UNIVERSITY OF ILLINOIS
Department of Electrical and Computer Engineering
ECE 417 Multimedia Signal Processing

Lecture 22 Sample Problems

Problem 22.1

You have a database full of 10×10 images, $x_i[m,n]$ where $0 \leq m \leq 9$, $0 \leq n \leq 9$. Images are drawn from one of two classes: either $y_i = 0$ or $y_i = 1$. If $y_i = 0$, then all of the pixels of x_i are either $x_i[m,n] = 0$ or $x_i[m,n] = 1$, with equal probability. If $y_i = 1$, then exactly one of the pixels, chosen at random, has a value of $x_i[m,n] = 200$; all of the other pixels are uniformly distributed between 0 and 1.

1. Construct a linear classifier, $\hat{y}_i = u(\vec{w}^T\vec{x}_i + b)$, where $u(\cdot)$ is the unit step function, $\vec{x}_i$ is a vectorized version of the image, $\vec{w}$ is a weight vector, and b is an offset. Choose values of $\vec{w}$ and b such that the classifier has zero error.
2. Start with the all-zeros image, $x_{\text{test}}[m,n] = 0$, which has the true label $y_{\text{test}} = 0$. Use your weight vector, from part (a), to create $x_{\text{adversarial}}[m,n] = x_{\text{test}}[m,n] + v[m,n]$ with the smallest possible $v[m,n]$ that will cause your linear classifier from part (a) to output a label of $\hat{y} = 1$. Argue that, though the classifier thinks that $x_{\text{adversarial}}[m,n]$ belongs to class 1, a human observer would probably think it still belongs to class 0.

Problem 22.2

Consider the following two-layer fully-connected neural network:

$$\vec{a} = U\vec{x}$$
$$\vec{y} = f(\vec{a})$$
$$\vec{b} = V\vec{y}$$
$$\vec{z} = g(\vec{b})$$

Suppose that your training database contains lots of noise-free examples of the signal $\vec{x} = \vec{s}$, and its true labels $\vec{\zeta}$, and your error criterion is cross-entropy, i.e.,

$$E = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{r}\zeta_{i\ell}\ln z_{i\ell}$$

Suppose that in the world, you expect to encounter noisy signals of the form $\vec{x} = \vec{s} + \vec{v}$ where $\vec{v}$ is a noise vector drawn from some known probability density $p(\vec{v})$ that is independent of the correct label. Design an adversarial training procedure that you can use to make sure your neural network is as effective as possible, despite the added noise. Specify the trainable parameters of the adversary. Specify both the noise-robust-primary training criterion, and the adversary's training criterion, and specify how both of them depend on the adversary's parameters and/or outputs.