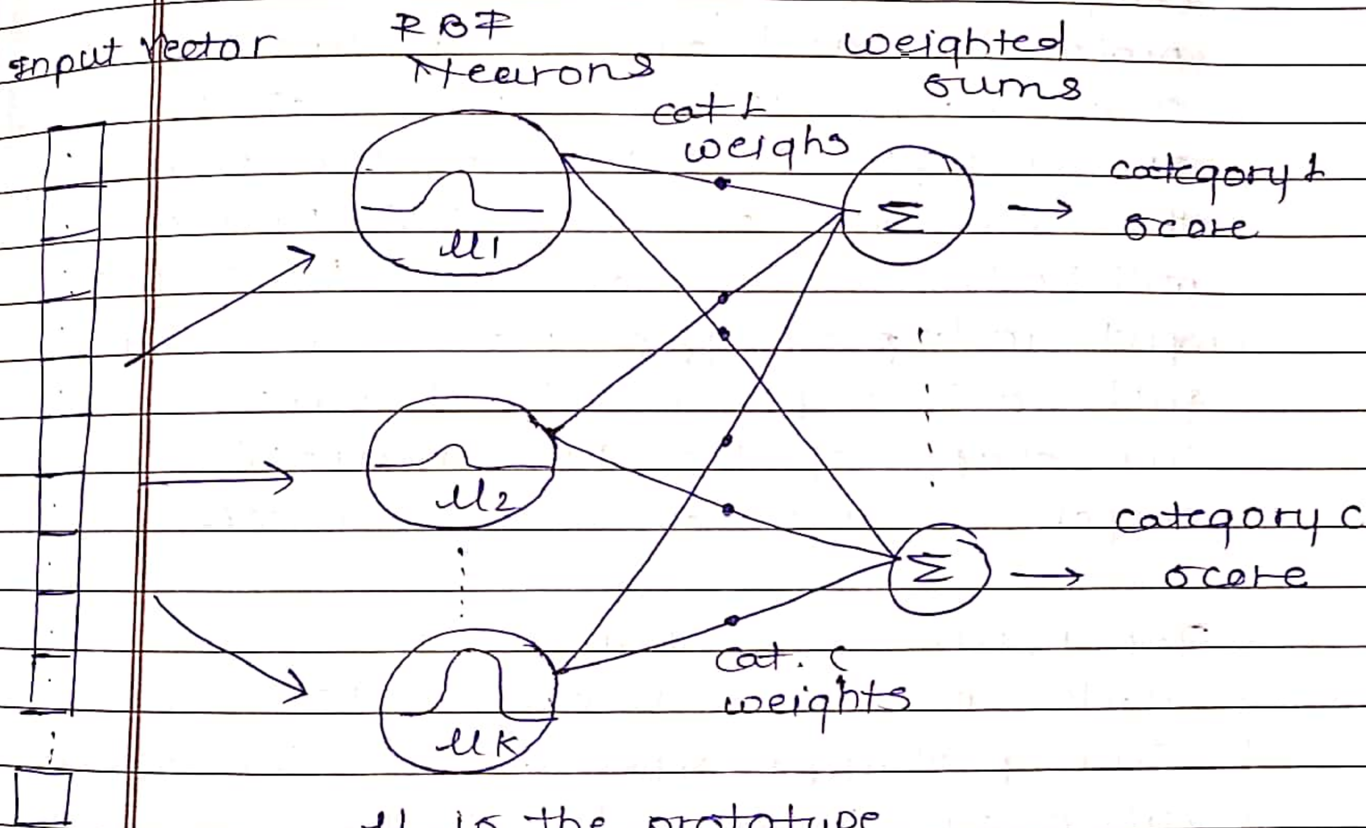
Assignment no-3

PAGE NO.:
DATE: / /

1. Explain the radial basis funⁿ neutral n/w with diagrammatic representⁿ and equation in detail.

→ A radial basis funⁿ n/w (RBFN) is particular type of neural n/w. RBFN consists of an input vector, a layer of RBF neurons, and an o/p layer with one node per category or class of data.



μ is the prototype to compare against

eklavyā PAGE NO.: DATE: / /

The Input Vector

The input vector is the n-dimensional vector that you are trying to classify. The entire input vector is shown to each of the RBF neurons.

The RBF Neurons-

Each RBF neuron stores a "prototype" vector which is just one of the vectors from the training set. Each RBF neuron compares the input vector to its prototype, and outputs a value betn 0 and 1 which is a measure of similarity. If the input is equal to the prototype, then the o/p of that RBF neuron will be 1. As the distance betn the input and prototype grows, the response falls off exponentially towards 0. The shape of the RBF neuron's response ~~falls of~~ is bell curve.

The neuron's response value is called its "activation" value.

The prototype vector is also often called the neuron's is "center", since, it's the value at the center of the bell curve.

Scanned with CamScanner

PAGE NO.:
DATE: / /

The output nodes

The output of the n/w consists of a set of nodes, one per category that we are trying to classify. Each output node computes a sort of score for the associated category. Typically, a classification decision is made by assigning the input to the category with the highest score.

$$\varphi(x) = e^{-\beta \| x - \mu \|^2}$$

In the gaussian distr^n, mu refers to the mean of the distr^n, here it is the prototype vector which is at the center of the bell curve.

Scanned with CamScanner

2] Describe Markov chain monte carlo method with help of real time instance.

→ Markov chain Monte Carlo (MCMC) methods comprise a class of algorithms for sampling from a probability distribution.
By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain.

Monte carlo is a technique for randomly sampling a probability distribution and approximating a desired quantity.

Markov chain is a systematic method for generating a sequence of random variables where the current value is probabilistically dependent on the value of the prior variable. Specifically, selecting the next variable is only dependent upon the last variable in the chain.

Scanned with CamScanner

eklavya
PAGE NO.:
DATE: / /

Example—

Markov chains applied to PageRank.

Google computes a value called PageRank for each entry in its search engine, Page Rank is part of what determines which web page is displayed in which rank when you hit the search button.

For some page p, the PageRank is defined as

$$PR(p) = \sum_{v \in B_p} \frac{PR(v)}{L(v)}$$

where B_p is the set of webpages that link to p and $L(v)$ is the no. of pages that p links to. Note that PageRank is essentially a recurrence relaⁿ, but the important thing is that the PageRank of p assumes that the user randomly sufer is, $v \in B_p$ i.e. random, it depends only on the previous page.

$$P(X_{n+1}=x \mid X_1=x_1, x_2, \ldots, X_n) = P(X_{n+1}=x \mid X_n=x_n)$$

Google is able to estimate the PageRank values using MCMC. ~~After~~

Scanned with CamScanner

Example- Random walks applied to sampling the web web.

To do research, we usually want a random example.

But how on the web, do we get a random sample of web pages?

1. Generate random IP address and find that most IP addresses are not associated with web server some IP address represent multiple web servers. FAIL.

2. generate random URLs. How would we even start to do that? FAIL.

3. Use a Monte Carlo method.

Rusmevichientong et al. (2001) proposed this method for sampling web pages, and Gjoka el. al (2009) used this method for selecting a random example of Facebook users for an analysis.

Scanned with CamScanner

PAGE NO.:
DATE: / /

~~1. The proble~~

The problem is formulated as follows:

1. Perform the following series of steps:

a] Pick a web page at random. We could google the word the and pick the 1st result.

b] Pick a link on the web page at random and move to the new page.

c] Repeat the previous step T times

d] Perform step (b) another k times. Let X1, ... Xk be the coll^n of web pages visited during this part of the crawl.

e] For each unique page p in Xk; i.e.,
(i) Pick a random link and move to the new page associated with that link.
(ii) Repeat the previous step M times yielding a new coll^n of pages Zp1, Zp2, ... ZpM
(iii) Compute the no. of times we visited page p: $\pi d(p) = pM$ $r=1$ $(Zpr=p)$ M where $1(Zpr=p)=1$

PAGE NO.:

DATE: / /

if $zpr = p \in 180$.

(x) Accept page p into the final sample with probability $\beta \pi_d(p)$ where $0 < \beta < \min_{p=1,\dots,k} \pi_b(x_p)$.

This is quite a complicated algorithm! Notice again that this process is a Markov chain ~~bran~~ because the page we visit ~~the~~ next depends only on the page we are currently at and no other page.

This is an example of Metropolis-Hastings random walk. While we will discuss this algorithm in class, we will not do anything remotely this complicated! All of this convergence talk may remind you of real analysis.

No need for real analysis, however, if you like analysis and MCMC, you could get carried away with theoretical research in this field.