We present a new supervised learning procedure for systems composed of many separate networks, each of which learns to handle a subset of the complete set of training cases. The new procedure can be viewed either as a modular version of a multilayer supervised network, or as an associative version of competitive learning. It therefore provides a new link between these two apparently different approaches. We demonstrate that the learning procedure divides up a vowel discrimination task into appropriate subtasks, each of which can be solved by a very simple expert network.

1 Making Associative Learning Competitive

If backpropagation is used to train a single, multilayer network to perform different subtasks on different occasions, there will generally be strong interference effects that lead to slow learning and poor generalization. If we know in advance that a set of training cases may be naturally divided into subsets that correspond to distinct subtasks, interference can be reduced by using a system composed of several different "expert" networks plus a gating network that decides which of the experts should be used for each training case. Hampshire and Waibel (1989) have described a system of this kind that can be used when the division into subtasks is known prior to training, and Jacobs et al. (1990) have described a related system that learns how to allocate cases to experts. The idea behind such a system is that the gating network allocates a new case to one or a few experts, and, if the output is incorrect, the weight changes are localized to these experts (and the gating network).

¹This idea was first presented by Jacobs and Hinton at the Connectionist Summer School in Pittsburgh in 1988.
So there is no interference with the weights of other experts that specialize in quite different cases. The experts are therefore local in the sense that the weights in one expert are decoupled from the weights in other experts. In addition they will often be local in the sense that each expert will be allocated to only a small local region of the space of possible input vectors.

Unfortunately, both Hampshire and Waibel and Jacobs et al. use an error function that does not encourage localization. They assume that the final output of the whole system is a linear combination of the outputs of the local experts, with the gating network determining the proportion of each local output in the linear combination. So the final error on case $c$ is

$$E^c = \|d^c - \sum_i p_i^c o_i^c\|^2$$  \hspace{1cm} (1.1)

where $o_i^c$ is the output vector of expert $i$ on case $c$, $p_i^c$ is the proportional contribution of expert $i$ to the combined output vector, and $d^c$ is the desired output vector in case $c$.

This error measure compares the desired output with a blend of the outputs of the local experts, so, to minimize the error, each local expert must make its output cancel the residual error that is left by the combined effects of all the other experts. When the weights in one expert change, the residual error changes, and so the error derivatives for all the other local experts change.2 This strong coupling between the experts causes them to cooperate nicely, but tends to lead to solutions in which many experts are used for each case. It is possible to encourage competition by adding penalty terms to the objective function to encourage solutions in which only one expert is active (Jacobs et al. 1990), but a simpler remedy is to redefine the error function so that the local experts are encouraged to compete rather than cooperate.

Instead of linearly combining the outputs of the separate experts, we imagine that the gating network makes a stochastic decision about which single expert to use on each occasion (see Fig. 1). The error is then the expected value of the squared difference between the desired and actual output vectors

$$E^c = \langle \|d^c - o_i^c\|^2 \rangle = \sum_i p_i^c \|d^c - o_i^c\|^2$$  \hspace{1cm} (1.2)

Notice that in this new error function, each expert is required to produce the whole of the output vector rather than a residual. As a result, the goal of a local expert on a given training case is not directly affected by the weights within other local experts. There is still some indirect

2For Hampshire and Waibel, this problem does not arise because they do not learn the task decomposition. They train each expert separately on its own preassigned subtask.
Figure 1: A system of expert and gating networks. Each expert is a feedforward network and all experts receive the same input and have the same number of outputs. The gating network is also feedforward, and typically receives the same input as the expert networks. It has normalized outputs $p_j = \exp(x_j)/\sum \exp(x_j)$, where $x_j$ is the total weighted input received by output unit $j$ of the gating network. The selector acts like a multiple input, single output stochastic switch; the probability that the switch will select the output from expert $j$ is $p_j$.

coupling because if some other expert changes its weights, it may cause the gating network to alter the responsibilities that get assigned to the experts, but at least these responsibility changes cannot alter the sign of the error that a local expert senses on a given training case. If both the gating network and the local experts are trained by gradient descent in this new error function, the system tends to devote a single expert to each training case. Whenever an expert gives less error than the weighted average of the errors of all the experts (using the outputs of the gating network to decide how to weight each expert’s error) its responsibility for that case
will be increased, and whenever it does worse than the weighted average its responsibility will be decreased.

The error function in equation 1.2 works in practice but in the simulations reported below we used a different error function which gives better performance:

\[ E^c = - \log \sum_i p_i^c e^{-\frac{1}{2}||d^c - o_i^c||^2} \]  

(1.3)

The error defined in equation 1.3 is simply the negative log probability of generating the desired output vector under the mixture of gaussians model described at the end of the next section. To see why this error function works better, it is helpful to compare the derivatives of the two error functions with respect to the output of an expert. From equation 1.2 we get

\[ \frac{\partial E^c}{\partial o_i^c} = -2p_i^c(d^c - o_i^c) \]  

(1.4)

while from equation 1.3 we get

\[ \frac{\partial E^c}{\partial o_i^c} = \frac{p_i^c e^{-\frac{1}{2}||d^c - o_i^c||^2}}{\sum_j p_j^c e^{-\frac{1}{2}||d^c - o_j^c||^2}} (d^c - o_i^c) \]  

(1.5)

In equation 1.4 the term \( p_i^c \) is used to weight the derivative for expert \( i \). In equation 1.5 we use a weighting term that takes into account how well expert \( i \) does relative to other experts. This is a more useful measure of the relevance of expert \( i \) to training case \( c \), especially early in the training. Suppose, for example, that the gating network initially gives equal weights to all experts and \( ||d^c - o_i^c|| > 1 \) for all the experts. Equation 1.4 will adapt the best-fitting expert the slowest, whereas equation 1.5 will adapt it the fastest.

2 Making Competitive Learning Associative

It is natural to think that the “data” vectors on which a competitive network is trained play a role similar to the input vectors of an associative network that maps input vectors to output vectors. This correspondence is assumed in models that use competitive learning as a preprocessing stage within an associative network (Moody and Darken 1989). A quite different view is that the data vectors used in competitive learning correspond to the output vectors of an associative network. The competitive network can then be viewed as an inputless stochastic generator of output vectors and competitive learning can be viewed as a procedure for making the network generate output vectors with a distribution that matches the distribution of the “data” vectors. The weight vector of each competitive hidden unit represents the mean of a multidimensional gaussian
distribution, and output vectors are generated by first picking a hidden unit and then picking an output vector from the gaussian distribution determined by the weight vector of the chosen hidden unit. The log probability of generating any particular output vector $o'$ is then

$$\log P^c = \log \sum_i p_i ke^{-\frac{1}{2}||\mu_i - o'||^2}$$  \hspace{1cm} (2.1)

where $i$ is an index over the hidden units, $\mu_i$ is the "weight" vector of the hidden unit, $k$ is a normalizing constant, and $p_i$ is the probability of picking hidden unit $i$, so the $p_i$ are constrained to sum to 1. In the statistics literature (McLachlan and Basford 1988), the $p_i$ are called "mixing proportions."

"Soft" competitive learning modifies the weights (and also the variances and the mixing proportions) so as to increase the product of the probabilities (i.e., the likelihood) of generating the output vectors in the training set (Nowlan 1990a). "Hard" competitive learning is a simple approximation to soft competitive learning in which we ignore the possibility that a data vector could be generated by several different hidden units. Instead, we assume that it must be generated by the hidden unit with the closest weight vector, so only this weight vector needs to be modified to increase the probability of generating the data vector.

If we view a competitive network as generating output vectors, it is not immediately obvious what role input vectors could play. However, competitive learning can be generalized in much the same way as Barto (1985) generalized learning automata by adding an input vector and making the actions of the automaton be conditional on the input vector. We replace each hidden unit in a competitive network by an entire expert network whose output vector specifies the mean of a multidimensional gaussian distribution. So the means are now a function of the current input vector and are represented by activity levels rather than weights. In addition, we use a gating network which allows the mixing proportions of the experts to be determined by the input vector. This gives us a system of competing local experts with the error function defined in equation 1.3. We could also introduce a mechanism to allow the input vector to dynamically determine the covariance matrix for the distribution defined by each expert network, but we have not yet experimented with this possibility.

3 Application to Multispeaker Vowel Recognition

The mixture of experts model was evaluated on a speaker independent, four-class, vowel discrimination problem (Nowlan 1990b). The data consisted of the first and second formants of the vowels [i], [I], [a], and [A] (usually denoted [A]) from 75 speakers (males, females, and children) uttered in a hVd context (Peterson and Barney 1952). The data forms two
Figure 2: Data for vowel discrimination problem, and expert and gating network decision lines. The horizontal axis is the first formant value, and the vertical axis is the second formant value (the formant values have been linearly scaled by dividing by a factor of 1000). Each example is labeled with its corresponding vowel symbol. Vowels [i] and [I] form one overlapping pair of classes, vowels [a] and [A] form the other pair. The lines labeled Net 0, 1, and 2 represent the decision lines for 3 expert networks. On one side of these lines the output of the corresponding expert is less than 0.5, on the other side the output is greater than 0.5. Although the mixture in this case contained 4 experts, one of these experts made no significant contribution to the final mixture since its mixing proportion $p_i$ was effectively 0 for all cases. The line labeled Gate 0:2 indicates the decision between expert 0 and expert 2 made by the gating network. To the left of this line $p_2 > p_0$, to the right of this line $p_0 > p_2$. The boundary between classes [a] and [A] is formed by the combination of the left part of Net 2's decision line and the right part of Net 0's decision line. Although the system tends to use as few experts as it can to solve a problem, it is also sensitive to specific problem features such as the slightly curved boundary between classes [a] and [A].

Pairs of overlapping classes, and different experts learn to concentrate on one pair of classes or the other (Fig. 2).

We compared standard backpropagation networks containing a single hidden layer of 6 or 12 units with mixtures of 4 or 8 very simple experts. The architecture of each expert was restricted so it could form only a linear decision surface, which is defined as the set of input vectors for which the expert gives an output of exactly 0.5. All models were trained with data from the first 50 speakers and tested with data from the remaining 25 speakers. The small number of parameters for each expert allows excellent generalization performance (Table 1), and permits
Table 1: Summary of Performance on Vowel Discrimination Task. Results are based on 25 simulations for each of the alternative models. The first column of the table indicates the system simulated. The second column gives the percent of training cases classified correctly by the final set of weights, while the third column indicates the percent of testing cases classified correctly. The last two columns contain the average number of epochs required to reach the error criterion, and the standard deviation of the distribution of convergence times. Although the squared error was used to decide when to stop training, the criterion for correct performance is based on a weighted average of the outputs of all the experts. Each expert assigns a probability distribution over the classes and these distributions are combined using proportions given by the gating network. The most probable class is then taken to be the response of the system. The identical performance of all the systems is due to the fact that, with this data set, the set of misclassified examples is not sensitive to small changes in the decision surfaces. Also, the test set is easier than the training set.

<table>
<thead>
<tr>
<th>System</th>
<th>Train % correct</th>
<th>Test % correct</th>
<th>Average number of epochs</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Experts</td>
<td>88</td>
<td>90</td>
<td>1124</td>
<td>23</td>
</tr>
<tr>
<td>8 Experts</td>
<td>88</td>
<td>90</td>
<td>1083</td>
<td>12</td>
</tr>
<tr>
<td>BP 6 Hid</td>
<td>88</td>
<td>90</td>
<td>2209</td>
<td>83</td>
</tr>
<tr>
<td>BP 12 Hid</td>
<td>88</td>
<td>90</td>
<td>2435</td>
<td>124</td>
</tr>
</tbody>
</table>

a graphic representation of the process of task decomposition (Figure 3). The number of hidden units in the backpropagation networks was chosen to give roughly equal numbers of parameters for the backpropagation networks and mixture models. All simulations were performed using a simple gradient descent algorithm with fixed step size $\epsilon$. To simplify the comparisons, no momentum or other acceleration techniques were used. The value of $\epsilon$ for each system was chosen by performing a limited exploration of the convergence from the same initial conditions for a range of $\epsilon$. Batch training was used with one weight update for each pass through the training set (epoch). Each system was trained until an average squared error of 0.08 over the training set was obtained.

The mixtures of experts reach the error criterion significantly faster than the backpropagation networks ($p \gg 0.999$), requiring only about half as many epochs on average (Table 1). The learning time for the mixture model also scales well as the number of experts is increased: The mixture of 8 experts has a small, but statistically significant ($p > 0.95$), advantage in the average number of epochs required to reach the error criterion. In contrast, the 12 hidden unit backpropagation network requires more epochs ($p > 0.95$) to reach the error criterion than the network with 6
Figure 3: The trajectories of the decision lines of some experts during one simulation. The horizontal axis is the first formant value, and the vertical axis is the second formant value. Each trajectory is represented by a sequence of dots, one per epoch, each dot marking the intersection of the expert's decision line and the normal to that line passing through the origin. For clarity, only 5 of the 8 experts are shown and the number of the expert is shown at the start of the trajectory. The point labeled **T0** indicates the optimal decision line for a single expert trained to discriminate [i] from [I]. Similarly, **T1** represents the optimal decision line to discriminate [a] from [A]. The point labeled **X** is the decision line learned by a single expert trained with data from all 4 classes, and represents a type of average solution.

hidden units (Table 1). All statistical comparisons are based on a *t* test with 48 degrees of freedom and a pooled variance estimator.

Figure 3 shows how the decision lines of different experts move around as the system learns to allocate pieces of the task to different experts. The system begins in an unbiased state, with the gating network assigning equal mixing proportions to all experts in all cases. As a result, each expert tends to get errors from roughly equal numbers of cases in all 4 classes, and all experts head towards the point **X**, which represents the optimal decision line for an expert that must deal with all the cases. Once one or more experts begin to receive more error from cases in one class pair than the other, this symmetry is broken and the trajectories begin to diverge as different experts concentrate on one class pair or the other. In this simulation, expert 5 learns to concentrate on discriminating classes [i] and [I] so its decision line approaches the optimal line for this discrimination (**T0**). Experts 4 and 6 both concentrate on discriminating classes [a] and [A], so their trajectories approach the
optimal single line (T1) and then split to form a piecewise linear approximation to the slightly curved optimal decision surface (see Fig. 2). Only experts 4, 5, and 6 are active in the final mixture. This solution is typical — in all simulations with mixtures of 4 or 8 experts all but 2 or 3 experts had mixing proportions that were effectively 0 for all cases.

Acknowledgments

Jordan and Jacobs were funded by grants from Siemens and the McDonnell–Pew program in Cognitive Neuroscience. Hinton and Nowlan were funded by grants from the Ontario Information Technology Research Center and the Canadian Natural Science and Engineering Research Council. Hinton is a fellow of the Canadian Institute for Advanced Research.

References


This article has been cited by:

2. Bruno Damas, José Santos-Victor. 2013. Online Learning of Single- and Multivalued Functions with an Infinite Mixture of Linear Experts. *Neural Computation* **25**:11, 3044-3091. [Abstract] [Full Text] [PDF] [PDF Plus]
10. Steven J. Simske. 2013. Introduction and Overview 1-41. [CrossRef]
13. Heeyoul Choi, Seungjin Choi, Yoonsuck Choe. 2013. Parameter Learning for Alpha Integration. *Neural Computation* **25**:6, 1585-1604. [Abstract] [Full Text] [PDF] [PDF Plus]


19. RASTISLAV J. R. STRUHARIK, LADISLAV A. NOVAK. 2013. HARDWARE IMPLEMENTATION OF DECISION TREE ENSEMBLES. *Journal of Circuits, Systems and Computers* 1350032. [CrossRef]


22. Rahul Kala, Anupam Shukla, Ritu Tiwari. 2013. Breast Cancer Diagnosis Using Optimized Attribute Division in Modular Neural Networks 34-47. [CrossRef]


52. B. Verma Neural Network Based Classifier Ensembles 229-239. [CrossRef]

53. CHUANYU SUN, XIAO-LIN WU, KENT A. WEIGEL, GUILHERME J. M. ROSA, STEWART BAUCK, BRENT W. WOODWARD, ROBERT D. SCHNABEL, JEREMY F. TAYLOR, DANIEL GIANOLA. 2012. An ensemble-based approach to imputation of moderate-density genotypes for genomic selection with application to Angus cattle. *Genetics Research* **94**:03, 133-150. [CrossRef]


55. Kristine Monteith, Tony Martinez. 2012. AGGREGATE CERTAINTY ESTIMATORS. *Computational Intelligence* no–no. [CrossRef]


60. MATTEO RE, GIORGIO VALENTINIE. Ensemble Methods 20124949. [CrossRef]


72. Andrew R. Webb, Keith D. Copsey. References 591-636. [CrossRef]


78. Yan Yang, Jinwen Ma. 2011. Asymptotic Convergence Properties of the EM Algorithm for Mixture of Experts. *Neural Computation* 23:8, 2140-2168. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplementary Content]


80. Shuhaida Ismail, Ani Shabri, Ruhaidah Samsudin. 2011. A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting. *Expert Systems with Applications* 38:8, 10574-10578. [CrossRef]

81. Lei Xu. Learning Algorithms for RBF Functions and Subspace Based Functions 1034-1065. [CrossRef]

82. Lei Xu, Shun-ichi Amari. Combining Classifiers and Learning Mixture-of-Experts 243-252. [CrossRef]


90. Haibo He. 2011. Ensemble Learning 108-139. [CrossRef]


96. Isobel Claire Gormley, Thomas Brendan Murphy. 2011. Mixture of Experts Modelling with Social Science Applications 101-121. [CrossRef]

97. Feng Li, Mattias Villani, Robert Kohn. 2011. Modelling Conditional Densities Using Finite Smooth Mixtures 123-144. [CrossRef]


112. Esma Kilic, Ethem Alpaydin. 2011. Learning the areas of expertise of classifiers in an ensemble. *Procedia Computer Science* 3, 74-82. [CrossRef]


118. Chrisantha Fernando, Richard Goldstein, Eörs Szathmáry. 2010. The Neuronal Replicator Hypothesis. *Neural Computation* **22**:11, 2809-2857. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplementary Content]

119. Luis M. Silva, J. Marques de Sá, Luís A. Alexandre. 2010. The MEE Principle in Data Classification: A Perceptron-Based Analysis. *Neural Computation* **22**:10, 2698-2728. [Abstract] [Full Text] [PDF] [PDF Plus]


138. Shile Zhang, Bin Li, Xiangyang Xue. 2010. Semi-automatic dynamic auxiliary-tag-aided image annotation#. *Pattern Recognition* 43:2, 470–477. [CrossRef]


146. RASTISLAV J. R. STRUHARIK, LADISLAV A. NOVAK. 2009. EVOLVING DECISION TREES IN HARDWARE. *Journal of Circuits, Systems and Computers* 18:06, 1033-1060. [CrossRef]


174. Serkan Tapkin, Ozdemir Akyilmaz. 2009. A new approach to neural trip distribution models: NETDIM. *Transportation Planning and Technology* **32**:1, 93-114. [CrossRef]

175. Mike Oaksford, Nick Chater. 2009. The uncertain reasoner: Bayes, logic, and rationality. *Behavioral and Brain Sciences* **32**:01, 105-120. [CrossRef]


177. Sergios Theodoridis, Konstantinos Koutroumbas. *Nonlinear Classifiers* 151-260. [CrossRef]


197. Jun Tani, Ryu Nishimoto, Jun Namikawa, Masato Ito. 2008. Codevelopmental Learning Between Human and Humanoid Robot Using a Dynamic Neural-


211. Mitsuo Kawato, Daniel Wolpert. Internal Models for Motor Control 291-307. [CrossRef]


214. L XU. 2007. A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving#. *Pattern Recognition* 40:8, 2129-2153. [CrossRef]


220. Abdelhamid Bouchachia. 2007. Learning with partly labeled data. *Neural Computing and Applications* 16:3, 267-293. [CrossRef]


226. Ludmila Kuncheva, Juan Rodriguez. 2007. Classifier Ensembles with a Random Linear Oracle. *IEEE Transactions on Knowledge and Data Engineering* **19**:4, 500-508. [CrossRef]


233. Xia Hong, Sheng Chen, Chris J. Harris. 2007. A Kernel-Based Two-Class Classifier for Imbalanced Data Sets. *IEEE Transactions on Neural Networks* **18**:1, 28-41. [CrossRef]


238. Junfeng Pan, Qiang Yang, Yiming Yang, Lei Li, Frances Li, George Li. 2007. Cost-Sensitive-Data Preprocessing for Mining Customer Relationship Management Databases. *IEEE Intelligent Systems* **22**:1, 46-51. [CrossRef]


254. Robi PolikarPattern Recognition . [CrossRef]

256. J. Zhang, Q. Jin, Y. Xu. 2006. Inferential Estimation of Polymer Melt Index Using Sequentially Trained Bootstrap Aggregated Neural Networks. *Chemical Engineering & Technology* 29:4, 442-448. [CrossRef]

257. Erhan Oztop, Mitsuo Kawato, Michael Arbib. 2006. Mirror neurons and imitation: A computationally guided review. *Neural Networks* 19:3, 254-271. [CrossRef]

258. D LOYOLAR. 2006. Applications of neural network methods to the processing of earth observation satellite data. *Neural Networks* 19:2, 168-177. [CrossRef]


262. Marcus FreanConnectionist Architectures: Optimization . [CrossRef]


264. Xin Yao, Yong Xu. 2006. Recent Advances in Evolutionary Computation. *Journal of Computer Science and Technology* 21:1, 1-18. [CrossRef]


267. J. Peres, F. Freitas, MAM Reis, S. Feyo de Azevedo, R. OliveiraHybrid modular mechanistic/ANN modelling of a wastewater phosphorus removal process 21, 1717-1722. [CrossRef]


269. Steven Walczak, Madhavan Parthasarathy. 2006. Modeling online service discontinuation with nonparametric agents. *Information Systems and e-Business Management* 4:1, 49-70. [CrossRef]

270. Sergios Theodoridis, Konstantinos KoutroubasNonlinear Classifiers 121-211. [CrossRef]

271. Biswanath BhattacharyaReferences . [CrossRef]


276. Geoffrey Hinton. *Artificial Intelligence: Neural Networks*. [CrossRef]


289. BRENT FERGUSON, RANADHIR GHOSH, JOHN YEARWOOD. 2005. MODULAR NEURAL NETWORK DESIGN FOR THE PROBLEM OF ALPHABETIC CHARACTER RECOGNITION. *International Journal of Pattern Recognition and Artificial Intelligence* 19:02, 249-269. [CrossRef]


292. A HAGEN. 2005. Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR. *Computer Speech & Language* 19:1, 3-30. [CrossRef]


295. David Windridge. Morphologically Debiased Classifier Fusion: A Tomography-Theoretic Approach 134, 181-266. [CrossRef]


299. Roelof K. Brouwer. 2004. A hybrid neural network for input that is both categorical and quantitative. *International Journal of Intelligent Systems* 19:10, 979-1001. [CrossRef]

301. SHIMON COHEN, NATHAN INTRATOR. 2004. ON DIFFERENT MODEL SELECTION CRITERIA IN A FORWARD AND BACKWARD REGRESSION HYBRID NETWORK. International Journal of Pattern Recognition and Artificial Intelligence 18:05, 847-865. [CrossRef]

302. CHRISTOPH KÖNIG, GIUSEPPINA GINI, MARIAN CRACIUN, EMILIO BENFENATI. 2004. MULTICLASS CLASSIFIER FROM A COMBINATION OF LOCAL EXPERTS: TOWARD DISTRIBUTED COMPUTATION FOR REAL-PROBLEM CLASSIFIERS. International Journal of Pattern Recognition and Artificial Intelligence 18:05, 801-817. [CrossRef]


308. J Brown. 2004. How laminar frontal cortex and basal ganglia circuits interact to control planned and reactive saccades. Neural Networks 17:4, 471-510. [CrossRef]

309. S.-K. Ng, G.J. McLachlan. 2004. Using the EM Algorithm to Train Neural Networks: Misconceptions and a New Algorithm for Multiclass Classification. IEEE Transactions on Neural Networks 15:3, 738-749. [CrossRef]


316. J. Peres, R. Oliveira, L.S. Serafim, P. Lemos, M.A. Reis, S. Fuyo de AzevedoHybrid Modelling of a PHA Production Process Using Modular Neural Networks **18**, 733-738. [CrossRef]


321. Kai Huang, Robert F. Murphy. 2004. From quantitative microscopy to automated image understanding. *Journal of Biomedical Optics* **9**:5, 893. [CrossRef]


335. RONAN COLLOBERT, YOSHUA BENGIO, SAMY BENGIO. 2003. SCALING LARGE LEARNING PROBLEMS WITH HARD PARALLEL MIXTURES. *International Journal of Pattern Recognition and Artificial Intelligence* 17:03, 349-365. [CrossRef]


342. J Tani. 2003. Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks* **16**:1, 11-23. [CrossRef]


348. L Xu. 2002. BYY harmony learning, structural RPCL, and topological self-organizing on mixture models1. *Neural Networks* **15**:8-9, 1125-1151. [CrossRef]


351. R Brouwer. 2002. A feed-forward network for input that is both categorical and quantitative. *Neural Networks* **15**:7, 881-890. [CrossRef]


353. Lizhong Wu, S.L. Oviatt, P.R. Cohen. 2002. From members to teams to committee—a robust approach to gestural and multimodal recognition. *IEEE Transactions on Neural Networks* **13**:4, 972-982. [CrossRef]


356. Peter Dayan Reinforcement Learning. [CrossRef]
357. PITIYO HARTONO, SHUJI HASHIMOTO. 2002. EXTRACTING THE PRINCIPAL BEHAVIOR OF A PROBABILISTIC SUPERVISOR THROUGH NEURAL NETWORKS ENSEMBLE. *International Journal of Neural Systems* **12**:03n04, 291-301. [CrossRef]


395. YOUNÈS BENNANI, FABRÌCE BOSSAERT. 2001. MODULAR CONNECTIONIST MODELLING AND CLASSIFICATION APPROACHES FOR LOCAL DIAGNOSIS IN TELECOMMUNICATION TRAFFIC MANAGEMENT. International Journal of Computational Intelligence and Applications 01:01, 53-70. [CrossRef]


401. ZOUBIN GHAHRAMANI. 2001. AN INTRODUCTION TO HIDDEN MARKOV MODELS AND BAYESIAN NETWORKS. *International Journal of Pattern Recognition and Artificial Intelligence* 15:01, 9-42. [CrossRef]


409. Martin Kreutz, Maik Anschütz, Thorsten Grünendick, Andreas Rick, Stefan Gehlen, Klaus Hoffmann. 2001. AUTOMATED DIAGNOSIS OF SKIN CANCER USING DIGITAL IMAGE PROCESSING AND MIXTURE-OF-EXPERTS. *Biomedizinische Technik/Biomedical Engineering* 46:s1, 376-377. [CrossRef]


413. T. Higuchi, Xin Yao, Yong Liu. 2000. Evolutionary ensembles with negative
correlation learning. *IEEE Transactions on Evolutionary Computation* 4:4, 380-387. [CrossRef]

distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22:11, 1318-1323. [CrossRef]

Research* 27:11-12, 1131-1152. [CrossRef]

416. E Adi-Japha. 2000. Regulation of division of labour between cognitive systems
controlling action. *Cognition* 76:1, 1-11. [CrossRef]

classification of gender, ethnic origin, and pose of human faces. *IEEE Transactions
on Neural Networks* 11:4, 948–960. [CrossRef]

418. Andreas S. Weigend, Shanming Shi. 2000. Predicting daily probability

Computation* 12:6, 1293-1301. [Abstract] [PDF] [PDF Plus]

420. Ichiro Takeuchi, Takeshi Furuhashi. 2000. A description of dynamic behavior of
sensory/motor systems with fuzzy symbolic dynamic systems. *Artificial Life and
Robotics* 4:2, 84-88. [CrossRef]

mixtures-of-experts for generalized linear models. *IEEE Transactions on
Information Theory* 46:3, 1005-1013. [CrossRef]

modeling. *IEEE Transactions on Fuzzy Systems* 8:2, 125-142. [CrossRef]


network forecasters for prediction of natural gas consumption. *IEEE Transactions
on Neural Networks* 11:2, 464-473. [CrossRef]

425. A. Karniel, G.F. Inbar. 2000. Human motor control: learning to control a time-
varying, nonlinear, many-to-one system. *IEEE Transactions on Systems, Man and
Cybernetics, Part C (Applications and Reviews)* 30:1, 1-11. [CrossRef]


427. Chapter 8 Design issues — Neural networks 1, 89-102. [CrossRef]

learning and representation. *Optical Engineering* 39:5, 1230. [CrossRef]


436. Paul J. Werbos. *Neurocontrollers*. [CrossRef]

437. Yair Bartal. *Divide-and-Conquer Methods*. [CrossRef]


442. KE CHEN, HUISHENG CHI. 1999. A MODULAR NEURAL NETWORK ARCHITECTURE FOR PATTERN CLASSIFICATION BASED ON DIFFERENT FEATURE SETS. *International Journal of Neural Systems* **09**:06, 563-581. [CrossRef]


448. H Ando. 1999. Unsupervised visual learning of three-dimensional objects using a modular network architecture. *Neural Networks* **12**:7-8, 1037-1051. [CrossRef]


466. GASSER AUDA, MOHAMED KAMEL. 1999. MODULAR NEURAL NETWORKS: A SURVEY. *International Journal of Neural Systems* 09:02, 129-151. [CrossRef]


473. Kenneth J. Kurtz, Dedre Gentner, Virginia Gunn. Reasoning 145-200. [CrossRef]


476. P Schyns. 1999. Dr. Angry and Mr. Smile: when categorization flexibly modifies the perception of faces in rapid visual presentations. *Cognition* 69:3, 243-265. [CrossRef]


500. David J. Miller, Hasan S. Uyar. 1998. Combined Learning and Use for a Mixture Model Equivalent to the RBF Classifier. *Neural Computation* 10:2, 281-293. [Abstract] [PDF] [PDF Plus]


503. Ian T Nabney, Christopher M Bishop. Modeling Wind Direction from Satellite Scatterometer Data 295-301. [CrossRef]

504. Itiel E. Dror, Christopher S. Schreiner. Chapter 4 Neural networks and perception 126, 77-85. [CrossRef]


506. Monica Bianchini, Paolo Frasconi, Marco Gori, Marco Maggini. Optimal learning in artificial neural networks: A theoretical view 2, 1-51. [CrossRef]

507. Colin Campbell. Constructive learning techniques for designing neural network systems 3, 91-145. [CrossRef]


534. Y. Bengio, P. Frasconi. 1996. Input-output HMMs for sequence processing. *IEEE Transactions on Neural Networks* **7**:5, 1231-1249. [CrossRef]


541. D. Sarkar. 1996. Randomness in generalization ability: a source to improve it. *IEEE Transactions on Neural Networks* 7:3, 676-685. [CrossRef]


545. David Heath, Simon Kasif, Steven Salzberg. Chapter 18 Committees of decision trees 113, 305-317. [CrossRef]


