1. Consider loss function $L(x_1, x_2) = 2x_1 + 3x_1x_2$ with learning rate γ. We want to use momentum method, where the friction parameter is 0.9. Suppose the current velocity is given by $v = (v_1, v_2)$. The current position x is given by $x = (x_1, x_2)$.

   (a) Write the update equation for v. (write in $v \leftarrow \dots$ form)

   (b) Write the update equation for x. (write in $x \leftarrow \dots$ form)

   (c) Suppose we are at t-th iteration, and want to perform bias correction. Rewrite the update equation for x. (write in $x \leftarrow \dots$ form)

   SOL:

   (a) $v \leftarrow 0.9v + 0.1(2 + 3x_2, 3x_1) = (0.9v_1 + 0.2 + 0.3x_2, 0.9v_2 + 0.3x_1)$

   (b) $x \leftarrow x - \gamma v = (x_1, x_2) - \gamma(0.9v_1 + 0.2 + 0.3x_2, 0.9v_2 + 0.3x_1)$

   (c) $x \leftarrow (x_1, x_2) - \frac{\gamma}{1-0.9^t}(0.9v_1 + 0.2 + 0.3x_2, 0.9v_2 + 0.3x_1)$

2. Consider a convolutional layer. The input map has shape $(3, 32, 32)$. That is, there are 3 channels, and the map width and height is 32. Now the filter has width and height of 3, and the depth is 3. The number of filters is given by 8. The stride is 1 and there is no zero padding. What is the shape of the output? Write the shape as

   $$(\text{number of filters}, \text{output map width}, \text{output map height})$$

   TO BE GRADED: SOL: (8,30,30)

3. Suppose you have linear layer $y = Wx + b$ where input $x \in \mathbb{R}^{100}$, and weights $W \in \mathbb{R}^{50\times100}$ and $b \in \mathbb{R}^{50}$. How would you randomly initiaize W? Use Xavier initialization.

   SOL: TO BE GRADED: The elements of W are generated according to $N(0, \sigma^2)$ where the standard deviation $\sigma = 0.1 = 1/\sqrt{100}$.

4. Suppose a batch consisting of following two samples: each sample is a feature map of spatial dimension 2×2, with 2 channels. The first sample x_1 is given by

```
[
[ [1,2], [3,4] ],
[ [5,6], [7,8] ],
]
```

   That is, the feature map corresponding to channel 0 is $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, etc. The second sample x_2 is given by

```
[
[ [1,-1], [0,2] ],
[ [-1,0], [2,1] ],
]
```

(a) Suppose we use batch normalization to this batch. What is mean parameter μ?

(b) Suppose we use layer normalization to this batch. What is mean parameter μ?

(c) Suppose we use instance normalization to this batch. What is mean parameter μ?

SOL:

(a) The mean for batch normalization obtains one number for each channel, where the average is over batch and spatial dimension. The first channel features are $\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ for batch 0 and $\begin{bmatrix}1 & -1\\ 0 & 2\end{bmatrix}$ for batch 1. Thus for channel 0,

$$\frac{1+2+3+4+1-1+0+2}{8} = 1.5$$

Similarly for channel 1, Thus for channel 0,

$$\frac{5+6+7+8-1+0+2+1}{8} = 3.5$$

Thus $\mu = (1.5, 3.5)$

(b) **TO BE GRADED:** The layer normalization gets one number per batch, where the average is over the feature and channel dimension. Thus for batch 0,

$$\frac{1+2+3+4+5+6+7+8}{8} = 4.5$$

For the second batch

$$\frac{1-1+0+2-1+0+2+1}{8} = 0.5$$

Thus $\mu = (4.5, 0.5)$

(c) Instance normalization, we average over feature dimension, leaving one average value per channel, and the per-channel averages are kept for each batch. Thus for batch 0,

$$(\frac{1+2+3+4}{4}, \frac{5+6+7+8}{4}) = (2.5, 6.5)$$

and

$$(\frac{1-1+0+2}{4}, \frac{-1+0+2+1}{4}) = (0.5, 0.5)$$

Thus $\mu = \begin{bmatrix}2.5 & 6.5\\ 0.5 & 0.5\end{bmatrix}$ (we assume μ has dimension $(N, C, 1, 1)$ where N is batch size and C is number of channels, and the rest of 1-dimensions in the shape are squeezed out, e.g., see `torch.squeeze`).

5. Suppose the input to a convolutional layer has height and width of 227 elements with the number of channels is given by 3. The layer has 64 filters with kernel size 11×11 with stride 4 and padding 2.

(a) Find the shape of the output in the form (channels, width, height).

(b) Find the number of learnable parameters (including bias) of this layer.

(c) Find the FLOP associated with this layer. Assume n-dimensional inner product uses n FLOPs.

(d) For the same input, suppose we have used 1) depthwise convolution, i.e., the convolutional filter has the same kernel size, stride and padding, however the convolution is done in depth (channel)-wise manner, 2) followed by 1-by-1 convolution to yield the output of the same shape as the previous convolutional layer. Find the total number of learnable parameters (including bias) of this layer with two-step operations.

(e) Find the total FLOP associated with this layer with two-step operations.

SOL:

(a) The width and height of the output is given by

$$\frac{N+2P-F}{S}+1=(227+4-11)/4+1=56$$

The output shape is (64, 56, 56).

(b) Convolutional filter has shape (3,11,11), and there are 64 filters. Thus (11*11*3+1)*64 = 23,296

(c) For each output, 11*11*3 FLOPs are incurred. Since there are 64*56*56 output feature values, thus 11*11*3*64*56*56=72.8M FLOPs

(d) The convolutional filter for the depthwise convolution has shape (1,11,11), and there are 3 filters, one for each input channel. Thus 3*(11*11+1) = 366. Next, 1×1 convolutional filter has shape (3,1,1), and there are 64 filters overall. Thus 64*(3*1*1+1)=256. Thus the total number of paratemters is 366+256=622

(e) The output from depthwise convolution is (3,56,56). For each output, 11-by-11 convolution filtering is used. Thus (11*11*3*56*56) = 1.1M FLOPs. The output from 1-by-1 convolution is (64,56,56), for each output 1-by-1-by-3 convolution filtering is used. Thus (64*56*56*1*1*3)= 0.6M FLOPs. Thus the total FLOP count is approximately 1.7M FLOPs

6. Consider RNN

$$h_t=\sigma(wh_{t-1})$$

where σ is a differentiable nonlinear function. Let $a=\frac{dL}{dh_t}$ where L is the loss function. Express $\frac{dL}{dh_{t-1}}$ using a, w and h_{t-1}. Assume all variables are scalar.

SOL:

$$\frac{dL}{dh_{t-1}}=\frac{dL}{dh_t}\frac{dh_t}{dh_{t-1}}=aw\sigma'(wh_{t-1})$$

7. Compute scaled dot-product attention with Q, K, V such that

$$Q=\sqrt{2}\begin{bmatrix}0 & \log 2\\ \log 2 & \log 3\end{bmatrix}, K=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, V=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

SOL:

$$\text{softmax}\left(\frac{QK^T}{\sqrt{2}}\right)V=\begin{bmatrix}2/3 & 1/3\\ 3/5 & 2/5\end{bmatrix}$$