

CS2109S Tutorial 8

AY 25/26 Sem 1 — [github/omgeta](https://github.com/omgeta)

- A. 1. $Y = \begin{bmatrix} -0.2 & 0 & 0.2 & 1.1 \\ -0.3 & 0.1 & 0.1 & 1.2 \\ -0.2 & 0.1 & 0.1 & 0.3 \\ -1.2 & 0 & 0.5 & 1 \end{bmatrix}$. This should be to detect the strength of right edges.
2. Max-Pool = $\begin{bmatrix} 0.1 & 1.2 \\ 0.1 & 1 \end{bmatrix}$ to keep strongest local feature
 Average-Pool = $\begin{bmatrix} -0.1 & 0.65 \\ -0.325 & 0.475 \end{bmatrix}$ to capture overall feature
3. Since $W = H = \lfloor \frac{224-11+2\cdot0}{4} \rfloor + 1 = 54$, output size is $54 \times 54 \times 96$
4. $(B, 54, 54, 96)$. This provides higher throughput (due to GPU parallelization) and steadier gradients.
- B. 1. Consider formula $r_i = r_{i-1} + (\text{kernel}_i - 1) \times j_{i-1}$ and $j_i = j_{i-1} \times \text{stride}_i$ with $r_0 = j_0 = 1$
 Then, $r_1 = 1 + (5 - 1) \times 1 = 5$, $j_1 = 1 \times 2 = 2$,
 and, $r_2 = 5 + (2 - 1) \times 2 = 7$
 Therefore, First Layer: 5×5 , Second Layer: 7×7
2. Larger receptive fields enable network to capture more global features and spatial context, better recognizing patterns across larger fields of an image. However, this involves more layers with increased computational cost.
- C. 1. Many-to-one
2. $h^{[1]} = (W^{[xh]})^\top x_1 + (W^{[hh]})^\top h^{[0]} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
 $h^{[2]} = (W^{[xh]})^\top x_2 + (W^{[hh]})^\top h^{[1]} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
3. $\hat{y}^{[2]} = \text{softmax}((W^{[hy]})^\top h^{[2]}) = \begin{bmatrix} 0.0418 \\ 0.1135 \\ 0.8390 \\ 0.0057 \end{bmatrix}$ so output is "coding"
4. $h^{[t]}$ depends on $h^{[t-1]}, x^{[t]}$ so order can change the trajectory of hidden values and hence the final output.
5. Vector size can get large for large vocabularies, and there is no relation between words (e.g. love and like)
- D. 1. Yes, but it would be less sensitive to order and more sensitive to global context
2. Preprocess with CNN to get a feature map, which is then read patch-by-patch as a sequence.