CS671 - Deep Learning and its Applications

Course Instructor : Aditya Nigam (Session : Feb-May 2017)

Assignment-02

Submission Date : March 22, 2017

Note:

1. Answer all questions. Maximum score is (345 + 5 bonus) points.
2. Make sure you clearly identify the question number and your final answer, and solve all parts of a question together. Show all your working and clearly mark all relevant axes and intercepts in plots.
3. You are free to make any reasonable assumption that you may need to logically answer a question.
4. Code has to be well commented and as general as possible.
5. You are expected to submit a make file to compile your code and a README file containing how to run your codes.
6. Some questions are theoretical and some are coding.
7. You are allowed to submit this assignment in group of two. Submit one single zip file that contain your theory part done in latex (submit full folder that we can compile again) and the code folder with make file, README and binary of all coding questions named. Make sure that your Makefile with by default can create all the required executables.

---

1. **Theory : Basic Probability**

   (a) There are 3 boxes A, B, C containing marbles of different colours. Their distribution is shown below:

	Green	Red	Blue
A	3	4	5
B	1	2	0
C	2	2	3

   i. If the probabilities of choosing a box are: A - 0.60, B - 0.30, C - 0.10 then what is the probability of choosing a green marble (assuming each marble can be chosen with the same probability)?

   ii. If the selected marble is actually red then what is the probability it was in box C?

   (b) You keep your papers in 3 different files. You are searching for a paper that is equally likely to be in any one of the 3 files. Let p_i be the probability that if in fact the paper is in file i and you quickly look through it you will find the paper. Suppose you quickly look through file 1 and do not find the paper. What is the probability that the paper is in file 1?

   [(3,5),7=15]

2. **Theory : Properties of Normal Distribution**

Verify the following properties of the Gaussian/normal distribution:

(a) $\int_{-\infty}^{\infty} N(x|\mu,\sigma)dx = 1$.

(b) $\mathcal{E}[x] = \mu$, $\mathcal{E}[x^2] = \mu^2 + \sigma^2$

(c) Show that the mode of the Gaussian is μ.

(d) If we draw n *i.i.d.* samples from a Gausssian distribution the probability of getting that set of n values is: $p(\{x_1,\ldots,x_n\}|\mu,\sigma) = \prod_1^n N(x_i|\mu,\sigma)$ We can treat this as a function of μ and σ and in this context it is called the *likelihood function.* We can try to maximize the likelihood function by adjusting the values of μ,σ. Using the log of the likelihood function show that:
i) $\mu_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i$
ii) $\sigma^2_{ML} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{ML})^2$
iii) $\mathcal{E}[\mu_{ML}] = \mu$
iv) $\mathcal{E}[\sigma^2_{ML}] = \frac{(n-1)}{n}\sigma^2$

(e) Let $\mathbf{x}$ be described by a multivariate Gaussian of d-dimensions show that if $\mathbf{x}$ is linearly transformed, $\mathbf{y}$=$\mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a d x d matrix, then $\mathbf{y}$ is also described by a Gaussian.

[4,8,4,4x5,4=40]

3. **Coding : Random sampling and Discriminant Function (Multi-variate Gaussian Distribution)**

(a) Write a function/procedure that takes $\overline{\mu}$, $\overline{\sigma}$, and d and generates a random sample value from the multi-variate Gaussian distribution $\mathcal{N}(\overline{\mu},\overline{\sigma})$ in d-dimensions. Using this function/procedure define a function that takes 4 arguments (the 3 above plus a positive integer N) that generates N random samples from the above multi-variate Gaussian distribution in d-dimensions.

(b) For a given multi-variate Gaussian distribution $\mathcal{N}(\overline{\mu},\overline{\sigma})$ write a function that given $\overline{x}$ computes the discriminant function $g_i(\overline{x})$ using the ln form discussed in class:
$g_i(\overline{x}) = -\frac{1}{2}(\overline{x}-\overline{\mu}_i)^T\overline{\sigma}^{-1}(\overline{x}-\overline{\mu}) - \frac{d}{2}ln(2\pi) - \frac{1}{2}ln|\sigma_i| + ln(P(C_i))$.

[20,10=30]

4. **Coding : Bayesian Decision Boundary**

Consider the following two data generators for a two class situation:

(a) The data in each class is generated from a bivariate Gaussian distribution with uncorrelated components and different means.

(b) The data in each class is generated from a mixture of a small number (5-10) of low variance Gaussian distributions whose means themselves are also distributed as a Gaussian.

Generate 100 points for each class in the two scenarios above and then calculate the Bayes decision boundaries in each case. Use the code written for the first problem. Choose the $\overline{\mu}_i$'s and $\overline{\sigma}_i$ suitably.

[25, 35=60]

5. **Theory : Multi-Class Classification**

One strategy for multi-class classification (C classes) is a pairwise strategy requiring ${}^{C}C_2$ binary classifiers - hyperplanes. Argue that the decision regions for such a classifier need not be convex and construct a non-trivial example of a classifier where at least one decision region is not convex. [20]

6. **Theory : One Versus Rest**

A linear machine classifies as follows:

Assign $\overline{x}$ to C_i if $g_i(\overline{x}) > g_j(\overline{x}),\ j \neq i$, and if such an i does not exist classify as *undefined*.

Another way to classify is to use *one-versus-rest* where a hyper-plane separates the class C_i from all the rest. Show that if samples can be classified using *one-versus-rest* then they can be classified by a linear machine but not the reverse. [20]

7. **Theory : Linear Separability**

The convex hull of vectors $\overline{x}_1, \ldots, \overline{x}_n$ is defined as the set: $\{\bar{(}x)|\sum_{i=1}^{n} \alpha_i \overline{x}_i,\ \alpha_i \geq 0,\ \sum_{i=1}^{n} \alpha_i = 1$. Given two sets $X_1,\ X_2$ of vectors belonging to classes $C_1,\ C_2$ argue that either they can be classified by a linear machine or their convex hulls have a non-empty intersection. [20]

8. **Theory : Convergence of Perceptron**

Supposing the perceptron update rule is:

$$\overline{a}(k+1) = \overline{a}(k) + \eta(k)\overline{y}^k,\ \text{iff } \overline{a}^T(k)\overline{y}^k \leq b$$

Assuming $0 < \eta_1 \leq \eta(k) \leq \eta_2 < \infty$ prove that the update process will converge. [20]

9. **Theory : Widrow-Hoff Convergence**

The Widrow-Hoff or LMS update rule is:

$\overline{a}(k+1) = \overline{a}(k) + \eta(k)(b(k) - \overline{a}^T(k)\overline{y}^k)\overline{y}^k$.

If $\eta(k) = \eta/k$ show that the update process converges to an $\overline{a}$ satisfying $Y^\dagger(Y\overline{a} - \bar{b}) = 0$ [20]

10. **Coding : Single and Batched Update Convergence Study of Perceptron**

Study the convergence behaviour (i.e. the number of iterations) of the perceptron algorithm in the following cases:

(a) Single sample with different η values: $0 < \eta < 1,\ \eta = 1,\ \eta \gg 1,\ \eta(k) = \frac{\eta}{k}$.

(b) Batched update with the η variation same as above.

Generate the three-dimensional two class distributions using $p(\overline{x}|C_1) \approx N(\overline{0}, I)$, $p(\overline{x}|C_2) \approx N(\overline{\mu}, I),\ \mu^T = [4.5, 4.5, 4.5]$

Check that the generated points are separable. Do not generate too many points, $20 - 30$ should be enough. [50]

11. **Coding : Single and Batched Update Convergence Study of Perceptron using Relaxation Update**

Using single sample and batch updates study the convergence behaviour of the relaxation update process for 3 different η values, $0 < \eta < 2$. The update rules are:

$$\overline{a}(k+1) = \overline{a}(k) + \eta(k)\frac{(\overline{b} - \overline{a}^T(k)\overline{y}^k)}{\|\overline{y}^k\|^2}\overline{y}^k \quad \text{single sample}$$

$$\overline{a}(k+1) = \overline{a}(k) + \eta(k)\sum_{\overline{y}^k \in \mathcal{Y}}\frac{(\overline{b} - \overline{a}^T(k)\overline{y}^k)}{\|\overline{y}^k\|^2}\overline{y}^k \quad \text{batch update}$$

Use the same data set as in the first question. Use fixed and η as a function of k, say η/k.

[50]