Exercise Sheet 5

Neural Computation
University of Birmingham

Week 5 2025

Exercise 1. You are given a mini-batch of two single-channel images, each of size 2×2:

$$X^{(1)} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad X^{(2)} = \begin{bmatrix} -1 & -2 \\ -3 & -4 \end{bmatrix}$$

Assume batch normalisation is applied. Let:

- $\gamma = 1$
- $\beta = 0$
- $\epsilon = 10^{-5}$ (is negligible here)

1. Compute the result of applying batch normalisation to this mini-batch.
2. Assume that the images are normalised using min-max normalisation, **using the min and max of the full batch** before they go into the batch norm layer. What is the result after the min-max normalisation and after the subsequent batch norm?
3. Assume that the images are separately normalised using min-max normalisation, **using the min and max of each image** before they go into the batch norm layer. What is the result after the min-max normalisation and after the subsequent batch norm?
4. What are the learnable parameters in batch normalisation?
5. How many learnable parameters are there in this example?
6. How would the number of learnable parameters change if:
   - The input images had 16 channels?
   - The spatial dimensions of the images were larger, 32×32?

Solution:

1. **Batch Normalisation Result:**

   The combined pixel values from both images are: $[1, 2, 3, 4, -1, -2, -3, -4]$

   Mean:

   $$\mu = \frac{1 + 2 + 3 + 4 + (-1) + (-2) + (-3) + (-4)}{8} = 0.00$$

   Biased Variance:

   $$\sigma^2 = \frac{1}{8} \sum_{i=1}^{8} (x_i - \mu)^2 = 7.50$$

   Standard Deviation:

   $$\sigma = \sqrt{7.50 + \epsilon} \approx 2.74$$

   Normalisation Equation:

   $$\hat{x}_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

Applying batch normalisation:

$$\text{Result} = \left[\begin{bmatrix} 0.37 & 0.73 \\ 1.10 & 1.46 \end{bmatrix}, \begin{bmatrix} -0.37 & -0.73 \\ -1.10 & -1.46 \end{bmatrix} \right]$$

2. **Min-Max Normalisation (Full Batch) then Batch Norm:**

   Min: -4, Max: 4 Min-max normalisation:

   $$x_i' = \frac{x_i + 4}{8}$$

   After batch normalisation (multiplying or shifting by a constant has no effect after normalisation):

   $$\text{Result} = \left[\begin{bmatrix} 0.37 & 0.73 \\ 1.10 & 1.46 \end{bmatrix}, \begin{bmatrix} -0.37 & -0.73 \\ -1.10 & -1.46 \end{bmatrix} \right]$$

3. **Min-Max Normalisation (Per Image) then Batch Norm:**

   Min-Max Normalised Image 1:

   $$\begin{bmatrix} 0.00 & 0.33 \\ 0.67 & 1.00 \end{bmatrix}$$

   Min-Max Normalised Image 2:

   $$\begin{bmatrix} 1.00 & 0.67 \\ 0.33 & 0.00 \end{bmatrix}$$

   Mean: 0.50 **Biased Variance:** 0.14 **Standard Deviation:** 0.37

   After batch normalisation:

   $$\text{Result} = \left[\begin{bmatrix} -1.34 & -0.45 \\ 0.45 & 1.34 \end{bmatrix}, \begin{bmatrix} 1.34 & 0.45 \\ -0.45 & -1.34 \end{bmatrix} \right]$$

4. **Learnable Parameters in Batch Normalisation:** γ (scale) and β (shift) per channel.
5. **Number of Learnable Parameters in This Example:** 2 (for the single channel).
6. **Effect of Input Dimensions:**
   - With 16 channels: $2 \times 16 = 32$ learnable parameters.
   - Larger spatial dimensions (e.g., 32×32) do not affect the number of parameters, which remain 2 per channel.

Exercise 2. You are given a row of three pixel values:

$$[0,\ 100,\ 0]$$

Your task is to upscale this row to a length of 6 pixels using **linear interpolation**, aligning the **outer edges** of the pixels (not the pixel centres).

Compute the upscaled 6-pixel row using linear interpolation.

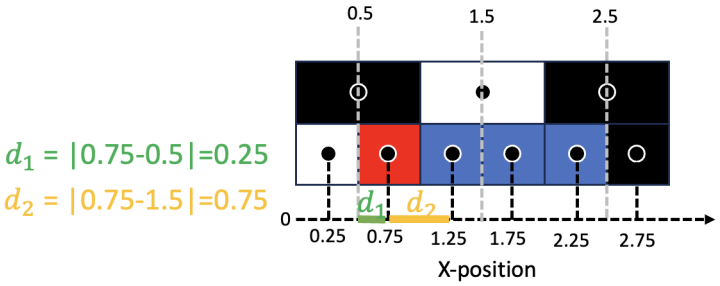
Solution:

Upscaling the row $[0,\ 100,\ 0]$ **to 6 pixels:**

Since we are aligning the edges, the interpolation points are evenly spaced between the outer edges of the three pixels. Using linear interpolation:

$$
\begin{aligned}
x_0 &= 0 \\
x_1 &= 0.75 * 0 + 0.25 * 100 = 25 \\
x_2 &= 0.25 * 0 + 0.75 * 100 = 75 \\
x_3 &= 0.75 * 100 + 0.25 * 0 = 75 \\
x_4 &= 0.25 * 100 + 0.75 * 0 = 25 \\
x_5 &= 0
\end{aligned}
$$

Every new pixel centre (except the first and last) is located between the centres of two old ones. The weights 0.75 and 0.25 correspond to the distances of the new centre to the two old ones that lie around it. Our old pixels have width 1, if our new row is 6 pixels wide and our old row is 3 pixels wide, then each new pixel is $3/6 = 1/2 = 0.5$ wide.



So the centre of the first new pixel will be located at position 0.25, the second at 0.75 etc... The first old pixel center is located at 0.5 (in the middle of the first pixel) the second at 1.5 the third at 2.5 ...

Let's now look at the first new pixel, it is located (at 0.25) to the left of the first old pixel, so it will only be influenced by the first pixel. The second new pixel is at 0.75, which it to the right of the first old pixel (at 0.5) and to the left of the second old pixel (at 1.5). The distance to the first old pixel is |0.75 - 0.5| = 0.25. The distance to the second old pixel is |0.75-1.5| = 0.75. These are the weights d1, d2 use to calculate the second new pixel. So the upscaled row is:

$$[0,\ 25,\ 75,\ 75,\ 25,\ 0]$$