CIS6930/4930 Intro to Computational Neuroscience Spring 2016
Home Work Assignment 3:
Due Thursday 04/07/16 before class

1. Consider the following function over the range $[0, 1]$

$$f(x) = -2 \times x \quad \text{if} \quad x \in [0, \frac{1}{3}]$$

$$f(x) = 1 \quad \text{if} \quad x \in (\frac{1}{3}, \frac{2}{3})$$

$$f(x) = 0 \quad \text{if} \quad x \in [\frac{2}{3}, 1]$$

   First translate and scale uniformly the domain of the function so that it now lies on $[-\pi, +\pi]$. All future references to $f(x)$ is this scaled and translated version. Your goal will be to find an approximation of this function as a Fourier series, and show the graphs of successive approximations overlayed on the actual function.

   Consider the Fourier basis e^{inx} for $n = -N, ..., +N$, and the corresponding sum

$$\sum_{n=-N}^{+N} c_n e^{inx}$$

   Calculate the values of c_n by numerically approximating the integral

$$\int_{-\pi}^{+\pi} f(x) e^{-inx} dx$$

   , that is, by dividing the range $[-\pi, +\pi]$, into small intervals and approximating the integral as a sum.

   Show graphs of how well $f(x)$ is approximated by overlaying the series over $f(x)$ for various values of N (for example, $N = 5, 10, 20, 50$).

2. (Asymptotic Equipartition) Consider the alphabet {A,B} with probabilities p(A)=0.27 and p(B)=0.73. Verify that the entropy of this distribution is $H = 0.841464$. Now consider all 1048576 sequences of length 20 generated out of this alphabet. Choose a small ϵ, say 0.001, and find the number of sequences that have probabulities between $2^{-20*(H+\epsilon)}$ and $2^{-20*(H-\epsilon)}$ (when the sequences are generated i.i.d). How many bits would be necessary to encode for any one of these sequences assuming that you have a simple index table for these sequences? Now compute the sum of the probabilities of these sequences. Assuming that you use 20 bits to encode for the remaining sequences, how many bits *on average* would be necessary to encode a string of length 20?

3. Code and test a feed forward net of sigmoidal nodes with two input units, ten hidden units and one output unit that learns the concept of a circle in 2D space. The concept is: $\langle x, y \rangle$ is labeled "+" if $(x - a)^2 + (y - b^2) < r^2$ and is labeled "-" otherwise. Set $a = 2, b = 3, r = 1$.

   Generate 100 random samples uniformly distributed from a cell in $\mathbb{R}^2$ to train the network using error backpropagation and 100 random samples to test it. Repeat the procedure multiple times and with multiple initial weights. Report the changing accuracy and the hyperplanes corresponding to the hidden nodes.

4. Get 20 images from the internet and reduce them to black and white dithered images of size 50×50 pixels. Now train a Hopfield type recurrent net to retrieve these images. Finally, initialize the net with noisy versions of each of these images (randomly inverted pixels or cropped versions of the image) and see whether the net converges to the original images.

   Images can be displayed easily in pbm format. For example, copy the next few lines into a file called test.pbm and use your favourite viewer (irfanview, xv, eog) to view it.

```
P1
# Width, height of image. Pixel values follow: < 70 chars per line
10 20
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 0 1 1 0 0
1 1 0 0 1 1 0 0 1 1
```