sum}(\exp(\mathbf{z}^{(fc2)}), \ \text{axis} = 1)}$$

c **Backward Pass (scalar form):**

$$\frac{\partial y_i}{\partial z_j^{(fc2)}} = y_i \left(\delta_{ij} - y_j\right), \qquad \delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

With the Cross Entropy Loss L (here and further above):

$$\delta_j^{(fc2)} = \frac{\partial L}{\partial z_j^{(fc2)}} = y_j - t_j$$

In total $520 + 1440500 + 5010 = 1\ 446\ 030$ learnable parameters.