ELSEVIER

Artificial Intelligence in Medicine 30 (2004) 71–84

Artificial Intelligence in Medicine

www.elsevier.com/locate/artmed

Use of genetic algorithms for neural networks to predict community-acquired pneumonia

Paul S. Heckerling[a,*], Ben S. Gerber[a,b], Thomas G. Tape[c], Robert S. Wigton[c]

[a]*Department of Medicine (M/C 787), University of Illinois, 840 South Wood Street, Chicago, IL 60612, USA*
[b]*Department of Bioengineering, University of Illinois, 840 South Wood Street, Chicago, IL 60612, USA*
[c]*Department of Medicine, University of Nebraska, Omaha, NE, USA*

Received 15 October 2002; received in revised form 7 May 2003; accepted 12 May 2003

Abstract

Background: Genetic algorithms have been used to solve optimization problems for artificial neural networks (ANN) in several domains. We used genetic algorithms to search for optimal hidden-layer architectures, connectivity, and training parameters for ANN for predicting community-acquired pneumonia among patients with respiratory complaints. *Methods*: Feed-forward back-propagation ANN were trained on sociodemographic, symptom, sign, comorbidity, and radiographic outcome data among 1044 patients from the University of Illinois (the training cohort), and were applied to 116 patients from the University of Nebraska (the testing cohort). Binary chromosomes with genes representing network attributes, including the number of nodes in the hidden layers, learning rate and momentum parameters, and the presence or absence of implicit within-layer connectivity using a competition algorithm, were operated on by various combinations of crossover, mutation, and probabilistic selection based on network mean-square error (MSE), and separately on average cross entropy (ENT). Predictive accuracy was measured as the area under a receiver-operating characteristic (ROC) curve. *Results*: Over 50 generations, the baseline genetic algorithm evolved an optimized ANN with nine nodes in the first hidden layer, zero nodes in the second hidden layer, learning rate and momentum parameters of 0.5, and no within-layer competition connectivity. This ANN had an ROC area in the training cohort of 0.872 and in the testing cohort of 0.934 (*P*-value for difference, 0.181). Algorithms based on cross-generational selection, Gray coding of genes prior to mutation, and crossover recombination at different genetic levels, evolved optimized ANN identical to the baseline genetic strategy. Algorithms based on other strategies, including elite selection within generations (training ROC area 0.819), and inversions of genetic material during recombination (training ROC area 0.812), evolved less accurate ANN. *Conclusion*: ANN optimized by genetic algorithms accurately discriminated pneumonia within a training cohort, and within a

* Corresponding author. Tel.: +1-312-996-1599; fax: +1-312-413-8283.
E-mail address: pshecker@uic.edu (P.S. Heckerling).

doi:10.1016/S0933-3657(03)00065-4

testing cohort consisting of cases on which the networks had not been trained. Genetic algorithms can be used to implement efficient search strategies for optimal ANN to predict pneumonia.

Keywords: Artificial neural network; Genetic algorithm; Pneumonia

1. Introduction

Artificial neural networks (ANN) are computer programs that can match patterns in predictor variables to patterns in outcome variables [10]. ANN are based on neural architectures of the brain, and consist of layers of virtual neurons (nodes) that are linked to each other by weighted connections analogous to dendrites. By virtue of hidden layers of nodes that lie between the input and output layers of the network, and of nonlinear transfer functions that convert nodal input to output, ANN can model highly nonlinear relationships in data [34]. This may give them an advantage over standard statistical prediction methods such as logistic regression, in which nonlinear interactions between variables must be modeled in explicit functional form [39]. ANN trained with standard feed-forward, back-propagation methods have been successfully used to predict several medical conditions, including myocardial infarction [3], pulmonary embolism [33,38], community-acquired pneumonia [21], and pulmonary tuberculosis [9], as well as others.

One disadvantage of ANN, however, is that their output, and their accuracy, may vary depending on the parameters used to structure and to train them [6,15,34]. Networks can be structured with different numbers of hidden layers, different numbers of nodes within these layers, and with or without implicit within-layer connectivity; and can be trained with different learning rates and momentum parameters that govern the updating of weights as new cases are learned. Although different network topologies and training algorithms may affect network performance, strategies for determining optimal network architectures and training parameters for a given data set are largely empirical [1]. (There are, however, methods of network regularization that can be used to optimize network training for any given architecture; see [4]). Exhaustive search strategies to find the best combinations of network parameters for a specific prediction problem, and for a specific data set, in general cannot be done.

Genetic algorithms [12,14,23], based on quasi-Darwinian evolutionary principles, can be used to implement efficient search strategies for optimal ANN configurations. In a genetic algorithm, neural network parameters are represented by mathematical ‘‘chromosomes’’, which can be modified by mutation and by recombination in a process analogous to crossing over during meiosis. Networks based on these genotypic structures are tested for phenotypic ‘‘fitness’’, based on their predictive accuracy, and are propagated to subsequent generations proportionate to their fitness. Cycles of recombination, mutation, and fitness-based selection are iterated until chromosomes representing sufficiently accurate ANN are evolved. Genetic algorithms have been used with ANN to search for input variables [2,25,26,32,35], or to determine the number of nodes or connections in the network [8,26]. We used a genetic algorithm to search for optimal architectures, connectivity, and training parameters for ANN for predicting community-acquired pneumonia among patients with respiratory complaints.

2. Methods

2.1. Data collection

The methods have been described previously [21,22]. Data were collected on patients who presented to the emergency departments of the University of Illinois at Chicago (the training cohort) and the University of Nebraska Medical Center at Omaha (the testing cohort) with complaints of fever or respiratory symptoms who had received a chest X-ray to evaluate these symptoms. All patients had been examined by the emergency department physician, and the decision to order a chest X-ray was made by the examining physician. Patients had information collected on 35 demographic, symptom, sign, and comorbidity variables, as listed in Table 1. All variables were classified as categorical (0, 1) variables except for age, temperature, respiratory rate, and pulse rate, which were coded as continuous variables and normalized on a closed [0, 1] domain. Chest X-ray results were obtained from the radiologists' reports, and were classified into one of four categories: no pneumonia; possible pneumonia; probable pneumonia; and definite pneumonia. Because microbiologic and serologic data were not available for many patients, they were not incorporated into the gold-standard classification of pneumonia.

A total of 1325 patients (1131 from the University of Illinois and 194 from the University of Nebraska) were enrolled in the original data collection [22]. To develop the neural networks, we excluded 148 patients with missing data (89 from the University of Illinois and 59 from the University of Nebraska), leaving a total of 1177 patients. We also excluded 154 patients with a chest X-ray classification of possible pneumonia (137 from the University of Illinois and 17 from the University of Nebraska), in order to train and test the networks on patients who unambiguously either had no pneumonia, or who had definite or probable radiographic pneumonia (aggregated for purposes of analysis into one 'pneumonia' group) [21,22]. Thus, the final study cohort included 1023 patients, with a training cohort of 907 patients, of whom 133 had pneumonia and 774 had no pneumonia; and a testing cohort of 116 patients, of whom 41 had pneumonia and 75 had no pneumonia.

2.2. Neural networks

ANN were feed-forward, back-propagation networks [30,36] with 35 nodes in the input layer, 1 node in the output layer, and between 0 and 15 nodes in each of 0, 1, or 2 hidden layers, determined by the genetic algorithm. All networks had bias nodes on the output and hidden layers, to set thresholds for nodal activation, and used a binary sigmoid (logistic) transfer function scaled on the (0, 1) range. All networks had their weight matrices for the output and hidden layers initialized to random values between -0.5 and 0.5, and were started with the same random seed, to maximize comparability of their output. Networks were trained with learning rate parameters (to constrain the magnitude of the connection weight updates during network training), and momentum parameters (to modify weight updates by a proportion of the updates from the previous training case), as determined by the genetic algorithm. In addition, networks could be trained using a competition algorithm, which allowed nodes within each hidden layer to compete with each other for weight updates during training, and rewarded the node with the largest magnitude

Table 1
Clinical variables used as input variables in the neural networks predicting pneumonia

Variable
Demographic
Age[a]
Sex
Race
Symptom
Cough
Sputum
Fever
Chills
Pleuritic chest pain
Non-pleuritic chest pain
Dyspnea
Wheezing
Orthopnea
Paroxysmal nocturnal dyspnea
Sign
Temperature[a]
Respiratory rate[a]
Pulse rate[a]
Impaired mental status
Splinting
Cyanosis
Percussion dullness
Rales
Rhonchi
Wheezes
Decreased breath sounds
Bronchial breath sounds
Egophony
Pleural friction rub
Decreased thoracic expansion
Comorbidity
Asthma
Chronic obstructive pulmonary disease
Other lung disease
Congestive heart failure
Immunocompromising disease[b]
Dementia
Other comorbid condition

[a] Continuous variables, normalized on a closed [0, 1] domain. All other variables were categorical.

[b] Lymphoreticular or solid neoplasms, human immunodeficiency virus infection, collagen vascular diseases, chronic renal failure, or any other disease for which the patient was being treated with immunosuppressive medications excluding corticosteroid therapy for asthma.

update by forcing other nodes on the layer to adjust their weights in the opposite direction [13,20]; this was determined by the genetic algorithm as well. In baseline analyses, all networks were trained through 10 epochs, which approximated the number previously determined, using early-stopping techniques, to minimize split-sample cross-validation

error in the training cohort with different network architectures [21]. In separate analyses, each network was individually cross-validated using split-sample techniques, with training stopped at the number of epochs for which the cross-validation error reached its minimum value. Using this method, networks with different hidden-layer architectures, learning parameters, and within-layer connectivities, could potentially be trained over different numbers of epochs. Network outputs ranged from 0 to 1, where values closer to 1 represented a prediction of pneumonia, and values closer to 0 represented a prediction of no pneumonia.

2.3. Genetic algorithm

Neural network structure and training parameters were represented by haploid chromosomes consisting of ''genes'' of binary numbers (Fig. 1). Each chromosome had five genes. The first two genes were 4-bit binary numbers, representing the number of nodes in the first and second hidden layers of the network, which could each range from 0 to 15 nodes. The third and fourth genes were 2-bit binary numbers, representing the learning rate and momentum with which the network was trained, which could each assume discrete values of 0.01, 0.05, 0.1, or 0.5. The fifth gene was a 1-bit binary number, representing whether implicit within-layer connectivity using the competition algorithm was enabled.

These mathematical chromosomes could be operated upon by quasi-genetic processes of crossing over and mutation. To implement crossovers, chromosomes were randomly paired, and segments of paired chromosomes between two randomly determined breakpoints were swapped. Crossovers could be implemented either across genes, so that gene boundaries might potentially be breached by the exchange of genetic material; or within

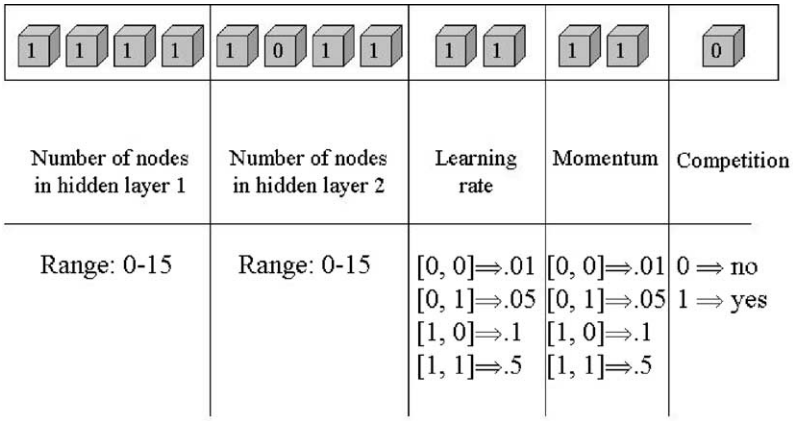

Fig. 1. Haploid chromosome with five binary numerical genes, representing structural and training parameters of a neural network. The first two genes represent the number of nodes in the first and second hidden layers of the network. The third and fourth genes represent discretized codes for the learning rate and momentum with which the network was trained. The fifth gene represented whether implicit within-layer connectivity using a competition algorithm was enabled. Thus, the chromosome shown in the figure would represent a network with 15 nodes in the first hidden layer, 11 nodes in the second hidden layer, a learning rate of 0.5, a momentum of 0.5, and no competition.

genes, so that gene boundaries would be preserved. Inversions could also be modeled, so that exchanged genetic material could be inverted before becoming incorporated into the recipient chromosome. Mutations were implemented by flipping a bit at a binary locus, so that a "0" bit was converted to a "1", or a "1" bit was converted to a "0". Mutations could be implemented either at the "gene" level, where one randomly-determined "hot spot" within each gene was at risk for mutation; or at the "nucleotide" level, where each locus in each gene was at risk. In a separate analysis, binary genes were Gray coded [12] prior to mutation, so that incrementing or decrementing their numerical value by unit amounts would always require only single-bit mutations. The crossover and mutation rates for each generation of chromosomes were controlled by probabilities, which for the purposes of this study were set to 1.0 and 0.1, respectively.

The genetic algorithm was started with 40 randomly generated chromosomes, with gene structures as described above. The genes were decoded, and neural networks with architecture and learning parameters represented by the decoded genes were trained on the training cohort. Phenotypic fitness of each network was measured in two ways: as its mean-square error (MSE), which is the squared difference between network output and pneumonia status averaged across all cases in the cohort; and as its average cross entropy (ENT), where the cross entropy for the ith case is defined as:

$$-(y_i \times \log_e[\text{net}_i] + (1 - y_i) \times \log_e[1 - \text{net}_i])$$

where y_i represents the correct output, and net_i the network output, for the ith case [36]. Networks with lower MSE (or ENT), which implied greater accuracy, were considered to be more fit. Forty networks were probabilistically propagated to the subsequent generation proportionate to their fitness, according to one of three strategies: based on the raw value of their MSE (or ENT); based on the rank of their MSE (or ENT); or based on an elite-selection strategy, in which only the 20 best networks of each generation were allowed to compete (by raw value or rank of MSE (or ENT)) for propagation. Because of the probabilistic selection for propagation under these strategies, any given network could be propagated more than once. In separate analyses, a non-probabilistic cross-generational strategy in which the 20 best networks from the parent generation, and the 20 best networks from the offspring generation, were propagated together was also examined. The chromosomes of propagated networks were then modified by crossing over and mutation, and the modified chromosomes were decoded to provide new parameters for the next round of network evolution. This process of phenotypic fitness measurement, selection, crossover recombination, and mutation was iterated through 50 generations; and the network with the lowest MSE (or ENT) in the final generation was designated as the optimal evolved network.

All genetic algorithms and ANN were programmed in Mathematica [40], based on previously described methods [13,20,24].

2.4. Data analysis

The accuracy of optimal networks within the training and testing cohort was measured as the area under a receiver-operating characteristic (ROC) curve, determined by the method of maximum likelihood using a binormal approximation [7,19]. The area under a ROC

curve ranges from 0.5 and 1.0, where a value of 1.0 represents perfect discriminatory accuracy, and a value of 0.5 represents accuracy no better than chance. Comparisons of ROC curve areas between the training and testing cohort were made using uncorrelated estimates of ROC area and their standard errors [18]. Sensitivities, specificities, and classification accuracies of network outputs for pneumonia, along with 95% confidence intervals (CI), were determined for several output thresholds [11].

3. Results

Using a baseline strategy of recombinant crossovers across gene boundaries, mutation at the gene level, a mutation rate of 0.1, and selection according to rank of MSE, the optimal evolved network had nine nodes in the first hidden layer, zero nodes in the second hidden layer, a learning rate of 0.5, a momentum of 0.5, and no within-layer competition connectivity (Table 2). The MSE of this optimal network was 0.079 in the training cohort and 0.185 in the testing cohort (Table 3). The ROC curve area of this network in the training cohort was 0.872, and in the testing cohort was 0.934 (P-value for difference, 0.181). Using an output cutoff value of 0.025, this network had a sensitivity of 0.917 (95% CI, 0.853–0.956), a specificity of 0.501 (95% CI, 0.466–0.537), and an accuracy of 0.562 (95% CI, 0.529–0.595), in the training cohort; and a sensitivity of 0.902 (95% CI, 0.759–0.968), a specificity of 0.493 (95% CI, 0.377–0.610), and an accuracy of 0.638 (95% CI, 0.543–0.724), in the testing cohort. Using a higher cutoff value of 0.25, the network had a sensitivity of 0.617 (95% CI, 0.528–0.698), a specificity of 0.930 (CI, 0.909–0.947), and an accuracy of 0.884 (95% CI, 0.861–0.904), in the training cohort; and a sensitivity of 0.634 (95% CI, 0.469–0.774), a specificity of 0.933 (95% CI, 0.845–0.975), and an accuracy of 0.828 (95% CI, 0.744–0.889), in the testing cohort. Although the optimal network was not represented by any of the network chromosomes in the initial (0th) generation, by the 50th generation it was represented by 18 of the 40 network chromosomes.

Table 2 also shows optimal parameters for networks evolved under other genetic strategies. Table 3 shows accuracy measures for these optimal networks in the training and testing cohorts. Optimal networks evolved using several other strategies, including crossovers within (rather than across) gene boundaries, Gray coding of binary genes prior to mutation, and non-probabilistic cross-generational selection, evolved optimized ANN identical to the baseline strategy. A strategy of implementing mutation at the nucleotide rather than at the gene level also evolved an accurate network, with an ROC area of 0.851 in the training cohort and of 0.936 in the testing cohort (P-value for difference, 0.071). Other genetic strategies, including inversions of chromosome material during recombination (training ROC area 0.812), elite selection of best networks within generations (training ROC area 0.819), and propagation based on raw value rather than rank of network error (training ROC area 0.812) evolved less accurate ANN.

We also examined the effect of modifying the network-training components of the algorithm, and the use of different starting populations of chromosomes. Training of networks using a cross entropy approximation to maximum likelihood, rather than mean-square error, resulted in the same or similar optimal networks under most strategies (Table 2). Training of networks using individual split-sample cross-validation for each

Table 2
Network parameters for neural networks evolved under different genetic strategies

Genetic strategy[a]		Optimal neural network parameters				
Crossover/mutation strategy	Selection criteria	Nodes in hidden layer 1	Nodes in hidden layer 2	Learning rate	Momentum	Within-layer competition
Across genes	MSE rank	9	0	0.5	0.5	No
	ENT rank	9	0	0.5	0.5	No
Within genes	MSE rank	9	0	0.5	0.5	No
	ENT rank	9	0	0.5	0.5	No
Across genes	MSE rank, elite[b]	8	11	0.5	0.5	No
	ENT rank, elite	8	13	0.5	0.5	No
Across genes	MSE rank, Gray code[c]	9	0	0.5	0.5	No
	ENT rank, Gray code	15	14	0.5	0.5	No
Across genes	MSE rank, cross-generation[d]	9	0	0.5	0.5	No
	ENT rank, cross-generation	9	0	0.5	0.5	No
Across genes	MSE value	13	13	0.5	0.05	No
	ENT value	4	0	0.5	0.1	No
Across genes, inversions[e]	MSE rank	15	14	0.5	0.5	No
	ENT rank	15	14	0.5	0.5	No
Across genes, mutation rate 0.01	MSE rank	9	0	0.5	0.5	No
	ENT rank	9	0	0.5	0.5	No
Across genes, mutation at nucleotide level	MSE rank	5	0	0.5	0.5	No
	ENT rank	5	0	0.5	0.05	No

[a] All genetic strategies used a crossover rate of 1.0, a mutation rate of 0.1, and mutation at the gene level, unless otherwise specified. MSE: mean-square error. ENT: average cross entropy.

[b] For the elite strategy, only the 20 best networks of each generation were allowed to compete for propagation to subsequent generations.

[c] Binary genes were Gray coded prior to mutation, so that incrementing or decrementing their numerical value by 1 unit required only a single-bit mutation.

[d] For the cross-generational strategy, the 20 best networks from the parent generation, and the 20 best networks from the offspring generation, were propagated together.

[e] During crossovers, exchanged genetic material was inverted before being incorporated into the recipient chromosome.

Table 3
Accuracy measures for neural networks evolved under different genetic strategies

Genetic strategy[a]		Optimal neural network accuracy measures				
Crossover/mutation strategy	Selection criteria	MSE training cohort	MSE testing cohort	ROC area training cohort	ROC area testing cohort	*P*-value for difference in ROC area
Across genes	MSE rank	0.079	0.185	0.872	0.934	0.181
	ENT rank	0.285	0.609	0.872	0.934	0.181
Within genes	MSE rank	0.079	0.185	0.872	0.934	0.181
	ENT rank	0.285	0.609	0.872	0.934	0.181
Across genes	MSE rank, elite[b]	0.086	0.169	0.819	0.858	0.576
	ENT rank, elite	0.301	0.541	0.834	0.911	0.120
Across genes	MSE rank, Gray code[c]	0.079	0.185	0.872	0.934	0.181
	ENT rank, Gray code	0.293	0.535	0.812	0.903	0.131
Across genes	MSE rank, cross-generation[d]	0.079	0.185	0.872	0.934	0.181
	ENT rank, cross-generation	0.285	0.609	0.872	0.934	0.181
Across genes	MSE value	0.089	0.183	0.812	0.899	0.144
	ENT value	0.304	0.576	0.806	0.873	0.337
Across genes, inversions[e]	MSE rank	0.084	0.173	0.812	0.903	0.131
	ENT rank	0.293	0.535	0.812	0.903	0.131
Across genes, mutation rate 0.01	MSE rank	0.079	0.185	0.872	0.934	0.181
	ENT rank	0.285	0.609	0.872	0.934	0.181
Across genes, mutation at nucleotide level	MSE rank	0.080	0.171	0.851	0.936	0.071
	ENT rank	0.296	0.571	0.847	0.884	0.547

[a] All genetic strategies used a crossover rate of 1.0, a mutation rate of 0.1, and mutation at the gene level, unless otherwise specified. MSE: mean-square error. ENT: average cross entropy.

[b] For the elite strategy, only the 20 best networks of each generation were allowed to compete for propagation to subsequent generations.

[c] Binary genes were Gray coded prior to mutation, so that incrementing or decrementing their numerical value by 1 unit required only a single-bit mutation.

[d] For the cross-generational strategy, the 20 best networks from the parent generation, and the 20 best networks from the offspring generation, were propagated together.

[e] During crossovers, exchanged genetic material was inverted before being incorporated into the recipient chromosome.

network, rather than using a fixed number of training epochs across networks, yielded optimal ANN similar to the baseline strategy, except for a momentum of 0.05 (instead than 0.5) under MSE, and for a first hidden-layer node count of 11 (instead of 9) under ENT. Training of networks using different random initializations of the weight matrices also resulted in similar optimal ANN, except for a first hidden-layer node count of eight (instead of nine) under both MSE and ENT. Training of networks using different random orderings of training cases within epochs resulted in an optimal ANN identical to the baseline strategy under MSE, but a different optimal ANN (with 15 nodes in the first hidden layer, and 14 nodes in the second hidden layer) under ENT. Finally, starting the genetic algorithm with a different population of 40 chromosomes resulted in evolution of an optimal ANN identical to that derived in the baseline analyses.

4. Discussion

ANN have been used to predict several medical conditions [9,21,33,38], in some cases more accurately than standard statistical methods such as logistic regression [28]. However, the accuracy of ANN may be highly sensitive to the nodal architectures, learning rates, and momentum parameters used to structure and train them [6,15,34]. Strategies for determining optimal network structures for a given data set are largely empirical [1], and exhaustive search strategies to find the best network for the data in general cannot be done. Our results show that genetic algorithms [12,14,23], based on evolutionary principles, can be used to evolve ANN with optimized parameters for predicting community-acquired pneumonia. ANN evolved using genetic algorithms discriminated pneumonia from other respiratory conditions accurately within a training cohort, and within an independent testing cohort consisting of cases on which the networks had not been trained. Operating at a low network-output threshold, these ANN ruled out pneumonia with high sensitivity, although the specificity to rule in pneumonia was more limited. Similar evolutionary algorithms have been used in other medical domains to derive ANN to predict response to warfarin [32], depression after mania [26], falls among the elderly [2], outcome in critical illness [8], and outcome after surgery for lung [25] and prostate [35] carcinoma.

Genetic algorithms work by allowing neural networks to compete with each other for survival based on their predictive accuracy, and by providing for continuous innovation in network structures through crossover recombination and mutation. In this way, genetic algorithms in effect search among the space of possible networks to evolve those adapted to the prediction problem at hand. We found that several genetic strategies, based on different implementations of crossover recombination, mutation, and fitness-based selection, evolved accurate networks for predicting pneumonia, and that some strategies tended to evolve more accurate networks than others. For example, strategies that allowed all networks from a generation to compete probabilistically for propagation, or that propagated best parent and children networks together, tended to evolve more accurate networks than strategies based on elite selection, in which only the best networks of each generation were allowed to compete probabilistically for propagation. Strategies based on mutation acting with constrained probabilities, operating at the nucleotide or the gene level, or operating on coded genes to allow for small changes in gene values, also appeared to evolve

more accurate networks. Whether similar strategies would evolve more accurate neural networks when applied to other data and prediction problems will require further study.

Although in many cases different genetic procedures evolved the same optimal ANN for predicting pneumonia, in some cases different procedures evolved somewhat different ANN. This reinforces the fact that genetic algorithms, while finding good solutions to a prediction problem, may not find the single best solution representing a global minimum on a parameter error surface, since an exhaustive combinatorial search has not been performed [26]. Nevertheless, the results of the different genetic procedures allow some unifying conclusions to be drawn about optimized networks for predicting pneumonia using our data. First, the most accurate optimized networks had only one hidden layer, with approximately nine hidden neurons on the layer. Because network training time rises with the number of layers, and the number of nodal connections, such single-layer networks represent efficient solutions to the pneumonia prediction problem. Second, learning rate and momentum parameters (governing the output and hidden layer weight updates) of 0.5 generated more optimal networks than other values tested. And third, establishment of implicit within-layer connectivity with a competition algorithm was never optimal when combined with any nodal architecture, or any training parameter. Although such competition functions have improved network performance in some studies [6], this did not appear to be the case with our data.

Although genetic algorithms have provided solutions for optimization problems in many domains, the theoretic mechanisms underlying their success are not completely defined [12]. Some theories suggest that genetic algorithms identify sparse patterns (''schema'') of mathematical alleles in search space, exponentially multiplying those with better-than-average fitness, and similarly suppressing those with less-than-average fitness [23]. In this synthesis, crossovers play a critical role, because they bring together low-order schema with few defined genetic loci to form higher-order schema with more defined loci, which can be tested for fitness and multiplied or suppressed [23]. Mutation becomes the primary source of innovation [24], re-directing the search to areas of schema space that might not have already been sampled. Although mutation may in some cases prevent premature convergence on local optima in search space, it is by no means guaranteed to do so [24].

Although our algorithm encoded network structure using discreet genes to represent the number of nodes per hidden layer, similar to the methods of others [8], there are other mechanisms for accomplishing this. For example, Jefferson et al. [26] encoded network architectures as mathematical chromosomes representing connections between nodes, applied crossover and mutation to these constructs, and then pruned networks that were not fully interconnected. In this manner, network configurations and best predictor variables were simultaneously evolved. In addition, genetic algorithms are not the only search method for determining optimal network structures. Greedy algorithms [17], in which satisficing solutions to segments of the network-search problem are pieced together to derive close-to-optimal networks, and simulated annealing [27], in which network structures are perturbed in a manner analogous to melting and optimal parameters are allowed to slowly ''freeze out'', are alternative methods for approaching this problem.

There are some limitations to our study that deserve comment. The definition of pneumonia that we used was based on chest X-ray findings alone, and did not incorporate microbiologic or serologic information [22]. As a result, distinction among cases of

bacterial, atypical bacterial, and viral pneumonia could not be made based on the data. It is possible that optimal networks for predicting community-acquired pneumonia may differ, depending on microbial etiology. In addition, because pulmonary infiltrates are an imperfect gold standard for pneumonia, it is possible that some cases classified as having pneumonia in reality had other non-infectious illnesses. Nevertheless, physicians' management decisions, such as those concerning anti-microbial therapy and hospitalization, are often based on radiographic evidence for pneumonia, making it a clinically relevant standard for neural networks, and other classifier systems, to predict [31]. We excluded patients with a radiographic classification of possible pneumonia during training, and structured our networks with one output node, to classify patients as having either no pneumonia, or probable or definite pneumonia. Had we structured our networks with three output nodes, to classify patients as having no pneumonia, possible pneumonia, or probable or definite pneumonia, networks with different hidden layer architectures, and different training parameters, may have evolved.

Another limitation is that in many network comparisons, ROC areas in the testing cohort exceeded those in the training cohort, suggesting better discriminatory accuracy of evolved networks in new cases compared with those on which they had been trained. In previous studies using judgment (lens model) analysis [5,16], we showed that pneumonia cases tended to be more ''detectable'' in the testing than in the training cohort, largely because testing cohort cases were more typical of community-acquired pneumonia [37]. Although this may explain the greater accuracy of the evolved networks in the testing cohort, it makes it more difficult to exclude possible degradation in their accuracy when applied to other cohorts with more atypical pneumonia cases. Further study of these networks in other independent testing cohorts will be necessary to better validate their discriminatory accuracy.

There are also some limitations to the genetic algorithm and network training. Although we explored several genetic strategies, there are others, including those using multiple crossovers, deletion or duplication of genetic material, and recombination of diploid chromosomes with modeling of allelic dominance, that might have yielded different results. Strategies using different crossover or mutation rates than the ones we used might have also yielded different results. In addition, we started with a population of 40 random networks, and evolved these networks through 50 generations, regardless of intra-generational network performance. We cannot exclude the possibility that a different number of networks, evolved using alternative generational strategies, would have yielded different results. We initialized the weight matrices of all networks using the same random seed, and processed all training cases in the same order, to allow for a more fair comparison between networks. Although our analyses suggest that the genetic algorithm was in general robust to changes in these training parameters, it is possible that use of other starting weights, and other stochastic samplings of training cases, might have yielded different optimized ANN [15,34]. We used early-stopping cross-validation techniques [4,34] to guard against over-fitting during network training. Use of other methods to prevent over-fitting, such at network regularization using weight decay [4,29], or the addition of noise to the training patterns [4], might have also yielded different ANN.

Finally, it should be noted that all neural network genetic algorithms, including our own, are very time intensive, since many networks must be trained during each generation, and

many generations must be sampled. On a Pentium III 733 MHz computer running Mathematica 3.0, each generation of 40 networks required approximately 30 min to process. Development of more time-efficient methods for implementing genetic algorithms for neural networks will be necessary to make these methods more widely useful for structuring networks, selecting optimal inputs, and other ANN-related applications.

In conclusion, genetic algorithms using a variety of coding, recombination, mutation, and fitness-based selection strategies evolved ANN with parameters optimized for predicting community-acquired pneumonia. Neural networks evolved using these algorithms discriminated pneumonia from other respiratory conditions accurately within a training cohort, and within a testing cohort consisting of cases on which the networks had not been trained. Our results show that genetic algorithms can be successfully used to adapt ANN to the prediction of pneumonia among patients with acute respiratory complaints. Such algorithms should be further studied on other clinical prediction problems, and in other medical domains.

References

[1] Astion ML, Wilding P. The application of backpropagation neural networks to problems in pathology and laboratory medicine. Arch Pathol Lab Med 1992;116:995–1001.
[2] Bath PA, Pendleton N, Morgan K, Clague JE, Horan MA, Lucas SB. New approach to risk determination: development of risk profile for new falls among community-dwelling older people by use of a genetic algorithm neural network (GANN). J Gerontol Med Sci 2000;A55:M17–21.
[3] Baxt WG. Use of an artificial neural network for the diagnosis of myocardial infarction. Ann Intern Med 1991;115:843–8.
[4] Bishop CM. Neural networks for pattern recognition. New York: Oxford University Press; 1995. pp. 338–49.
[5] Brunswik E. Perception and the representative design of psychological experiments. Berkeley, California: University of California Press; 1956.
[6] Cho S, Reggia JA. Multiple disorder diagnosis with adaptive competitive neural networks. Artif Intell Med 1993;5:469–87.
[7] Dorfman DD, Alf Jr. E. Maximum likelihood estimation of parameters of signal detection theory and determination of confidence intervals—rating-method data. J Math Psych 1969;6:487–96.
[8] Dybowski R, Weller P, Chang R, Gant V. Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm. Lancet 1997;347:1146–50.
[9] El-Solh AA, Hsiao CB, Goodnough S, Serghani J, Grant BJ. Predicting active pulmonary tuberculosis using an artificial neural network. Chest 1999;116:968–73.
[10] Faussett L. Fundamentals of neural networks: architectures, algorithms, and applications. Englewood Cliffs, New Jersey: Prentice Hall; 1994.
[11] Fleiss JL. Statistical methods for rates and proportions. 2nd ed. New York: Wiley; 1981.
[12] Forrest S. Genetic algorithms: principles of natural selection applied to computation. Science 1993;261:872–8.
[13] Freeman JA. Simulating neural networks with Mathematica. Reading, Massachusetts: Addison-Wesley; 1994.
[14] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. 1st ed. Reading, Massachusetts: Addison-Wesley; 1989.
[15] Gottschalk A, Hyzer C, Geer RT. A comparison of human and machine-based predictions of successful weaning from mechanical ventilation. Med Decis Making 2000;20:160–9.
[16] Hammond KB, Hursch CJ, Todd FJ. Analyzing the components of clinical inference. Psychol Rev 1964;71:528–30.

[17] Handels H, Roá T, Kreusch J, Wolff HH, Poppl SJ. Feature selection for optimized skin tumor recognition using genetic algorithm. Artif Intell Med 1999;16:283–97.
[18] Hanley JA, McNeil BJ. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. Radiology 1983;148:839–43.
[19] Heckerling PS. Parametric receiver operating characteristic (ROC) curve analysis using Mathematica. Comput Meth Prog Biomed 2002;69:65–73.
[20] Heckerling PS, Gerber BS. Feed-forward, back-propagation neural network modeling using Mathematica. Comput. Meth. Prog. Biomed., submitted for publication.
[21] Heckerling PS, Gerber BS, Tape TG, Wigton RS. Prediction of community-acquired pneumonia using artificial neural networks. Med Decis Making 2003;23:112–21.
[22] Heckerling PS, Tape TG, Wigton RS, et al. Clinical prediction rule for pulmonary infiltrates. Ann Intern Med 1990;113:664–70.
[23] Holland JH. Adaptation in natural and artificial systems. Ann Arbor, Michigan: University of Michigan Press; 1975. Reprinted: Cambridge, Massachusetts: MIT Press; 1992.
[24] Jacob C. Illustrating evolutionary computation with Mathematica. San Diego, California: Academic Press; 2001.
[25] Jefferson MF, Pendleton N, Lucas SB, Horan MA. Comparison of a genetic algorithm neural network with logistic regression in predicting outcome after surgery for patients with non-small cell lung carcinoma. Cancer 1997;79:1338–42.
[26] Jefferson ME, Pendleton N, Lucas CP, Lucas SB, Horan MA. Evolution of artificial neural network architecture: prediction of depression after mania. Meth Inform Med 1998;37:220–5.
[27] Kirkpatrick S, Gelatt Jr. CD, Vecchi MP. Optimization by simulating annealing. Science 1983;220:671–80.
[28] Lette J, Colletti BW, Cerino M, et al. Artificial intelligence versus logistic regression statistical modeling to predict cardiac complications after non-cardiac surgery. Clin Cardiol 1994;17:609–14.
[29] Lisboa PJG. A review of evidence of health benefit from artificial neural networks in medical intervention. Neural Networks 2002;15:11–39.
[30] McClelland JL, Rumelhart DE. Training hidden units. In: McClelland JL, Rumelhart DE, editors. Explorations in parallel distributed processing. Cambridge, MA: MIT Press; 1988. pp. 121–160.
[31] Metlay JP, Fine MJ. Testing strategies in the initial management of patients with community-acquired pneumonia. Ann Intern Med 2003;138:109–18.
[32] Narayanan MN, Lucas SB. A genetic algorithm to improve a neural network to predict a patient's response to warfarin. Meth Inform Med 1993;32:55–8.
[33] Patil S, Henry JW, Rubenfire M, Stein PD. Neural network in the clinical diagnosis of acute pulmonary embolism. Chest 1993;104:1685–9.
[34] Penny W, Frost D. Neural networks in clinical medicine. Med Decis Making 1996;16:386–98.
[35] Potter SR, Miller MC, Mangold LA, Jones KA, Epstein JI, Veltri RW, et al. Genetically engineered neural networks for predicting prostate cancer progression after radical prostatectomy. Urology 1999;54:791–5.
[36] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propogation. In: Rumelhart DE, McClelland JL, editors. Parallel distributed processing: explorations in the microstructure of cognition. Cambridge, MA: MIT Press; 1986. pp. 318–64.
[37] Tape TG, Heckerling PS, Ornato JP, Wigton RS. Use of clinical judgment analysis to explain regional variations in physicians' accuracies in diagnosing pneumonia. Med Decis Making 1991;11:189–97.
[38] Tourassi GD, Floyd CE, Sostman HD, Coleman RE. Acute pulmonary embolism: artificial neural network approach for diagnosis. Radiology 1993;189:555–8.
[39] Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression in predicting medical outcomes. J Clin Epidemiol 1996;49:1225–31.
[40] Wolfram S. Mathematica: a system for doing mathematics by computer. 2nd ed. Reading, Massachusetts: Addison-Wesley; 1991.