ELEG 5491: Homework #1

Due on Tuesday, February 14, 2017, 3:30pm (in class)

Xiaogang Wang

Problem 1

[**15 points**]

Cross entropy is often used as the objective function when training neural networks in classification problems. Suppose the training set includes N training pairs $\mathcal{D} = \{(\mathbf{x}_i^{(\text{train})}, y_i^{(\text{train})})\}_{i=1}^N$, where $\mathbf{x}_i^{(\text{train})}$ is a training sample and $y_i^{(\text{train})} \in \{1, \ldots, c\}$ is its class label. $\mathbf{z}_i$ is the output of the network given input $\mathbf{x}_i^{(\text{train})}$ and the nonlinearity of the output layer is softmax. $\mathbf{z}_i$ is a c dimensional vector, $z_{i,k} \in [0, 1]$ and $\sum_{k=1}^c z_{i,k} = 1$. Please write the objective function of cross entropy and show that it is equivalent to the negative log-likelihood on the training set, assuming the training samples are independent.

Problem 2

[**25 points**]

x_1 and x_2 are two input variables, and y is the target variable to be predicted. The network structure is shown in Figure 1(a). $h_{11} = f_{11}(x_1)$, $h_{12} = f_{12}(x_2)$, and $y = g(h_{11}, h_{12})$.

- Assuming $x_1 \in [0, 1]$ and $x_2 \in [0, 1]$, in order to obtain the decision regions in Figure 1(b), decide functions f_{11}, f_{12}, and g. [**5 points**]
- Now we extend the range of x_1 and x_2 to $[0, 2]$. Please add one more layer to Figure 1(a) in order to obtain the decision regions in Figure 1(c). [**5 points**]
- Although the decision boundaries in Figure 1(c) look complicated, there exist regularity and global structure. Please explain such regularity and global structure. Based on your observation, draw the decision boundaries when the range of x_1 and x_2 are extended to [0, 4]. [**5 points**]

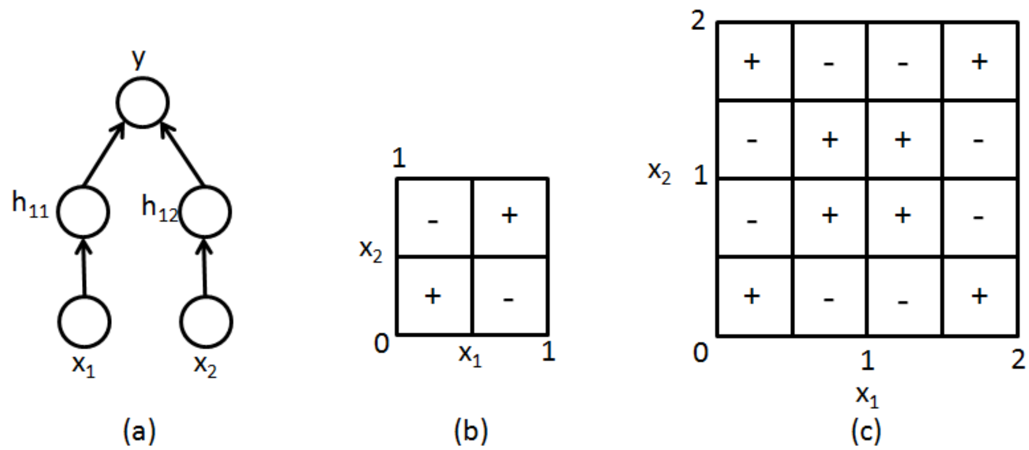

Figure 1:

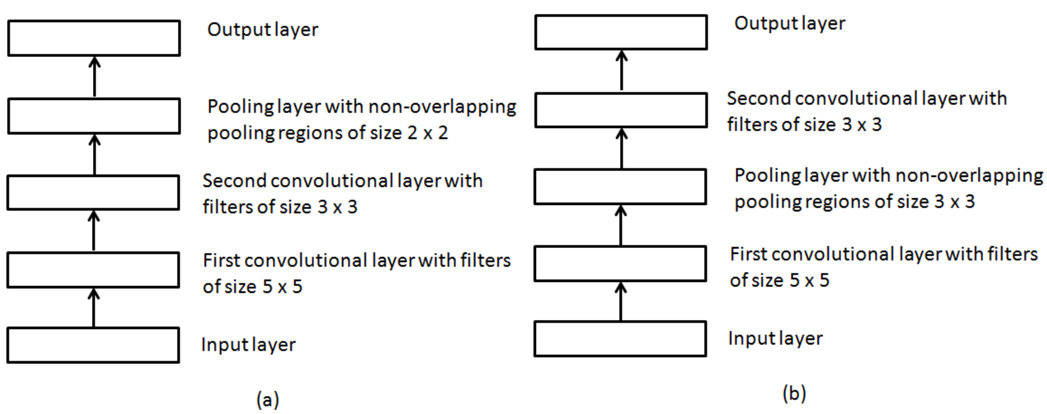

Figure 2:

- Following the question above and assuming the range of x_1 and x_2 is extended to $[0, 2^n]$, draw the network structure and the transform function in each layer, in order to obtain the decision regions with the same regularity and global structure in Figure 1 (b) and (c). The complexity of computation units should be $O(n)$. [**5 points**]
- Assuming the range of x_1 and x_2 is $[0, 2^n]$ and only one hidden layer is allowed, specify the network structure and transform functions. [**5 points**]

Problem 3

[**20 points**]

Figure 2 shows two convolutional neural networks. What is the receptive field of a neuron as the output the pooling layer in (a)? What is the receptive field of a neuron as the output the second convolutional layer in (b)? Justify your answers. We assume the the stride is 1 in the covlutional layer and the stride is equal to the size of the pooling region in the pooling layer.

Problem 4

[**30 points**]

Equivariance is an appealing property when design neural network operations. It means that transforming the input image (*e.g.*, translation) will also transform the output feature maps similarly after certain operations.

Formally, denote the image coordinate by $\mathbf{x} \in \mathbb{Z}^2$, and the pixel values at each coordinate by a function $f : \mathbb{Z}^2 \mapsto \mathbb{R}^K$, where K is the number of image channels. A convolution filter can also be formulated as a function $w : \mathbb{Z}^2 \mapsto \mathbb{R}^K$. Note that f and w are zero outside the image and filter kernel region, respectively. The convolution operation (correlation indeed for simplicity) is thus defined by

$$[f * w](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(\mathbf{y}) w_k(\mathbf{y} - \mathbf{x}). \tag{1}$$

1. **[15 pts]** Let $L_\mathbf{t}$ be the translation $\mathbf{x} \to \mathbf{x} + \mathbf{t}$ on the image or feature map, *i.e.*, $[L_\mathbf{t} f](\mathbf{x}) = f(\mathbf{x} - \mathbf{t})$. Prove that convolution has equivariance to translation:

$$[[L_\mathbf{t} f] * w](\mathbf{x}) = [L_\mathbf{t}[f * w]](\mathbf{x}), \tag{2}$$

   which means that first translating the input image then doing the convolution is equivalent to first convolving with the image and then translating the output feature map.

2. **[15 pts]** Let $L_\mathbf{R}$ be the 90°-rotation on the image or feature map, where

$$\mathbf{R} = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) \\ \sin(\pi/2) & \cos(\pi/2) \end{bmatrix}, \tag{3}$$

   then $[L_\mathbf{R} f](\mathbf{x}) = f(\mathbf{R}^{-1}\mathbf{x})$. However, convolution is not equivariant to rotations, *i.e.*, $[L_\mathbf{R} f] * w \neq L_\mathbf{R}[f * w]$, which is illustrated by Figure 3 ((a) is not equivalent to (b) rotated by 90^o). In order to establish the equivalence, the filter also needs to be rotated (i.e. (b) is equivalent to (c) in Figure 3). Prove that:

$$[[L_\mathbf{R} f] * w](\mathbf{x}) = L_\mathbf{R}[f * [L_{\mathbf{R}^{-1}} w]](\mathbf{x}). \tag{4}$$

3. **[optional]** To make convolution equivariant to rotations, we need to extend the definition of convolution and transformation. Recall a group $(G, \otimes)$ in algebra is a set G, together with an binary operation $\otimes$, which satisfies four requirements:

   Closure $a \otimes b \in G, \forall a, b \in G$.

   Associativity $(a \otimes b) \otimes c = a \otimes (b \otimes c), \forall a, b, c \in G$.

   Identity element There exists a unique $e \in G, e \otimes a = a \otimes e = a, \forall a \in G$.

   Inverse element $\forall a \in G, \exists a^{-1} \in G, a \otimes a^{-1} = a^{-1} \otimes a = e$.

   We can formulate 90°-rotation and translation by a group $(G, \otimes)$ consisting of

$$\mathbf{g}(r, u, v) = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

   where $r \in \{0, 1, 2, 3\}$ and $(u, v) \in \mathbb{Z}^2$. $G = \{g\}$ and $\otimes$ is matrix multiplication. Translation is a special case of G when $r = 0$ (i.e. $g(0, u, v)$) and rotation is a special case of G when $u = v = 0$ (i.e. g(r, 0, 0)).

Problem 4 continued on next page. . .

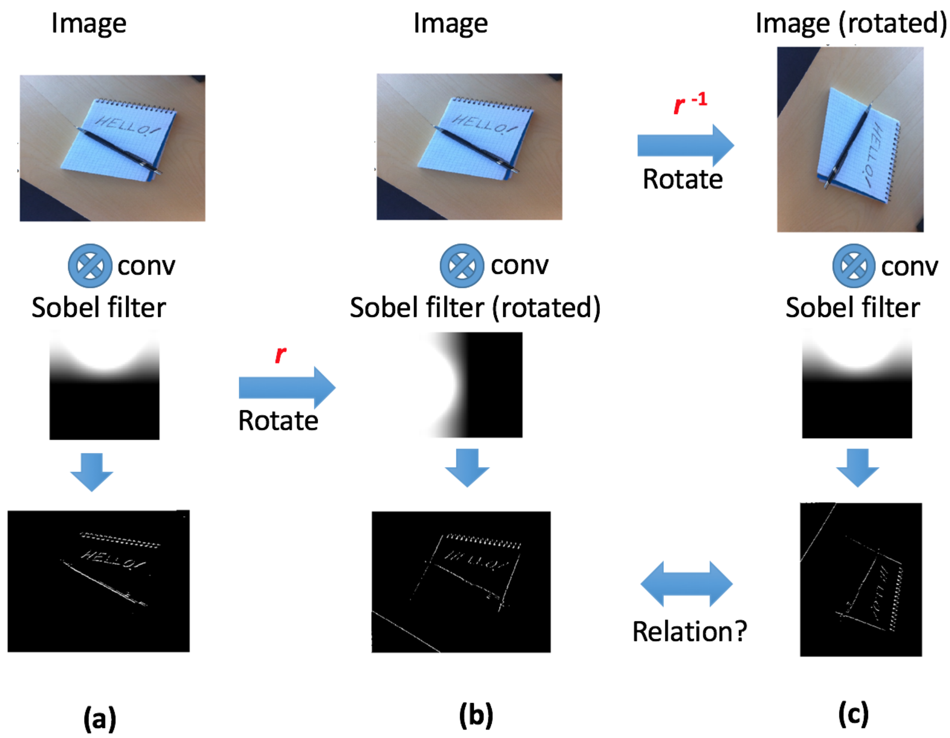

Figure 3: Equivariance relationship between convolution and rotation. (a) An image is convolved with a Sobel filer to detect horizontal edges. (b) The filter is rotated counterclockwise and then convolves the original image. (c) The image is first rotated clockwise, then it is convolved with the filter.

A key concept is to extend the definition of both the feature f and the filter w to G. Imagine the feature map is duplicated four times with rotation of $0°$, $90°$, $180°$, and $270°$. Then $f(\mathbf{g})$ is the feature values at particular rotated pixel coordinate, and the convolution operation becomes

$$[f * w](\mathbf{g}) = \sum_{\mathbf{h} \in G} \sum_{k=1}^{K} f_k(\mathbf{h}) w_k(\mathbf{g}^{-1}\mathbf{h}). \tag{6}$$

A rotation-translation $u \in G$ on the feature map is thus $[L_{\mathbf{u}} f](\mathbf{g}) = f(\mathbf{u}^{-1}\mathbf{g})$. Prove that under such extensions, the convolution is equivariant to rotation-translation:

$$[[L_{\mathbf{u}} f] * w](\mathbf{g}) = [L_{\mathbf{u}}[f * w]](\mathbf{g}). \tag{7}$$

Briefly explain how to implement this group convolution with traditional convolution and by rotating the feature map or filter.