Assignment 3 Key

1. General Questions

   (a) The network amounts to a one-layer linear module, because the convolution and other linear layers can be combined by matrix multiplications.

   (b) The answer to the question can be seen in the following two questions equations; for sparse coding, the objective function can be written

   $$\min_{\alpha,D} \|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1,$$

   where α, D are the code and dictionary respectively, and λ is a hyperparameter which weighs the loss and sparsity penalty.

   On the other hand, the auto-encoders minimize the following objective function:

   $$\min_{E,D} \|x - g_D(f_E(x))\|_2^2$$

   where E, D denote the encoder weight and decoder weight respectively.

2. Softmax regression gradient calculation

   (a) Assume $y_k = 1$, which indicates

   $$l(\mathbf{y}, \hat{\mathbf{y}}) = -\log \hat{y}_k \tag{1}$$

   $$\frac{\partial l}{\partial \hat{y}_k} = -\frac{1}{\hat{y}_k} \tag{2}$$

   Denote $\mathbf{a} = \mathbf{W}\mathbf{x} + \mathbf{b}$, from assignment 1, we have

   $$\frac{\partial \hat{y}_k}{\partial a_i} = \begin{cases} \hat{y}_k(1 - \hat{y}_i), & \text{for } i = k \\ -\hat{y}_k\hat{y}_i, & \text{for } i \neq k \end{cases} \tag{3}$$

   $$= \hat{y}_k(y_i - \hat{y}_i) \tag{4}$$

   Because $\frac{\partial a_i}{\partial \mathbf{W}_j} = \mathbf{0}$ for $i \neq j$, we can have

   $$\frac{\partial l}{\partial W_{i,j}} = \frac{\partial l}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial a_i} \cdot \frac{\partial a_i}{\partial W_{i,j}} \tag{5}$$

   $$= -\frac{1}{\hat{y}_k} \cdot \hat{y}_k(y_i - \hat{y}_i) \cdot x_j \tag{6}$$

   $$= (\hat{y}_i - y_i)x_j \tag{7}$$

(b) Gradient backpropagated from criterion module (Eq. 1) will explode. In practice, this will not happen unless intentionally initialized to be that way. Softmax function will guarantee none of the activations reach exact zero.

3. Chain rule

(a) We can use sigmoid, square, weighted sum, and division as basic modules. Gradients can be calculated with a forward pass followed by a backward pass.
Denote:

$$\begin{aligned}\sigma_x &= \sigma(x)\\ \sigma_y &= \sigma(y)\\ a &= x^2\\ b &= a + \sigma_y\\ c &= 3x + y - \sigma_x\\ f &= b/c\end{aligned}$$

(b) Forward pass:

$$\begin{aligned}\sigma_x &= 0.7311\\ \sigma_y &= 0.5\\ a &= 1\\ b &= 1.5\\ c &= 2.2689\\ f &= 0.6611\end{aligned}$$

Backward pass:

$$\begin{aligned}\frac{\partial f}{\partial b} &= \frac{1}{c} = 0.4407\\ \frac{\partial f}{\partial c} &= -\frac{b}{c^2} = -0.2914\\ \frac{\partial f}{\partial a} &= \frac{\partial f}{\partial b}\cdot 1 = 0.4407\\ \frac{\partial f}{\partial \sigma_x} &= \frac{\partial f}{\partial c}\cdot(-1) = 0.2914\\ \frac{\partial f}{\partial \sigma_y} &= \frac{\partial f}{\partial b}\cdot 1 = 0.4407\\ \frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\cdot\frac{\partial a}{\partial x} + \frac{\partial f}{\partial c}\cdot\frac{\partial c}{\partial x} + \frac{\partial f}{\partial \sigma_x}\cdot\frac{\partial \sigma_x}{\partial x} &= 0.4407*2 - 0.2914*3 + 0.2914*0.7311*(1-0.7311)\\ &= 0.0645\\ \frac{\partial f}{\partial y} = \frac{\partial f}{\partial c}\cdot\frac{\partial c}{\partial y} + \frac{\partial f}{\partial \sigma_y}\cdot\frac{\partial \sigma_y}{\partial y} &= -0.2914*1 + 0.4407*0.5*(1-0.5) = -0.1812\end{aligned}$$

4. Variants of pooling

(a) i. Max-pooling, `nn.SpatialMaxPooling` or `cudnn.SpatialMaxPooling`;
ii. Average-pooling, `nn.SpatialAveragePooling`;
iii. LP-pooling, `nn.SpatialLPPooling`

(b) Denote the input feature maps to the pooling module as $X \in \mathbb{R}^{n \times H_{in} \times W_{in}}$ where n, H_{in}, W_{out} represent the number of feature maps, height and width of them respectively, the output feature maps of the module as $Y \in \mathbb{R}^{n \times H_{out} \times W_{out}}$ where H_{out} and W_{out} are the height and size of the output maps.

- For Max-pooling,
$$Y^c_{i,j} = \max X^c_{(x,y) \in S^c_{i,j}},$$
where c indexes the feature map and $S^c_{i,j}$ indexes the sub-region of input.
- For Avearge-pooling,
$$Y^c_{i,j} = \frac{1}{h \times w} \sum_x^h \sum_y^w X^c_{(x,y) \in S^c_{i,j}},$$
where $h \times w$ denote the pooling region sizes.
- For LP-pooling,
$$Y^c_{i,j} = (\sum_x^h \sum_y^w (X^c_{(x,y) \in S^c_{i,j}})^p)^{\frac{1}{p}},$$

(c) Pooling is adopted mainly due to two reasons; (i)- it introduces spatial invariance to the network; (ii)- it effectively reduces the computation cost.

5. Convolution

(a) 9

(b) Left table of Table 1

(c) Right table of Table 1

109	92	72
108	85	74
110	74	79

4	7	10	6	3
9	17	25	16	8
11	23	34	23	11
7	16	24	17	8
2	6	9	7	3

Table 1: Forward propagated output and Backpropagated gradient

6. Optimisation

(a) $L(\mathbf{x}) = \|\mathbf{x} - \sigma_2(\mathbf{W}'(\sigma_1(\mathbf{W}\mathbf{x} + b)) + b')\|^2$

(b) Let $A = \sigma_2(\mathbf{W}'(\sigma_1(\mathbf{W}\mathbf{x} + b)) + b')$ and $B = \sigma_1(\mathbf{W}\mathbf{x} + b)$
Then,

$$L(\mathbf{x}) = \|\mathbf{x} - A\|^2 \tag{8}$$
$$\frac{\partial L}{\partial A} = -2(\mathbf{x} - A) \tag{9}$$
$$\frac{\partial A}{\partial B} = \sigma_2'(\mathbf{W}'B + b')\mathbf{W}' \tag{10}$$
$$\frac{\partial A}{\partial \mathbf{W}'} = \sigma_2'(\mathbf{W}'B + b')B \tag{11}$$
$$\frac{\partial A}{\partial b'} = \sigma_2'(\mathbf{W}'B + b') \tag{12}$$
$$\frac{\partial B}{\partial \mathbf{W}} = \sigma_1'(\mathbf{W}\mathbf{x} + b)\mathbf{x} \tag{13}$$
$$\frac{\partial B}{\partial b} = \sigma_1'(\mathbf{W}\mathbf{x} + b) \tag{14}$$

Now,

$$\frac{\partial L}{\partial \mathbf{W}'} = \frac{\partial L}{\partial A}\frac{\partial A}{\partial \mathbf{W}'} \tag{15}$$
$$\frac{\partial L}{\partial b'} = \frac{\partial L}{\partial A}\frac{\partial A}{\partial b'} \tag{16}$$
$$\frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L}{\partial A}\frac{\partial A}{\partial B}\frac{\partial B}{\partial \mathbf{W}} \tag{17}$$
$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial A}\frac{\partial A}{\partial B}\frac{\partial B}{\partial b} \tag{18}$$

(c) The gradient descent steps are the following (Assuming the parameter at timestep n is the parameter subscripted by n and γ is the learning rate):

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \gamma\frac{\partial L}{\partial \mathbf{W}_n} \tag{19}$$
$$\mathbf{W}'_{n+1} = \mathbf{W}'_n - \gamma\frac{\partial L}{\partial \mathbf{W}'_n} \tag{20}$$
$$b'_{n+1} = b'_n - \gamma\frac{\partial L}{\partial b'_n} \tag{21}$$
$$b_{n+1} = b_n - \gamma\frac{\partial L}{\partial b_n} \tag{22}$$

(d) Assuming μ is the momentum parameter.

$$\mathbf{V}_{n+1} = \mu \mathbf{V}_n - \gamma \frac{\partial L}{\partial \mathbf{W}_n} \tag{23}$$

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \mathbf{V}_{n+1} \tag{24}$$

$$\mathbf{V'}_{n+1} = \mu \mathbf{V'}_n - \gamma \frac{\partial L}{\partial \mathbf{W'}_n} \tag{25}$$

$$\mathbf{W'}_{n+1} = \mathbf{W'}_n + \mathbf{V'}_{n+1} \tag{26}$$

$$c_{n+1} = \mu c_n - \gamma \frac{\partial L}{\partial b_n} \tag{27}$$

$$b_{n+1} = b_n + c_{n+1} \tag{28}$$

$$c'_{n+1} = \mu c'_n - \gamma \frac{\partial L}{\partial b'_n} \tag{29}$$

$$b'_{n+1} = b'_n + c'_{n+1} \tag{30}$$

7. Top-k error
   This term is mainly used in large-scale many-class classification problem, such as ImageNet, due to the fact that classes are not completely mutual exclusive.

   Top-1 error measures only the top prediction of the model. On the contrary, when using Top-5 error, it is counted as correct as long as the top five predicted labels contain the ground-truth label.

8. t-SNE

   (a) Because of an exponential growth in the number of datapoints at a particular distance from a point in the high-dimensional space, the area of the two-dimensional map that is available to accommodate moderately distant datapoints is not large enough compared with the area available to accommodate nearby datapoints. This is called the crowding problem.

   t-SNE alleviates this problem by using t-distribution to model distances in the low-dimensional map. This is because the t-distribution has much heavier tails than a Gaussian to convert distances into probabilities. This allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map and, as a result, it eliminates the unwanted attractive forces between map points that represent moderately dissimilar datapoints.

   (b) Let $d_{ij} = \|y_i - y_j\|$ and $Z = \sum_{k \neq l} exp(-\|y_k - y_l\|^2)$.
   Then, $q_{ij} = \frac{exp(-d_{ij}^2)}{Z}$.

$$\frac{\partial C}{\partial y_i} = \sum_j (\frac{\partial C}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial y_i} + \frac{\partial C}{\partial d_{ji}} \frac{\partial d_{ji}}{\partial y_i}) \tag{31}$$

$$= 2 \sum_j (\frac{\partial C}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial y_i}) \tag{32}$$

$$\frac{\partial d_{ij}}{\partial y_i} = \frac{y_i - y_j}{d_{ij}} \tag{33}$$

$$\frac{\partial C}{\partial d_{ij}} = -\sum_{k \neq l} p_{kl} \frac{\partial (log(q_{kl}))}{\partial d_{ij}} \tag{34}$$

$$= -\sum_{k \neq l} p_{kl} \frac{\partial (log(\frac{q_{kl} Z}{Z}))}{\partial d_{ij}} \tag{35}$$

$$= -\sum_{k \neq l} p_{kl} \frac{\partial (log(q_{kl} Z) - log(Z))}{\partial d_{ij}} \tag{36}$$

$$= -\sum_{k \neq l} p_{kl} (\frac{\partial (log(exp(-d_{kl}^2))}{\partial d_{ij}} - \frac{\partial log(Z))}{\partial d_{ij}}) \tag{37}$$

$$= 2 p_{ij} d_{ij} + \sum_{k \neq l} p_{kl} \frac{\partial log(Z))}{\partial d_{ij}} \tag{38}$$

$$= 2 p_{ij} d_{ij} + 1. \frac{exp(-d_{ij}^2)}{Z} (-2 d_{ij}) \tag{39}$$

$$= 2 p_{ij} d_{ij} - 2 q_{ij} d_{ij} \tag{40}$$

$$= 2 d_{ij} (p_{ij} - q_{ij}) \tag{41}$$

$$\frac{\partial C}{\partial y_i} = 2 \sum_j (2 d_{ij} (p_{ij} - q_{ij}) \frac{y_i - y_j}{d_{ij}}) \tag{42}$$

$$= 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \tag{43}$$

9. Proximal gradient descent

   (a) $\frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 + t\|\mathbf{z}\|_1 = \sum_{i=1}^n \frac{1}{2}(z_i - x_i)^2 + t|z_i|$. Minimizing this function is equivalent to minimizing each subfunction independently.

      i. When $z_i \neq 0$, taking the derivative with respect to z_i and set to zero, we have

$$z_i - x_i + t \cdot \text{sign}(z_i) = 0 \tag{44}$$

      Notice that in this case $x_i = z_i + t \cdot \text{sign}(z_i)$ which implies $|x_i| = |z_i| + t > t$ and $\text{sign}(x_i) = \text{sign}(z_i)$ (we assume $t > 0$). Thus,

$$z_i = x_i - t \cdot \text{sign}(x_i) = (|x_i| - t)_+ \text{sign}(x_i) \tag{45}$$

ii. When $z_i = 0$, the optimality condition becomes

$$0 \in z_i - x_i + t \cdot \partial |z_i| \tag{46}$$
$$-x_i - t \leq 0 \leq -x_i + t \tag{47}$$
$$-t \leq x_i \leq t \tag{48}$$

z_i can also be written as $(|x_i| - t)_+ \text{sign}(x_i)$ which in this case is zero.

(b) Consider the sparse coding objective function

$$f(x) = \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1 \tag{49}$$

Substitute into the proximal gradient descent framework and use the result from (a), we get the update rule for x:

$$x_{k+1} = \text{prox}_{h,\alpha_k}(x_k - \alpha_k A^T(Ax - y)) \tag{50}$$
$$= S_{\lambda\alpha_k}(x_k - \alpha_k A^T(Ax - y)) \tag{51}$$

which is the ISTA algorithm.

(c) $\mathbf{u} = \text{prox}_{h,t}(\mathbf{x})$ means $\mathbf{u}$ is the minimizer of function $\phi(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 + th(\mathbf{z})$. Using the optimality condition:

$$0 \in \mathbf{u} - \mathbf{x} + t \cdot \partial h(\mathbf{u}) \tag{52}$$

Rearranging

$$\frac{\mathbf{x} - \mathbf{u}}{t} \in \partial h(\mathbf{u}) \tag{53}$$

(d) We are given

$$x_k - \alpha_k G_{\alpha_k}(x_k) = \text{prox}_{h,\alpha_k}(x_k - \alpha_k \nabla g(x_k)) \tag{54}$$

Using the result from (c)

$$\frac{x_k - \alpha_k \nabla g(x_k) - x_k + \alpha_k G_{\alpha_k}(x_k)}{\alpha_k} \in \partial h(x_{k+1}) \tag{55}$$
$$G_{\alpha_k}(x_k) - \nabla g(x_k) \in \partial h(x_{k+1}) \tag{56}$$