IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Mixtures of Experts Estimate A Posteriori Probabilities

Perry Moerland *

IDIAP–RR 97-07

September 97

published in
Proceedings of The International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 1997, 499–504

Dalle Molle Institute for Perceptive Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

* e-mail: Perry.Moerland@idiap.ch

IDIAP Research Report 97-07

Mixtures of Experts Estimate A Posteriori Probabilities

Perry Moerland

September 97

published in
Proceedings of The International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 1997, 499–504

Abstract. The mixtures of experts (ME) model offers a modular structure suitable for a divide-and-conquer approach to pattern recognition. It has a probabilistic interpretation in terms of a mixture model, which forms the basis for the error function associated with MEs. In this paper, it is shown that for classification problems the minimization of this ME error function leads to ME outputs estimating the a posteriori probabilities of class membership of the input vector.

1 Introduction

It is well-known that for artificial neural networks trained by minimizing sum-of-squares or cross-entropy error functions for a classification problem, the optimal network outputs approximate the a posteriori probabilities of class membership [2] [10]. This property is a very useful one, especially when the network outputs are to be used in a further decision-making stage (e.g. rejection thresholds) or integrated in other statistical pattern recognition methods (as in hybrid NN-HMMs).

Recently, a modular architecture of neural networks known as a *mixture of experts* (ME) [7][8] has attracted quite some attention. MEs are mixture models which attempt to solve problems using a divide-and-conquer strategy; that is, they learn to decompose complex problems in simpler subproblems. In particular, the *gating* network of a ME learns to partition the input space (in a soft way, so overlaps are possible) and attributes *expert* networks to these different regions. The divide-and-conquer approach has shown particularly useful in attributing experts to different regimes in piece-wise stationary time series [12], modeling discontinuities in the input-output mapping, and classification problems [4][11].

The ME error function is based on the interpretation of MEs as a mixture model [9] with conditional densities as mixture components (for the experts) and gating network outputs as mixing coefficients. This error function is in fact a generalization of the sum-of-squares and cross-entropy error functions which arise in the special case of a ME with only one expert network. The purpose of this paper is to show that at the global minimum of this general ME error function, the optimal ME outputs estimate a posteriori probabilities of class membership.

The paper is organized as follows. First the ME architecture is briefly described. This is followed by the description of the general framework of maximum likelihood for the derivation of error functions. In the case of a mixture of conditional densities this approach leads to the ME error function. The rest of the paper deals with the interpretation of the optimal ME outputs at the minimum of this error function for the case that the conditional density of each of the experts is either a multidimensional Gaussian or multinomial. In both cases it is shown that the optimal ME outputs estimate a posteriori Bayesian probabilities.

2 Mixtures of Experts

In this section the basic definitions of the mixture of experts model are given which will be used in the rest of the paper.

Figure 1 shows the architecture of a ME network, consisting of three expert networks and one gating network both having access to the input vector $\mathbf{x}$; the gating network has one output g_i per expert. The standard choices for gating and expert networks are generalized linear models [8] and multilayer perceptrons [12]. The output vector of a ME is the weighted (by the gating network outputs) mean of the expert outputs:

$$\mathbf{y}(\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x})\mathbf{y}_j(\mathbf{x}). \tag{1}$$

The gating network outputs $g_j(\mathbf{x})$ can be regarded as the probability that input $\mathbf{x}$ is attributed to expert j. In order to ensure this probabilistic interpretation, the activation function for the outputs of the gating network is chosen to be the soft-max function [3]:

$$g_j = \frac{\exp(z_j)}{\sum_{i=1}^{m} \exp(z_i)}, \tag{2}$$

where the z_i are the gating network outputs before thresholding. This soft-max function makes that the gating network outputs sum to unity and are non-negative; thus implementing the (soft) competition between the experts.

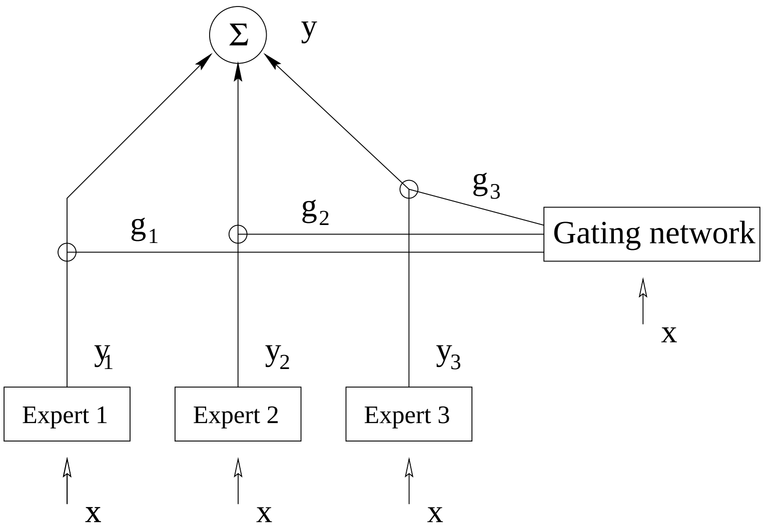

Figure 1: Architecture of a mixture of experts network.

A probabilistic interpretation of a ME can be given in the context of mixture models for conditional probability distributions (see section 6.4 in [1]):

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}), \tag{3}$$

where the ϕ_j represent the conditional densities of target vector $\mathbf{t}$ for expert j. The use of a soft-max function in the gating network and the fact that the ϕ_j are densities guarantee that the distribution is normalized: $\int p(\mathbf{t}|\mathbf{x})\,d\mathbf{t} = 1$.

As outlined in the next section this distribution forms the basis for the ME error function which can be optimized using gradient descent or the Expectation-Maximization (EM) algorithm [8].

3 Estimating Posterior Probabilities

A standard way to motivate error functions is from the principle of maximum likelihood of the (independently distributed) training data with input vectors $\mathbf{x}^n$ and target vectors $\mathbf{t}^n$: $\{\mathbf{x}^n, \mathbf{t}^n\}$ (see section 6.1 in [1]):

$$\mathcal{L} = \prod_n p(\mathbf{x}^n, \mathbf{t}^n) = \prod_n p(\mathbf{t}^n|\mathbf{x}^n)p(\mathbf{x}^n),$$

where dependence of $p(\mathbf{x}^n, \mathbf{t}^n)$ and $p(\mathbf{t}^n|\mathbf{x}^n)$ on the network parameters has been left implicit. A cost function is then obtained by taking the negative logarithm of the likelihood (and dropping the term $p(\mathbf{x}^n)$ which does not depend on the network parameters):

$$E = -\sum_n \ln p(\mathbf{t}^n|\mathbf{x}^n). \tag{4}$$

The most suitable choice for the conditional probability density depends on the problem. For regression problems a Gaussian noise model is often used (leading to the sum-of-squares error function); for classification problems with a 1-of-c coding scheme, a multinomial density is most suitable (leading to the cross-entropy error function). It is well-known that at the global minimum of these error functions when trained for classification problems, the optimal network outputs approximate a posteriori probabilities (independent of the network topology) [2] [6]. In this paper, it is shown that the minimization of the *ME error function* based on a *mixture* of conditional densities (3) also leads to network outputs that estimate a posteriori probabilities.

In its most general form this ME error function to be minimized is (substituting (3) in (4)):

$$E = -\sum_{n} \ln \sum_{j=1}^{m} g_j(\mathbf{x}^n)\phi_j(\mathbf{t}^n|\mathbf{x}^n),$$

the exact formulation of which depends on the choice for the conditional densities $\phi_j(\mathbf{t}^n|\mathbf{x}^n)$ of the experts.

In the limit of an infinite data set (to avoid bias and variance) the finite sum over patterns (divided by the size of the training data set) can be replaced with an integral:

$$E = -\int\int \ln\left(\sum_{j=1}^{m} g_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x})\right) p(\mathbf{t},\mathbf{x})\, d\mathbf{t} d\mathbf{x},$$

factoring the joint distribution:

$$E = -\int\int \ln\left(\sum_{j=1}^{m} g_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x})\right) p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\, d\mathbf{t} d\mathbf{x}.$$

The interpretation of the ME outputs when this error function is minimized, can be obtained by setting to zero the functional derivatives [5] of E with respect to the gating network outputs $z_j(\mathbf{x})$ and the expert network outputs $y_{jc}(\mathbf{x})$. The solution of these equations will then result in expressions for $g_j(\mathbf{x})$ and $\mathbf{y}_j(\mathbf{x})$ at the minimum of E (along the lines of section 6.1.3 of [1] for the sum-of-squares error function).

Defining:

$$E' = -\ln \sum_{j=1}^{m} g_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}), \tag{5}$$

we are then interested in the following two functional derivatives set to zero. For the gating network:

$$\frac{\delta E}{\delta z_j} = \int \left(\frac{\partial E'}{\partial z_j}\right) p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\, d\mathbf{t} \; = 0, \tag{6}$$

and for the expert network:

$$\frac{\delta E}{\delta y_{jc}} = \int \left(\frac{\partial E'}{\partial y_{jc}}\right) p(\mathbf{t}|\mathbf{x})p(\mathbf{x}) d\mathbf{t} = 0. \tag{7}$$

In section 6.4 of [1], the partial derivative for the gating network occurring in (6) have been calculated in the context of a gradient descent algorithm for the mixture model (3). Bishop's outcomes are restated here (using the chain rule):

$$\frac{\partial E'}{\partial z_j} = \sum_{k} \frac{\partial E'}{\partial g_k}\frac{\partial g_k}{\partial z_j} = \sum_{k} -\frac{\pi_k}{g_k}(\delta_{jk} g_k - g_j g_k) = g_j - \pi_j, \tag{8}$$

where the posterior probability π_j is defined as:

$$\pi_j(\mathbf{x},\mathbf{t}) = \frac{g_j \phi_j}{\sum_i g_i \phi_i}, \tag{9}$$

and δ_{jk} is the Kronecker delta. The functional derivative set zero with respect to the gating network outputs is (substituting (8) in (6)):

$$\frac{\delta E}{\delta z_j} = \int (g_j - \pi_j)\; p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\, d\mathbf{t} \; = 0. \tag{10}$$

Next, the expert network equation (7) will be treated for both multinomial and Gaussian conditional densities ϕ_j as expert mixing components.

3.1 Multinomial Conditional Density

A suitable choice for the expert conditional density function in classification problems with 1-of-c coding is a multinomial density:

$$\phi_j(\mathbf{t}^n|\mathbf{x}^n) = \prod_{c=1}^{C} (y^n_{jc})^{t^n_c}. \tag{11}$$

With multinomial conditional densities, a suitable choice for the activation function for the expert output units of is the soft-max function (2):

$$y_{jc} = \frac{\exp(a_{jc})}{\sum_k \exp(a_{jk})}, \tag{12}$$

where the a_{jc} are the expert network outputs before thresholding. In this case, the y_{jc} are dependent, therefore we investigate the functional derivative with respect to a_{jc} instead of (7):

$$\frac{\delta E}{\delta a_{jc}} = \int \left(\frac{\partial E'}{\partial a_{jc}}\right) p(\mathbf{t}|\mathbf{x})p(\mathbf{x})d\mathbf{t} = \int \sum_k \frac{\partial E'}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial a_{jc}} p(\mathbf{t}|\mathbf{x})p(\mathbf{x})d\mathbf{t} = 0. \tag{13}$$

Because of the soft-max function, the second partial derivative in (13) is similar to its counterpart in the gating network equation $\partial g_k/\partial z_j$ (see the second term in (8)):

$$\frac{\partial y_{jk}}{\partial a_{jc}} = \delta_{ck} y_{jk} - y_{jc} y_{jk}. \tag{14}$$

Using the definition of E' (5) and of the multinomial density ϕ_j (11) gives for the first partial derivative in (13):

$$\frac{\partial E'}{\partial y_{jk}} = \frac{\partial\left(-\ln \sum_{i=1}^{m} g_i \phi_i\right)}{\partial y_{jk}} = \frac{\partial\left(-\ln \sum_{i=1}^{m} g_i \prod_{c=1}^{C} (y_{ic})^{t_c}\right)}{\partial y_{jk}},$$

that is, taking the partial derivative and using (9):

$$\frac{\partial E'}{\partial y_{jk}} = -\frac{g_j y_{jk}^{(t_k-1)} t_k}{\sum_{i=1}^{m} g_i \phi_i} \prod_{c=1, c\neq k}^{C} (y_{jc})^{t_c} = -\frac{g_j \phi_j}{\sum_{i=1}^{m} g_i \phi_i} \frac{t_k}{y_{jk}} = -\pi_j \frac{t_k}{y_{jk}}. \tag{15}$$

Preparing for the solution of (13) one needs (using (14) and (15)):

$$\sum_k \frac{\partial E'}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial a_{jc}} = -\sum_k \pi_j(\mathbf{x},\mathbf{t}) \frac{t_k}{y_{jk}} (\delta_{ck} y_{jk} - y_{jc} y_{jk}) = \pi_j(\mathbf{x},\mathbf{t}) y_{jc} - \pi_j(\mathbf{x},\mathbf{t}) t_c, \tag{16}$$

where in the last step it has been used that for 1-of-c classification problems, $\sum_k t_k = 1$. The functional derivative set to zero with respect to the expert network outputs is (substituting (16) in (13)):

$$\frac{\delta E}{\delta a_{jc}} = \int (\pi_j(\mathbf{x},\mathbf{t}) y_{jc} - \pi_j(\mathbf{x},\mathbf{t}) t_c)\ p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\,d\mathbf{t} = 0. \tag{17}$$

3.2 Gaussian Conditional Density

In section 6.4 of [1] mixture models are considered with multi-dimensional Gaussian conditional densities (where the covariance matrix is the identity matrix) as mixture components:

$$\phi_j(\mathbf{t}^n|\mathbf{x}^n) = \frac{1}{(2\pi)^{(d/2)}} \exp\left(-\frac{||\mathbf{t} - \mathbf{y}_j(\mathbf{x})||^2}{2}\right), \tag{18}$$

where d is the dimensionality of $\mathbf{t}$. The activation function for the expert output units is chosen to be the linear, so $y_{jc} = a_{jc}$. Using the definition of E' (5) with Gaussian densities gives for the partial derivative in (7):

$$\frac{\partial E'}{\partial y_{jc}} = \frac{\partial\left(-\ln \sum_{i=1}^{m} g_i \phi_i\right)}{\partial y_{jc}} = -\frac{1}{\sum_{i=1}^{m} g_i \phi_i} \frac{\partial (g_j \phi_j)}{\partial y_{jc}} = \pi_j (y_{jc} - t_c) = \pi_j(\mathbf{x}, \mathbf{t}) y_{jc} - \pi_j(\mathbf{x}, \mathbf{t}) t_c,$$

which is the same as for the multinomial case (16) and therefore also leads to (17).

3.3 Interpretation of Network Outputs

What is left is to determine the $g_j(\mathbf{x})$ and $y_j(\mathbf{x})$ that solve (10) and (17) (and therefore minimize the ME error function). For the gating network outputs (10):

$$\frac{\delta E}{\delta z_j} = g_j p(\mathbf{x}) \int p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \;-\; p(\mathbf{x}) \int \pi_j(\mathbf{x}, \mathbf{t})\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \;= 0.$$

using that the conditional probability $p(\mathbf{t}|\mathbf{x})$ is normalized:

$$\frac{\delta E}{\delta z_j} = g_j p(\mathbf{x}) \;-\; p(\mathbf{x}) \int \pi_j(\mathbf{x}, \mathbf{t})\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \;= 0.$$

Therefore, at the minimum of the ME error function the gating network outputs satisfy:

$$g_j = \int \pi_j(\mathbf{x}, \mathbf{t})\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t}. \tag{19}$$

For the expert network outputs (17):

$$\frac{\delta E}{\delta a_{jc}} = y_{jc} p(\mathbf{x}) \int \pi_j(\mathbf{x}, \mathbf{t}) - p(\mathbf{x}) \int \pi_j(\mathbf{x}, \mathbf{t}) t_c\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \;\; p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \;= 0.$$

Therefore, at the minimum of the ME error function the expert network outputs satisfy:

$$y_{jc} = \frac{\int \pi_j(\mathbf{x}, \mathbf{t}) t_c\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t}}{\int \pi_j(\mathbf{x}, \mathbf{t})\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t}}. \tag{20}$$

Finally, using (19) and (20), the output vector of a mixture of experts that minimizes the ME error function is (1):

$$y_c(\mathbf{x}) = \sum_j g_j(\mathbf{x}) y_{jc}(\mathbf{x}) = \sum_j \int \pi_j(\mathbf{x}, \mathbf{t}) t_c\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t},$$

exchanging integration and summation:

$$\int \sum_j \pi_j(\mathbf{x}, \mathbf{t}) t_c\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} = \int t_c\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} := \langle t_c|\mathbf{x}\rangle, \tag{21}$$

where we have used that the posterior probabilities $\pi_j(\mathbf{x}, \mathbf{t})$ (9) sum to unity. The interpretation of (21) is that the output $y_c(\mathbf{x})$ of a ME at the minimum of the ME error function is equal to the conditional average of the target data. This is exactly the same as for the outputs of a network trained by minimizing the sum-of-squares or cross-entropy error functions [1][10]. It is a well-known result that for a classification problem with 1-of-c coding the conditional average of the target data is (see, for example, section 6.6 in [1]) :

$$y_c(\mathbf{x}) = \langle t_c|\mathbf{x}\rangle = P(\mathcal{C}_c|\mathbf{x}),$$

so that the outputs of a ME do indeed estimate the a posteriori probability that $\mathbf{x}$ belongs to class $\mathcal{C}_c$.

4 Discussion

In this paper, it has been shown that the minimization of the ME error function for classification problems leads to optimal ME outputs that estimate the a posteriori probabilities of class membership. This property is a very useful one, for example, for the integration of MEs in hybrid HMM/ANN systems. Future work should indicate how accurate these probability estimates are on real-world problems with a limited amount of data and local minima during training.

Acknowledgments

The author gratefully acknowledges the Swiss National Science Foundation (FN:21-45621.95) for their support of this research.

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[2] Hervé Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers, 1994.

[3] J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs with relationships to statistical pattern recognition. In F. Fogelman Soulié and J. Hérault, editors, *Neurocomputing: Algorithms, Architectures, and Applications*, pages 227–236. Springer Verlag, New York, 1990.

[4] Jürgen Fritsch, Michael Finke, and Alex Waibel. Context-dependent hybrid HME/HMM speech recognition using polyphone clustering decision trees. In *Proceedings of ICASSP-97*, 1997.

[5] Mariano Giaquinta and Stefan Hildebrandt. *Calculus of Variations.* Springer Verlag, Berlin, 1996.

[6] J. B. Hampshire and B. Pearlmutter. Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, pages 159–172, San Mateo, CA, 1990. Morgan Kaufmann.

[7] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[8] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[9] Geoffrey J. McLachlan and Kaye E. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, Inc., New York, 1988.

[10] M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation*, 3:461–483, 1991.

[11] S. R. Waterhouse and A. J. Robinson. Classification using hierarchical mixtures of experts. In *Proceedings 1994 IEEE Workshop on Neural Networks for Signal Processing*, pages 177–186, Long Beach CA, 1994. IEEE Press.

[12] Andreas S. Weigend, Morgan Mangeas, and Ashok N. Srivastava. Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6:373–399, 1995.