Deep Learning Teaching Kit

Lab 3 (designed for individual work)

---

1. **General Questions**

   (a) Consider the case where no non-linear activation functions are applied between modules, explain the approach to simplify (distill) the whole network into one module?

   (b) What is the difference between the dictionary used in sparse coding and the counterpart in autoencoders?

2. **Softmax regression gradient calculation**

   Consider a simple Softmax regression model,

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

$$\mathbf{x} \in \mathbb{R}^d, \mathbf{W} \in \mathbb{R}^{k \times d}, \mathbf{b} \in \mathbb{R}^k$$

   where d is the input dimension, k is the number of classes, σ is the softmax function:

$$\sigma(\mathbf{a})_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

   (a) Given the cross-entropy loss

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log \hat{y}_i$$

   $\mathbf{y}$ is the one-hot vector representing true labels ($[0, 0, \cdots, 1, 0, 0, \cdots]^T$ with the 1 corresponding to the true label), derive $\frac{\partial l}{\partial W_{i,j}}$. (You can use your results from Assignment 1)

   (b) What happens to the loss function and the gradients when $y_{c_1} = 1, \hat{y}_{c_2} = 1, c_1 \neq c_2$? Why there is no need to worry about this situation?

3. **Chain rule**

   We have the following function

$$f(x,y) = \frac{x^2 + \sigma(y)}{3x + y - \sigma(x)}$$

   where σ is the sigmoid function.

   (a) Without explicitly deriving the formula for $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$, describe how you would calculate the gradients using chain rule.

   (b) Evaluate the gradients at $x = 1, y = 0$ using your approach.

4. **Variants of pooling**

   (a) List three different kinds of pooling, and their corresponding module implemented in torch.

   (b) Write down the mathematical forms of these three pooling modules.

   (c) Pick one of the pooling listed and describe the reason for incorporating it into a deep learning system.

5. **Convolution**

   Table 1 depicts two matrices. One of them (5x5 one) represents an image. The second (3x3 matrix) represents a convolution kernel. (Consider the bias term to be zero)

   (a) How many values will be generated if we forward propagate the image over the given convolution kernel?

   (b) Calculate these values.

   (c) Suppose the gradient backpropagated from the layers above this layer is a 3x3 matrix of all 1s. Write the value of the gradient (w.r.t. input) backpropagated out of this layer.

4	5	2	2	1
3	3	2	2	4
4	3	4	1	1
5	1	4	1	2
5	1	3	1	4

4	3	3
5	5	5
2	4	3

Table 1: Image Matrix (5x5) and a convolution filter (3x3)

6. **Optimization**

   (a) Write down the mathematical formula for the reconstruction loss of an autoencoder.

   (b) Write down the mathematical formula for the gradient of the loss with respect to the parameters.

(c) Write down the gradient descent step for this reconstruction loss.

(d) Write down this step with a momentum term.

7. **Top-k error**

   ImageNet uses top-5 and top-1 errors to evaluate classification performances. Define top-k error. Why do you think ImageNet uses both top-5 and top-1 errors?

8. **t-SNE**

   (a) What is the crowding problem and how does t-SNE alleviate it? Give details.

   (b) The cost function of symmetric SNE is given by:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

   where:

$$q_{ij} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} exp(-\|y_k - y_l\|^2})$$

   and:

$$p_{ij} = \frac{exp(-\|x_i - x_j\|^2/2\sigma^2)}{\sum_{k \neq l} exp(-\|x_k - x_l\|^2/2\sigma^2)}.$$

   Derive $\frac{\partial C}{\partial y_i}$.

9. **Proximal gradient descent**

   (a) Proximal operator is defined as

$$\text{prox}_{h,t}(\mathbf{x}) = \text{argmin}_{\mathbf{z}} \frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 + th(\mathbf{z})$$

   Prove that when $h(\mathbf{z}) = \|\mathbf{z}\|_1$, the solution is the soft-thresholding function $S_t(\mathbf{x}) = (|\mathbf{x}| - t\mathbf{1})_+ \odot \text{sign}(\mathbf{x})$. (operations are all element-wise)

   (b) Recall the optimization problem we discussed with function $f(x)$

$$f(x) = g(x) + h(x)$$

$$g(x) : \text{convex, differentiable}$$

$$h(x) : \text{convex, simple}$$

   Proximal gradient descent uses the following update rule:

$$x_{k+1} = \text{prox}_{h,\alpha_k}(x_k - \alpha_k \nabla g(x_k))$$

   Show that ISTA is one example of proximal gradient descent methods.

   (c) Given $u = \text{prox}_{h,t}(x)$, show

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

   where $\partial h(u)$ is the subdifferential of function h evaluated at u.

(d) The update rule for proximal gradient descent method can be reformulated as

$$x_{k+1} = x_k - \alpha_k G_{\alpha_k}(x_k)$$

$$G_{\alpha_k}(x_k) = \frac{x_k - \mathrm{prox}_{h,\alpha_k}(x_k - \alpha_k \nabla g(x_k))}{\alpha_k}$$

Show that

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

Submission

Send your submission to your corresponding TA by the deadline. Please use the following title for your email.

`[`*CourseName* `YOUR_NAME] Submission Lab3`