ELEG 5491: Homework #1

Due on Thursday, February 21, 2019, 7:30pm

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 $y_i^{(\text{train})} \in \{1, \dots, c\}$ 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

[**20 points**]

Design a three layer neural network whose decision boundary is as shown in Figure 1. The gray region belongs to class 1 and other region belongs to class 0. Show your network structure, weights and nonlinear activation function.

Problem 3

[**20 points**]

Consider a three layer neural network whose structure is shown in Figure 2. You are required to calculate the sensitivty $\delta_k = -\frac{\partial J}{\partial net_k}$ at the output node k, where J is the objective function to be minimized and net_k is the net activation of the output node k. We consider two cases, where the objective function J and the nonlinear activation function at the output layer are chosen differently.

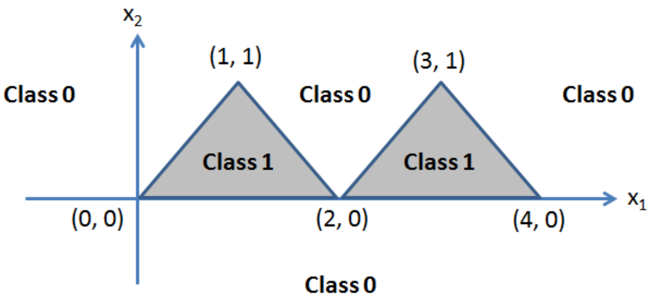

Figure 1:

- In the first case, J is chosen as the squared error $J(\mathbf{W}) = \frac{1}{2}||\mathbf{t} - \mathbf{z}||_2^2 = \frac{1}{2}\sum(t_k - z_k)^2$, where z_k is the prediction at the output node k and t_k is the corresponding target value. In the classification problem, only one t_k equals to 1 (corresponding the ground truth class) and all the other t_ks are all zeros. Sigmoid $f(net_k) = 1/(1 + e^{-net_k})$ is chosen as the activation function at the output layer. Calculate the sensitivity δ_k in terms t_k, z_k and net_k. Show that all the δ_k could be close to zero even if the prediction error is large and explain why this is bad.
- In the second case, the objective function is chosen as cross entropy $J(\mathbf{W}) = -\sum_{k=1}^{c} t_k \log(z_k)$ and the nonlinear activation function at the output layer is chosen as softmax $f(net_k) = \frac{e^{net_k}}{\sum_{k'=1}^{C} e^{net_{k'}}}$. Calculate the sensitivity t_k again. Prove that if the prediction error is large, at least one of the δ_k will be large.

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

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

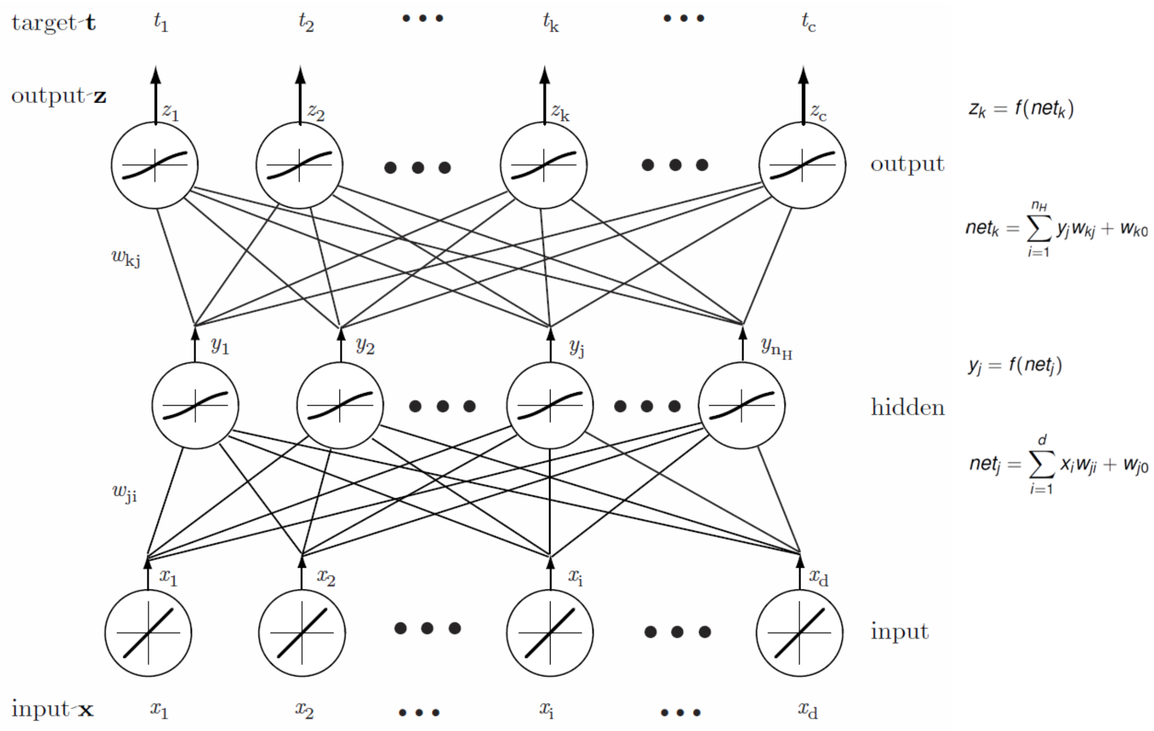

Figure 2:

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)).

   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

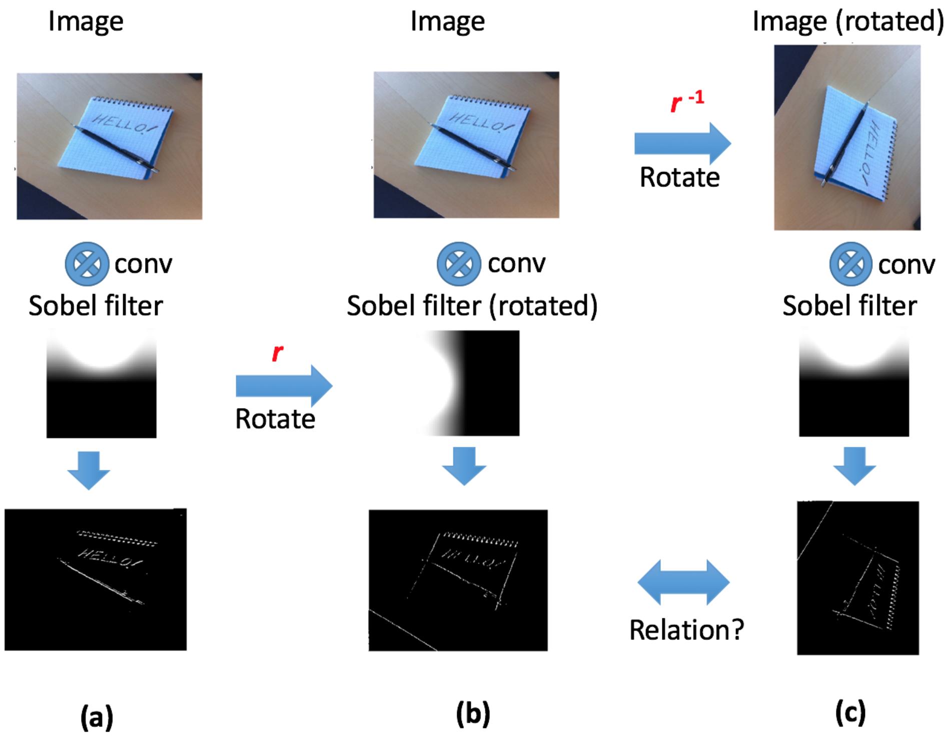

Figure 3: Equivariance relationship between convolution and rotation. (a) An image is convolved with a Sobel filer to detect horizontal edges. (b) The filter is rotated counterclockwise and then convolves the original image. (c) The image is first rotated clockwise, then it is convolved with the filter.

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.