Bayesian Decisions on Differentially Fed Hyperplanes

Bangalore University
Manjunath.R, Dr K.S.Gurumurthy
Dept of EC & CSE, UVCE, Bangalore, INDIA
manju_r_99@yahoo.com

Abstract: Generally the neural networks employing Bayesian decision do not output one simple hypothesis, but a manifold of probability distributions. This throws out the bayes posterior coefficients as a large number of classifiers. Here a novel method based on differential feedback is explored to merge these classifiers. The experimental results confirm affine transportation of these classifiers. Also, it has been shown that the differentially fed Artificial Neural Networks (ANNs) learn in much the same way as Bayesian learning and are hence resistant to over fitting

1 INTRODUCTION

In Information geometry, a family of probability distributions are made use to compute quantities related to pdf including mutual interactions. It was used to study multilayer perceptrons [1]. A family of distributions called exponential family has the pdf

$$P\ (y,\theta)=\exp\{\Sigma\theta_i k_i(y)-\ \psi(\theta)\} \qquad (1)$$

Where θ is the cosystem, $k= k_i(y)$ are adequate functions of y, ψ the offset. The exponential family forms $\alpha=\pm1$ flat manifold. For this Riemann-christoffel curvature vanishes identically. This is non Euclidean space. For this manifold, there exists $\alpha=\pm1$ affine cosystem. This is because log of pdf is linear in θ . For any two distributions p (y) and q (y) the geodesic connecting them is given by

$$\log (p\ (y,\ t))=(1-t)\log p(y)+t \log q(y)-\ \psi(t) \quad (2)$$

A family of Divergence measures namely α-divergence is associated with the manifold of pdf [2]. The $\alpha=-1$ divergence is known as Kullback divergence. These divergence functions give a Reimannian metric to the manifold of pdf.

The networks are represented by a set of parameters called weights $\theta=w_{ij}$ .A family of networks parameterized by θ forms a manifold, θ playing the role of a cosystem.. In this paper, it is shown that they form hyperplanes for different choices of feedback. The amount of feedback is proportional to Kullback divergence distance i.e., planes are Kullback distance apart. The kullback divergence of pdf p from pdf q is given by

$$p*\log(p/q) \quad (3)$$

In this paper, different orders of differential feedback form a manifold of hyperplanes and are related to manifolds of pdfs. The distance between them and the trajectory of a point on these planes is explored. In section 2 differentially fed ANNs are introduced. In section 3 the superposition of hyperplanes is explained. The simulation results are given in section 4 section 5 concludes

2. FORMALISM OF DIFFERENTIALLY FED ANN

The output y of a neural network except for the nonlinearities can be written as

$$y=\Sigma wixi. \qquad (4)$$

Where xi are the inputs wi, the corresponding weights. The thing to be noted is weight cannot span the entire input space, whatever may be the training mode. Again the linearity of the output (1) may be viewed as a particular case of ARMA

$$y(n+1)=b0y(n)+b1y(n-1)+\ldots..+a0xn+\ldots \quad (5)$$

Where b0.. and a0.. are constants. The auto regressive terms b0…bn may be realized using inherent differential feedback [3]. With differential feedback it has been found out [3] that the no of iterations required for training is reduced as shown in the table I.XOR gate is considered for simulation. Gaussian distributed random input with seed value 1000 is taken as input. With I order different feedback, the output may be written as:

$$\Sigma Wixi+b1y1 \qquad (6)$$

y1 being the I order differential. This equation once again represents a plane parallel to Σwixi. Thus the set of differentially fed ANNs form a manifold of parallel planes, with ∞ order feedback being the plane with zero error.

Table 1. Performance with feedback

Order of differential	Square error	Iterations
No feedback	18	1156
I order	18	578
II order	18	289

Table 2..Performance with II order feedback

Order of differential	Square error	Iterations
II order Feedback	18	578
Equivalent Output	18	578

Also, simulation results of table 2 show that two terms of II order differential feedback i.e., y2-y1 and y1-y0 can be replaced by a single equivalent plane represented by

Weq=(w1*iextra+w2*iextra1)/y0 (7)

In II order differential feedback system, the two differential terms can be replaced by a single term. Extending this principle, the ∞ terms of ∞ order differential feedback can be replaced by a single term.

This is termed as eigen plane which is the practical way of generating lowest error. Now the differential feedback becomes

$dy/dt+d^2y/dt^2+\ldots$ (8)

Taking Z transform, & then the inverse,

yeq = IZT{Y (z)/(1-z)} (9)

2.1. Information geometry of differential feedback

For less error, the plane spanned by the weight vectors should be as close as possible to the eigen plane. When I order differential feedback is given, the new plane is given by

ynew =∑wixi+a*yold (10)

Which is a parallel plane. To start with, set Yold=0.So, y = ∑wixi. Since error varies asymptotically with order, the gap between parallel planes decreases and the infinite order Plane coincides with the eigen plane. If still more feedback is given, error increases further as shown in fig.2

To show that the entropy is minimum on the eigen plane, consider the exponential family as given in [4] . The error may be assumed to be Gaussian distributed. In Gaussian distribution with Zero mean, the pdf can be written as

$p(x)=\exp((\text{energy of } x)/\sigma^2)$ (11) .

The error energy of a plane x may be written as $(x-d)^2$. (x-d) being the distance of x from eigen plane d (or the actual value). The entropy of such a distribution takes minimum value when (x-d)=0. i.e. entropy is min when the plane reaches the eigen plane The Natural learning algorithm is given by [5]

$\theta(t+1)=\theta(t)-\eta G^{-1}$ (12)

i.e.,newplane=oldplane+deviation This shows that the repeated learning in gradient descent algorithm shifts the planes towards the eigen plane in the same way the diff feed back will do ,but fails to reach it because eigen plane does not belong to the space spanned by inputs alone.

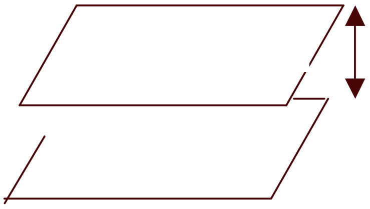

Figure.1.Differential feedback planes

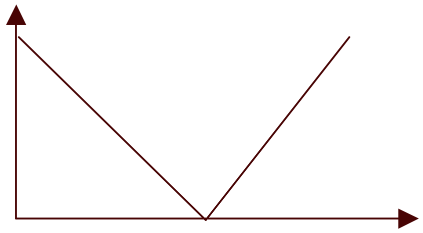

Figure2. Feedback v/s error

3. SUPERPOSITION OF HYPERPLANES

Bayesian learning [6] is a pdf over hypothesis (parameterized) space, expressing degree of belief in a specific hypothesis. A neural network trained with Bayesian learning algorithm outputs entire distribution of probabilities over hypothesis set rather than a single hypothesis. Such a distribution is Bayes' aposterior and depends up on the training data and on the prior distribution. This dependency on the aprior distributions make it a memory system where previous outputs take a role in deciding the present output. In the present context each hypothesis corresponds to one hyper plane i.e., different orders of feedback. Alternative perspective is that each hyperplane may be taught of as a classifier with an associated probability density function. Degree of belief is 0 for no feedback and increases towards 1 for infinite feedback or when all classifiers merge. In such a classifier the actual output may be thought of as superposition of beliefs [7] i.e., addition of different feedbacks. As the order of differential feedback increases, the number of estimators considered in the sum increases by the same amount and hence the degree of belief moves towards 1.The addition is not simple but weighted by belief or pdf. In [1]. it has been proved that the n number of differential terms with a feedback of degree n may be replaced by a single nth degree differential term. It follows that, a single equivalent or effective distribution will be thrown out with nth degree differential feedback. It may be seen that equation. 13 runs in much the same way as that of equation.7.Finally we arrive at the eigen plane. I.e. the superposed effect of all classifiers is the eigen plane. This gives

P0*no feedback+p1*I order differential feedback=p2*II order differential feed back (13.a)

P1* I ordered differential+…infinite order =Eigen plane (13.b)

I.e. weighted sum of different ordered differentials.

P1*distance between I order and nofeedback+p2*II order and no feedback+…=1*distance between no feedback and Eigen plane (13.c)

The equations show that the learning algorithms with differential feedback do indeed resemble Bayesian learning algorithms and are hence resistant to over fitting [7]. It may be attributed to the hidden Auto regression associated with differential feedback

3.1. Resistance for over training

The posterior has two components-a data independent Gaussian prior part and a data dependent term. Logically, the Gaussian part may be attributed to the previous or differential terms of the output since the weighted sum of any probability distribution function in general turns towards Gaussian, by central limit theorem. Such a Gaussian classifier is known to resistant to over fitting

3.2..Bayesian learning

The result of Bayesian learning is a pdf over the hypothesis space each expressing the degree of belief in a specific hypothesis as an approximation to the target function. The aprior distribution P(λ) generally encodes some prior knowledge. With the arrival of data pattern D the aprior distribution gets updated using Baye's rule as P (λ|D) ∝ P (D|λ) P (λ).Taking Logarithm both sides, we get

Log (P (λ|D)) ∝ log (P (D|λ))+log (P (λ)) (14)

The equation has two terms-one current data dependent term and one data independent term where the prior or previous outputs (Gaussian as a result of superposition) are considered.The posterior distribution so obtained hence encodes information coming from the training set and prior knowledge.

Consider the example of II order feedback which makes use of two previous or priori terms

P (λ1) and P (λ2). With this the equation may be rewritten as P (λ|D)= P (D|λ1)* P (λ1)+ P (D|λ2)* P (λ2)Which leads to the equation

p2*II order differential feed back =P0*no feedback+p1*I order differential feedback (15)

This is analogous to the famous equation

$$P\ (y|x,\ D)= \int_{\lambda} f(x,\lambda)\,p(\lambda \mid D)\,dP(\lambda) \qquad (16)$$

in probability space.As can be seen here, the probabilities are proportional to the weights. The equation tries to expand the (k+1) th order differential feedback plane with0,1..K th order differential feedback planes. The weighing factors may be taught of as the projection or dot product of the hyper plane over lower order hyper planes.

3.3.Hilbert space

The set of probabilities form Hilbert space H={z:Λ→ℜ| such that $\int_{\lambda\in\Lambda} z(\lambda^2)dp(\lambda) < \infty$}

with the inner product

$$<z1.z2>= \int_{\lambda\in\Lambda} z1(\lambda)z2(\lambda)dp(\lambda) \qquad (17)$$

Since the same constraints are also satisfied by the hyper planes, they form Hilbert space.The output of a neural network is subjected to nonlinearity or fair quantization. Let sk be the number of points or planes which go wrong because of this nonlinear round off. The fractional error ek is defined as sk/l , l being the number of hyperplanes considered. Because of the uniform nature of this error, the probability of this error equal to r/l is

$\dfrac{\ell!}{2^l (l\varepsilon)!(\ell - \ell\varepsilon)!}$ which gives average probability of realizing different patterns of r errors.Now mapping the hypothesis space to error shells or differential feedback hyperplanes, We get

$$\sum_{j\in J} p_j \varepsilon_j = \frac{1}{2^l}\sum_{r=0}^{l}\left(\frac{r}{l}\right) * B \qquad (18)$$

Where B=(r,l). Hence the error in classification has to vary asymptotically as a power of 2 with increase in the order of differential feedback. This is indeed the case as given in table I. But for the nonlinearity, any hyperplane or the k+1 th degree feedback happens to be a linear weighted sum of the k hyperplanes found in H. This threshholding makes the output a subset C (H) the convex hull of H rather than H itself. It can be shown that the bias term or the datum or reference depends just up on the instantaneous data and independent of the feedback inputs. I.e. it remains the same for all orders of the feedback. Here also, P (D|λ1) and P (D|λ2) are the same and independent of λ or the feedback but depends only up on D the data. Hence the above equation may be rewritten as

P (λ|D)=P (D){P (λ1)+ P (λ2)} (19)

I.e., Output without feedback or the bias term*Gaussian like pdf. especially with higher Orders of the feedback

.

4. SIMULATION

The differentially fed Artificial neural networks are made to learn the psd of random data .The Normal distributed data is generated using Matlab.The error after learning and the differentials of the error are stored The probability distribution of each of them is computed using Parzen equation p (x)= $1/\sqrt{2\pi\sigma}$ exp (-($\dfrac{(mean - x)^2}{\sigma^2}$)).The weighted sum of the zero th order feedback and I order feedback data with their corresponding pdfs is found identical to the weighted second ordered differential feedback with the corresponding pdf as given in the equation In fig.3 signals of I and zero order weighed with pdf and weighed II order signal are shown.

5. CONCLUSIONS

From the simulation results it is clear that the classifier represented by a certain hyper pane is the weighted sum of the hyper planes o classifiers below. This way, ideal classifier is the weighted sum of all the classifiers.

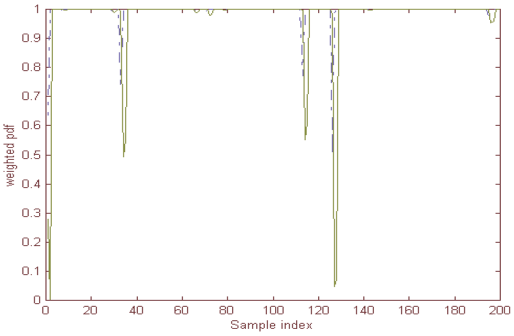

Figure.3.signals of I and zero order weighed with pdf and weighed II order signal

Legend: - Weighted second ordered feedback. -. Sum of weighted zeroth and first ordered FB

REFERENCES

1. S.Amari, 1995, Information Geometry of the EM and em algorithms for neural networks, Neural networks, 8, No.9

2. 7.S.Amari and H.Nagaoka, 2000, Methods of information geometry, AMS and Oxford University press.

3. Manjunath.R and K.S.Gurumurthy, oct2002, System design using differentially fed Artificial Neural networks, TENCON'02

4. S.Amari, 1995, Information Geometry of the EM and em algorithms for neural networks, Neural networks, 8,No.9

5 Aarts, E.H.L. and Korst, J.H.M, 1989,.Simulated Annealing and Boltzmann Machines, Chichester: Wiley.

6.D.J.C.Mac Kay, 1992, A practical Bayesian framework for back propagation networks Neural computation 4:448-472

7. Radford Neal, 1996, Bayesian learning in neural networks springer verlag