NNDL Exercise set 3 – Mathematical exercises

1.

a)

Assuming no padding (only valid convolutions), a single kernel can take $D - W$ steps of unit size before reaching the end of the input, since it must fit within the input dimensions. Furthermore, there will be one additional output from the initial placement of the kernel. Therefore one kernel produces $D-W+1$ outputs. Since there are M kernels, the total number of outputs will be $M(D - W + 1)$. Each kernel has W weights (equal to the width), so in total the layer has MW weights, because there are M kernels.

b)

The number of neurons must be the same as previously, i.e. $M(D - W + 1)$, in order to produce output of the same dimension (perhaps subject to reshaping). However, in this case all D elements of the input are connected to $M(D-W+1)$ neurons, so the layer contains $MD(D - W + 1)$ weights.

c)

We could add l_1-regularization to promote sparsity, i.e. add a term of the form

$$\lambda_1 ||w||_1 \tag{1}$$

to the objective function, where λ_1 is some controllable hyperparameter and $|| \cdot ||_1$ is the l_1-norm. For weight sharing something like

$$\lambda_2 \sum_{i \neq j} |w_i - w_j| \tag{2}$$

could be used. Here λ_2 is another controllable hyperparameter and the sum is over the weights.

2.

Let $f_i(x) = w_i x + b_i$ and $\text{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$.

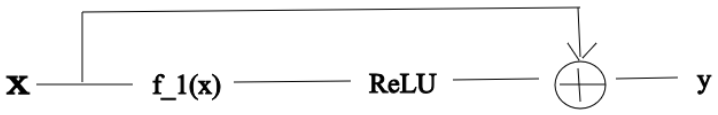

Figure 1: $y = x + \mathrm{ReLU}(f_1(x))$

1.

Consider

$$y = x + \mathrm{ReLU}(f_1(x)) = \begin{cases} x, & w_1 x + b_1 \leq 0 \\ x + w_1 x + b_1, & w_1 x + b_1 > 0 \end{cases}. \tag{3}$$

The derivative is then

$$y' = \begin{cases} 1, & w_1 x + b_1 \leq 0 \\ 1 + w_1, & w_1 x + b_1 > 0 \end{cases}. \tag{4}$$

The information flow is depicted in Fig. 1.

2.

Consider

$$\begin{aligned} y &= f_1(x) + f_2(\mathrm{ReLU}(f_1(x))) \\ &= \begin{cases} w_1 x + b_1 + b_2, & w_1 x + b_1 \leq 0 \\ w_1 x + b_1 + w_2(w_1 x + b_1) + b_2, & w_1 x + b_1 > 0 \end{cases}. \end{aligned} \tag{5}$$

Figure 2: $y = f_1(x) + f_2(\text{ReLU}(f_1(x)))$

The derivative is then The derivative is then

$$y' = \begin{cases} w_1, & w_1 x + b_1 \leq 0 \\ w_1 + w_2 w_1, & w_1 x + b_1 > 0 \end{cases}. \tag{6}$$

The information flow is depicted in Fig. 2.

3.

Consider

$$\begin{aligned} y &= x + f_2(\text{ReLU}(f_1(\text{ReLU}(x)))) \\ &= \begin{cases} x + b_2, & x \leq 0, \quad b_1 < 0 \\ x + w_2 b_1 + b_2, & x \leq 0, \quad b_1 > 0 \\ x + b_2, & x > 0, \quad w_1 x + b_1 \leq 0 \\ x + w_2(w_1 x + b_1) + b_2, & x > 0, \quad w_1 x + b_1 > 0 \end{cases}. \end{aligned} \tag{7}$$

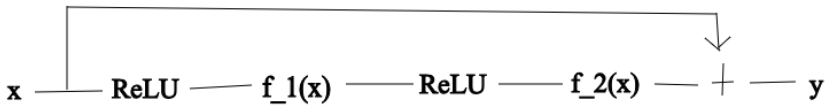

Figure 3: $y = x + f_2(\text{ReLU}(f_1(\text{ReLU}(x))))$

The derivative is then

$$y' = \begin{cases} 1, & x \leq 0, \quad b_1 < 0 \\ 1, & x \leq 0, \quad b_1 > 0 \\ 1, & x > 0, \quad w_1 x + b_1 \leq 0 \\ 1 + w_2 w_1, & x > 0, \quad w_1 x + b_1 > 0 \end{cases}. \tag{8}$$

The information flow is depicted in Fig. 3.

Discussion

In the first case the residual block can only add positive values to the input. In the second case, the residual block can also add negative values, and there is an initial linear transformation to account for all negative input. In the third case the first ReLU will zero out all negative input. Based on this I would choose the second option.

Computer assignment 3 – algorithm

Denote an image by $\mathbf{I}$ and assume we have a training set $D_{\text{base}} = \{(\mathbf{I}, y_1), \ldots, (\mathbf{I}_N, y_N)\}$. Furthermore, assume a support dataset D_{support} of labeled images consisting of

C novel classes, each novel class having K examples. The algorithm then works as follows:

- Train a feature extractor CNN $f_\theta(\mathbf{I})$ on D_{base} by minimizing the loss $\arg\min_{\theta,\mathbf{W}} \sum_{(\mathbf{I},y)\in D_{\text{base}}} l(\mathbf{W}^T f_\theta(\mathbf{I}), y)$, where l is the cross-entropy loss.
- If multi-shot, then compute the centroids for each class in D_{support}.
- Select test image in feature space $\hat{x} = f_\theta(\mathbf{I})$
- Center the image $\hat{x} \leftarrow \hat{x} - \frac{1}{|D_{\text{base}}|} \sum_{x\in D_{\text{base}}} x$
- Normalize $\hat{x} \leftarrow \frac{\hat{x}}{||\hat{x}||_2}$
- Assign the label of the most similar support image to the test image $\hat{x}$: $y(\hat{x}) = \arg\min_{c\in\{1,\ldots,C\}} d(\hat{x}, \hat{x}_C)$, where $d(\hat{x}, \hat{x}_C)$ is the Euclidean distance. In multi-shot setting this equation is applied to the centroids instead of individual images.