Estimation of Conditional Densities: A Comparison of Neural Network Approaches

R.Neuneier‡, F.Hergert‡, W.Finnoff*, D.Ormoneit†
‡ Siemens AG, Corporate Research and Development
Otto-Hahn-Ring 6, D 81730 Muenchen, Germany
Email: Ralph.Neuneier@zfe.siemens.de
† Dept. of Computer Science, TUM, D-80290 Munich
*Prediction Company, 234 Griffin St., Santa Fe, NM 8750

1 Introduction and the Task

In recent years, neural networks have been successfully used to attack a wide variety of difficult nonlinear regression and classification tasks and their effectiveness, particularly when the dimension of the problem measured in the number of variables involved, has been widely documented (Finnoff 1993).

To adequately address certain issues, coming from the field of stochastic control or portfolio optimization, more complete information about the distribution of the data is required than can be provided by a simple regression function. This information can only be obtained by an estimator of the complete probability density of the data. Although there is a wealth of literature from the statistics community dealing with this subject, close inspection reveals that most is either of purely theoretical nature or only considers very low dimensional problems or tasks with a relatively simple probablistic structure.

This area has recently attracted growing interest in the neural network community (White, 1992, Mackey, 1991 and Nowlan, 1991), still many issues remain to be addressed as to the effectiveness of competing techniques when applied to difficult problems.

The task is to estimate the conditional densities $p(y|x)$ given empirical data points (y, x) produced by a time series, for example, we wish to know the probability for a (future) value y having observed the value x. In this paper we will compare several architectures and learning algorithms using both an artificially generated and a real world data set.

The first example is a Monte Carlo simulation with known probability density which can be viewed as a bounded Brownian process. A particle is allowed to move freely in a double well potential (Ormoneit, 1993) with momentum and a friction term, perturbed by small normally distributed random shocks, presenting the price dynamics of a market with two long term equilibria perturbed by small information "shocks" in which some of the market participants use trend following strategies. The task is to predict the position of the particle 25 time steps ahead.

The second example is a real world time series consisting of US$-SFR exchange rates sampled every 10 minutes. Using information of the past as input the task is to produce the conditional probability distribution of the exchange rate 30 minutes into the future. Estimates of that kind will be used in a system under development for making sell or buy decisions.

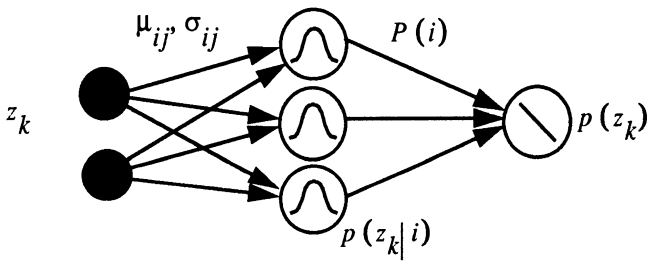
M. Marinaro et al. (eds.), *ICANN '94*

2 Conditional Density Estimation with NN

To estimate probability densities we restrict ourselves to the function class of linear mixtures of probability densities $p(z) = \sum_{i=1}^{c} P(i)p(z|i)$ where the component densities $p(z|i)$ are multi-dimensional Gaussians with weighting factors $P(i)$ satisfying $P(i) \geq 0$, $\sum_i P(i) = 1$. From the perspective of a data generating process the weighting factors $P(i)$ can be interpreted as a priori probabilities for a data point belonging to the component distribution i, and $p(z|i)$ is a measure of belief that data point z was produced by component i.

There are two basic approaches to find the conditional density. Either one can estimate the conditional density directly, or, first estimate the joint density and then compute the conditional density using the relation $p(y|x) = \frac{p(y,x)}{p(x)}$.

Gaussian Mixture Network and EM-Algorithm: For the Gaussian Mixture Network (GMN) we follow the latter approach, estimating the joint density. Linear mixtures of Gaussians are easily translated to GMNs, by letting each hidden unit i represent a component density $p(z|i)$ and summing up the activation values of these units in a linear output neuron with hidden-to-output weightings given by the $P(i)$s. Denoting by $z_k = (y_k, x_k)$ the set of training patterns the GMN will try to maximize the joint likelihood $L = \prod_k p(z_k)$. This is equivalent to minimizing the logarithm of the inverse joint likelihood, yielding the error function $E := -\log L = -\sum_k \log p(z_k)$, which is well suited for use in conjunction with the backpropagation algorithm (Tresp, 1993).



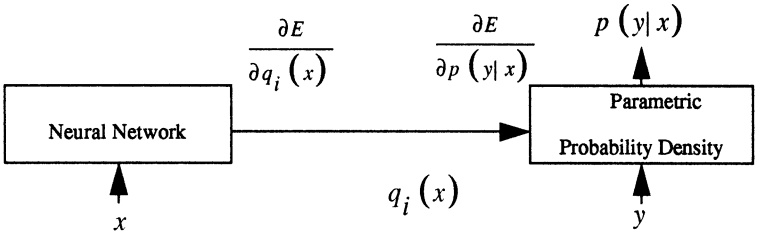
Instead of using the very general approach of gradient descent the Expectation Maximization (EM) Algorithm (Redner, 1984 and Duda, 1973) tries to exploit more specifically the properties of the likelihood function. This algorithm can be modified to become an on-line, incremental learning scheme (Ormoneit, 1993) which allows to integrate the algorithm easily into a usual neural network setting, in particular one can use 'standard' backpropagation to initialize the parameters and then continue fast learning with the EM-algorithm.

Probabilistic Neural Network: The Probabilistic Neural Network (PNN) provides a conceptually very simple way to estimate a probability density (Parzen, 1962 and Specht, 1990). The data points define the centers of the Gaussian component densities. Since all a priori probabilities are assumed to be equal the only parameters that remain to be adjusted are the variances of the Gaussians (exhaustive search!). In our examples the quality of the probability density estimate was very sensitive to the appropriate choice of the variance.

Conditional Density Estimation Network: The last two networks considered here estimate the conditional densities $p(y|x)$ directly. The networks try to maximize the conditional density likelihood $L^c = \prod_k p(y_k|x_k)$ for the empirical data points $z_k = (y_k, x_k)$, or equivalently minimize

$E^c := \sum_k E_k^c = -\sum_k \log p(y_k|x_k) = -\sum_k \log p(y_k, x_k) / \int p(y, x_k) dy.$

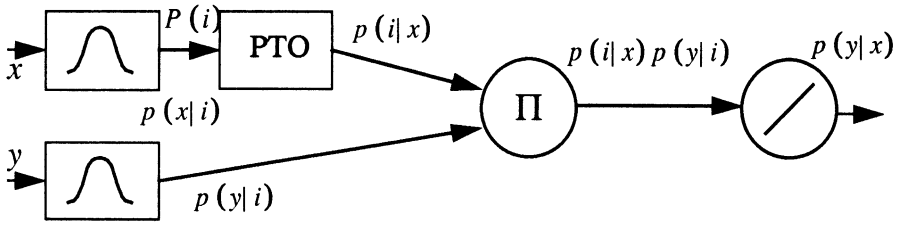
For a Conditional Density Estimation Network (CDEN) one assumes that the parametric conditional density $p(y|x)$ can be defined as $p(y|\theta(x))$, where $\theta(x) = \{q_1(x), \ldots, q_r(x)\}$ denotes a set of parameters functionally depending on x. The function $\theta(x)$ is realized as a network whose outputs are the parameters for the parameterized conditional density estimator.



During forward pass the network outputs are used as density parameters, while during error backpropagation the derivatives of the error function E_k^c with respect to the parameters are used to initialize the external error of the parameter estimation network (see schematic CDEN in figure above).

Distorted Probability Mixture Network: Finally we investigated the Distorted Probability Mixture Network (DPMN), which is another possibility to estimate the conditional density directly. The network is based on the assumption that we can model the densities with elementwise independent component densities (diagonal matrices for the variances of the Gaussian units). This allows to rewrite $p(y, x|i)$ as $p(y|i)p(x|i)$ (Tresp, 1993) yielding the following expression for the conditional density which can be directly implemented into a network topology (Ormoneit, 1993) as shown in figure below

$$p(y|x) = \frac{p(y,x)}{p(x)} = \frac{\sum_i p(y,x|i)P(i)}{\sum_l p(x|l)P(l)} = \sum_i p(y|i) \frac{p(x|i)P(i)}{\sum_l p(x|l)P(l)} = \sum_i p(y|i)p(i|x).$$



3 Results

The results for the Brownian process and the exchange rate are summerized in the table below. Using a validation set to detect overfitting and to stop the training process the performance of the resulting network was evaluated on a generalization set (for more details on this stopped training technique and related concepts see Finnoff, 1993). The first column of the table gives the network used. Several network topologies were explored for each experiment. The reported performance always refers to the best network found. The performance given in columns Example 1 resp. 2, represents the negative average logarithm of the conditional likelihood of the data points of the generalization

set – smaller values therefore denote better estimates. Note that values on different data sets are not directly comparable.

Network	Algorithm	Brownian	Exch. Rate
GM	Bp.	0.525	-0.04
GM	EM Bp.	0.413	-0.07
PNN	random search	0.556	—
CDE	Bp.	0.372	-0.32
DPM	Bp.	0.334	> 1

In the case of the Brownian motion the training set consisted of 3,000 data points, drawn from the 12,000 data points produced by a Monte Carlo simulation of the process. The 13-dimensional input vector encoded the increments during the last ten steps, the current position itself, and exponentially smoothed averages of the previous increments and positions. The target is the position of the process 25 time steps ahead.

In the second example the network used 20 inputs computed from the last six values of the exchange rate plus the value of the future exchange rate itself. Training, validation, and generalization set contained 3691, 1230, and 1610 data points, respectively. Due to the large size of the data set and the inferior performance of the PNN in example 1, no trials with that network were undertaken on the second task.

From our tests we conclude that the direct approaches to conditional density estimation (CDEN, DPMN) appears superior to the indirect methods (GMN, PNN). On the other hand the failure of the DPM-network on the second task indicates that the performance of alternative algorithms can be very domain dependent especially for problems with high dimensional and noisy data.

Acknowledgements: The authors are grateful to the influential ideas of Volker Tresp and the extensive discussions with him.

References:

• Duda R. O. and Hart P. E. (1973), Pattern Classification and Scene Analysis
• Finnoff W., Hergert F. and Zimmermann H. G. (1993), Improving Model Selection by Nonconvergent Methods, Neural Networks Vol. 6, pages 771-783
• MacKay D. J. C. (1991), Bayesean Modelling and Neural Networks, PhD-Thesis at California Institute of Technology, Pasadena
• Nowlan S. J. (1991), Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures, PhD-Thesis at School of Computer Science, Carnegie Mellon University, Pittsburgh
• Ormoneit D. (1993),Estimation of Probability Densities using Neural Networks, Master-Thesis: Dept. of Computer Science, TU Munich
• Parzen E. (1962), On Estimation of a Probability Density Function and Mode, Annals of Mathematical Statistics 33
• Redner R. A. and Walker H. F. (1984), Mixture Densities, Maximum Likelihood and the EM Algorithm, SIAM Review, 26
• Specht D. F. (1990), Probabilistic Neural Networks, Neural Networks 3
• Tresp V., Hollatz J. and Ahmad S. (1993), Network Structuring and Training Using Rule-Based Knowledge, Advances in NIPS 5
• White H. (1992), Parametrical Statistical Estimation with Artificial Neural Networks, Techreport University of California, San Diego