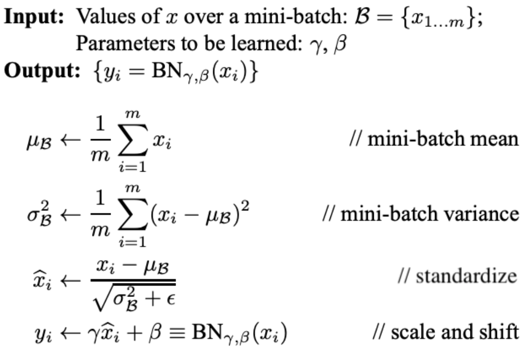
EECS 182 Deep Neural Networks
Fall 2023 Anant Sahai

Homework 3

This homework is due on Sunday, Sep 17, 2023, at 10:59PM.

1. Normalization Layers

Recall the pseudocode for a batchnorm layer (with learnable scale and shift) in a neural network:



```
Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
       Parameters to be learned: $\gamma$, $\beta$
Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$    // mini-batch mean
$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\mathcal{B})^2$    // mini-batch variance
$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$    // standardize
$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i)$    // scale and shift
```

(a) If our input data (1-dimensional) to batchnorm follows roughly the distribution on the left:

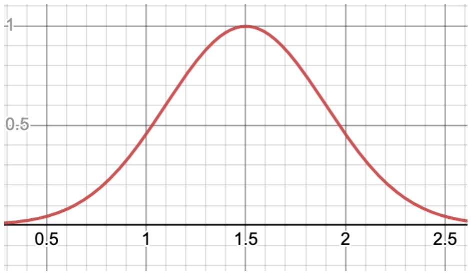


Figure 1: Gaussian with mean $\mu = 1.5$, variance $\sigma^2 = 0.16$

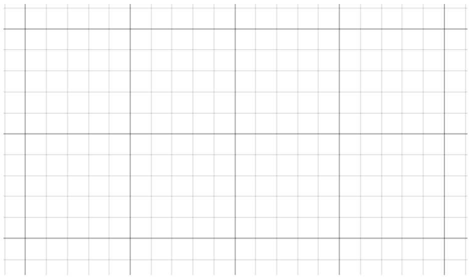


Figure 2: Blank grid for your answer

What does our data distribution look like after batch normalization with $\beta = 3$ and $\gamma = 1$ parameters? Draw your answer on the blank grid above, give a scale to the horizontal axis, and label β. You can assume that the batch-size is very large.

(Note: You do not have to give a scale to the vertical axis.)

(b) Say our input data (now 2-dimensional) to the batchnorm layer follows a Gaussian distribution. The mean and contours (level sets) of points that are 1 and 2 stdev away from the mean are shown below. **On the same graph, draw what the mean, 1-SD, and 2-SD contours would look like after batchnorm without any shifting/scaling (i.e. $\beta = 0$ and $\gamma = 1$).** You can assume that the batch-size is very large.

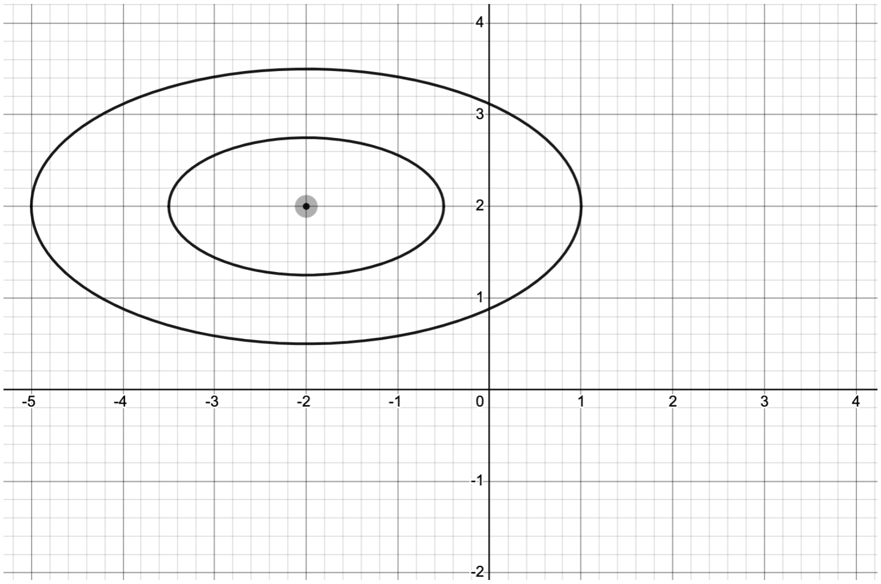


Figure 3: Draw your answer on the grid

2. Understanding Convolution as Finite Impulse Response Filter

For the discrete time signal, the output of linear time invariant system is defined as:

$$y[n] = x[n] * h[n] = \sum_{i=-\infty}^{\infty} x[n-i] \cdot h[i] = \sum_{i=-\infty}^{\infty} x[i] \cdot h[n-i] \tag{1}$$

where x is the input signal, h is impulse response (also referred to as the filter). Please note that the convolution operations is to 'flip and drag'. But for neural networks, we simply implement the convolutional layer without flipping and such operation is called correlation. Interestingly, in CNN those two operations are equivalent because filter weights are initialized and updated. Even though you implement 'true' convolution, you just ended up with getting the flipped kernel. **In this question, we will follow the definition in 1**.

Now let's consider rectangular signal with the length of L (sometimes also called the "rect" for short, or, alternatively, the "boxcar" signal). This signal is defined as:

$$x(n) = \begin{cases} 1 & n = 0, 1, 2, ..., L-1 \\ 0 & \text{otherwise} \end{cases}$$

Here's an example plot for $L = 7$, with time indices shown from -2 to 8 (so some implicit zeros are shown):

(a) The impulse response is define as:

$$h(n) = (\frac{1}{2})^n u(n) = \begin{cases} (\frac{1}{2})^n & n = 0, 1, 2, ... \\ 0 & \text{otherwise} \end{cases}$$

Compute and plot the convolution of $x(n)$ and $h(n)$. For illustrative purposes, your plot should start at -6 and end at +12.

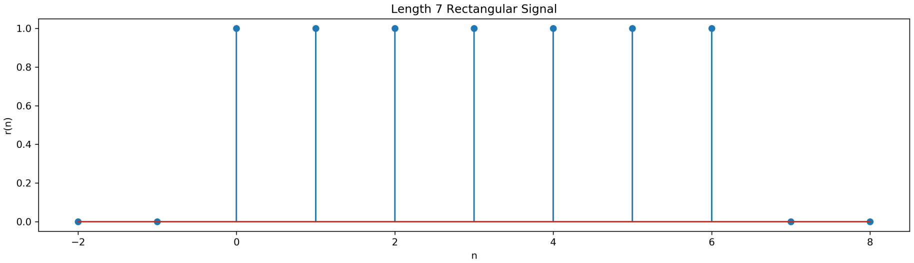


Figure 4: The rectangular signal with the length of 7

(b) Now let's shift $x(n)$ by N, i.e. $x_2(n) = x(n-N)$. Let's put $N = 5$ **Then, compute** $y_2(n) = h(n) * x_2(n)$**. Which property of the convolution can you find?**

Now, let's extend 1D to 2D. The example of 2D signal is the image. The operation of 2D convolution is defined as follows:

$$y[m,n] = x[m,n] * h[m,n] = \sum_{i,j=-\infty}^{\infty} x[m-i,n-j] \cdot h[i,j] = \sum_{i,j=-\infty}^{\infty} x[i,j] \cdot h[m-i,n-j] \tag{2}$$

, where x is input signal, h is FIR filter and y is the output signal.

(c) 2D matrices, x and h are given like below:

$$x = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix} \tag{3}$$

$$h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

Then, evaluate y**. Assume that there is no pad and stride is 1.**

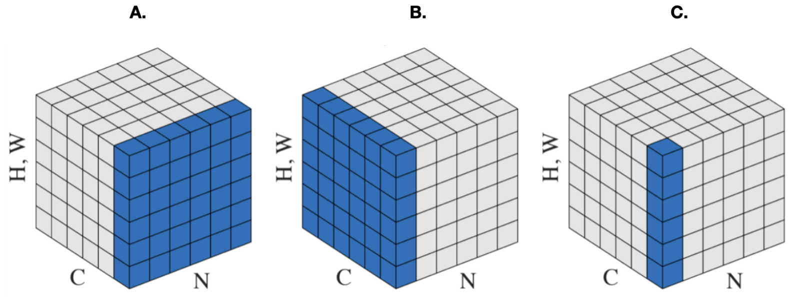
(d) Now let's consider striding and padding. Evaluate y for following cases:

i. stride, pad = 1, 1

ii. stride, pad = 2, 1

3. Normalization

(a) Consider the following digram where the shaded blocks are the entries participating in one normalization step for a CNN-type architecture. N represents the mini-batch, H, W represent the different pixels of the "image" at this layer, and C represents different channels.



- **Which one denotes the process of batch normalization?** Please use ■ for your selections.

  □ A □ B □ C

- **Which one denotes layer normalization?** Please use ■ for your selections.

  □ A □ B □ C

(b) Consider a simplified BN where we do not divide by the standard deviation of the data batch. Instead, we just de-mean our data batch before applying the scaling factor γ and shifting factor β. For simplicity, consider scalar data in an n-sized batch: $[x_1, x_2, \ldots, x_n]$. Specifically, we let $\hat{x}_i = x_i - \mu$ where μ is the average $\frac{1}{n}\sum_{j=1}^{n} x_j$ across the batch and output $[y_1, y_2, \ldots, y_n]$ where $y_i = \gamma\hat{x}_i + \beta$ to the next layer. Assume we have a final loss L somewhere downstream. **Calculate $\frac{\partial L}{\partial x_i}$ in terms of $\frac{\partial L}{\partial y_j}$ for $j = 1, \ldots, n$ as well as γ and β as needed.**

Numerically, **what is $\frac{\partial L}{\partial x_1}$ when $n = 1$ and our input batch just consists of $[x_1]$ with an output batch of $[y_1]$?** (Your answer should be a real number. No need to justify.)

What happens when $n \to \infty$? (Feel free to assume here that all relevant quantities are bounded.)

4. Optimizers

Algorithm 1 SGD with Momentum

```
Given $\eta = 0.001, \beta_1 = 0.9$
Initialize:
    time step $t \leftarrow 0$
    parameter $\theta_{t=0} \in \mathbb{R}^n$
Repeat
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla f_t(\theta_{t-1})$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $\theta_t \leftarrow \theta_{t-1} - \eta m_t$
Until the stopping condition is met
```

Algorithm 2 Adam Optimizer (without bias correction)

```
Given $\eta = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$
Initialize time step $t \leftarrow 0$, parameter $\theta_{t=0} \in \mathbb{R}^n$, $m_{t=0} \leftarrow 0, v_{t=0} \leftarrow 0$
Repeat
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla f_t(\theta_{t-1})$
    $m_t \leftarrow$ ____(A)____
    $v_t \leftarrow$ ____(B)____
    $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \frac{m_t}{\sqrt{v_t}}$
Until the stopping condition is met
```

(a) Complete part **(A)** and **(B)** in the pseudocode of Adam.

(b) This question asks you to establish the relationship between

- **L2 regularization** for vector-valued weights θ refers to adding a squared Euclidean norm of the weights to the loss function itself:

$$f_t^{reg} = f_t(\theta) + \frac{\lambda}{2}||\theta||_2^2$$

- **Weight decay** refers to explicitly introducing a scalar γ in the weight updates assuming loss f:

$$\theta_{t+1} = (1-\gamma)\theta_t - \eta\nabla f(\theta_t)$$

where $\gamma = 0$ would correspond to regular SGD since it has no weight-decay.

Show that SGD with weight decay using the original loss $f_t(\theta)$ is equivalent to regular SGD on the L2-regularized loss $f_t^{reg}(\theta)$ when γ is chosen correctly, and **find such a γ in terms of λ and η.**

5. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **Roughly how many total hours did you work on this homework?**

Contributors:

- Saagar Sanghavi.
- Suhong Moon.
- Dominic Carrano.
- Babak Ayazifar.
- Sukrit Arora.
- Romil Bhardwaj.
- Kevin Li.