Mid-Term Exam

Artificial Neural Networks & Computational Neuroscience
Seoul National University

Thursday, October 26, 2017

Answer the following 5 questions. Use no more space than one page for each question. Attempt to address the general theme of each problem using the subquestions as guidelines, rather than give fragmented answers to each subquestion.

1. (**30 points**) Consider a linear neuron with Hebbian adaptation. How is the output y of the neuron is computed from m inputs x_i, $i = 1,\ldots, m$? The synaptic weights of the neuron can be trained by the Hebbian learning rule:

   $$w_i(n+1) = w_i(n) + \eta y(n) x_i'(n)$$

   $$x_i'(n) = x_i(n) - y(n) w_i(n)$$

   This results in the maximum eigenfilter. Explain the relationship of this neuron (i.e. maximum eigenfilter) and PCA. Explain why we need to have the $x_i'(n)$ term. What problem do we have if we use $w_i(n)$ instead of $w_i'(n)$? Extend the learning rule to a feedforward network consisting of m inputs and l outputs ($l < m$), resulting in the Generalized Hebbian Algorithm (GHA). Show that (or explain why) GHA is computing PCA.

2. (20 points) Consider the Kohonen's self-organizing map (SOM). Describe the learning procedure. Give the weight update rule, i.e. learning rule, of SOM? Why is this algorithm called winner-take-all? SOM can be viewed as density estimation, dimension reduction, and clustering. In each of these views, explain the effects of the followings: lattice size, neighborhood size, and learning rate. Is it possible to adapt the neighborhood size as a function of time? How can it be done, and why?

3. (20 points) Give the definition and explain its meaning of information, entropy, conditional entropy, and mutual information, respectively. What is Kullback-Leibler (KL) divergence and how is it related to mutual information? Give three examples of application of the mutual information in machine learning or computational neurobiology. Give an application example of the KL divergence

and explain it.

4. (**30 points**) Describe the architecture (activation functions of neurons and their connectivity) of the restricted Boltzmann machine (RBM). Derive the learning rule of the RBM (or give a sketch of deriving the learning rule). Describe the procedure for learning an RBM from a training dataset. Why do we need to use a Monte Carlo method like Metropolis, simulated annealing or Gibbs sampling to train the RBM? Explain the similarities and differences of the above three methods in their procedures, properties, and applications.

5. (20 points) Describe the architecture (activation functions of neurons and their connectivity) of a deep belief network (DBN). Explain how we can train the deep structure of DBN without the vanishing gradient problem. What is the vanishing gradient problem? Why does it happen? Give an application example of DBN that cannot (or hardly) be demonstrated by using multilayer perceptrons. What properties of the DBN and the multilayer perceptron cause their distinguishing capabilities?

(Total 120 points)