



PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com

POWERED BY SCIENCE @ DIRECT®

Neural Networks 16 (2003) 865–871

Neural
Networks

www.elsevier.com/locate/neunet

2003 Special issue

Divide-and-conquer approach for brain machine interfaces: nonlinear mixture of competitive linear models

Sung-Phil Kim^{a,*}, Justin C. Sanchez^a, Deniz Erdogmus^a, Yadunandana N. Rao^a,
Johan Wessberg^b, Jose C. Principe^a, Miguel Nicolelis^b

^aDepartment of Electrical Engineering, University of Florida, Gainesville, FL 32611, USA

^bDepartment of Neurobiology, Duke University, Durham, NC 27710, USA

Abstract

This paper proposes a divide-and-conquer strategy for designing brain machine interfaces. A nonlinear combination of competitively trained local linear models (experts) is used to identify the mapping from neuronal activity in cortical areas associated with arm movement to the hand position of a primate. The proposed architecture and the training algorithm are described in detail and numerical performance comparisons with alternative linear and nonlinear modeling approaches, including time-delay neural networks and recursive multilayer perceptrons, are presented. This new strategy allows training the local linear models using normalized LMS and using a relatively smaller nonlinear network to efficiently combine the predictions of the linear experts. This leads to savings in computational requirements, while the performance is still similar to a large fully nonlinear network.

© 2003 Published by Elsevier Science Ltd.

Keywords: Brain machine interfaces; Multiple local linear models; Competitive learning

1. Introduction

Brain machine interfaces (BMI) are a developing technology that aims to directly transfer the subject's intent of movement to a machine. Our goal is to engineer devices that are able to interpret neural activity originating in the motor, premotor, and posterior parietal cortices and generate accurate predictions of hand position. We envision the application of these devices in BMI, where movement can be restored in individuals suffering from neurological disorders.

Recently, a number of groups have demonstrated that linear and nonlinear adaptive system identification approaches can lead to BMI that effectively predict the hand position of primates using only neural signals (Chapin, Moxon, Markowitz, & Nicolelis, 1999; Moran & Schwartz, 1999; Serruya, Hatsopoulos, Paninski, Fellows, & Donoghue, 2002; Wessberg et al., 2000). The topologies studied thus far include FIR filters, time-delay neural networks (TDNN), Kalman filter and extensions, and recursive multilayer perceptrons (RMLP). In the BMI experimental paradigm, large arrays of microelectrodes are implanted in the cortical areas listed above and the corresponding neural

activity is recorded. Spike-detection and sorting algorithms are used to process these analog potentials to determine the firing times of single neurons. Typically, the spike-time information is summarized into bin counts using short windows (100 ms in this paper). In all previous work, the approach had been to try to determine a functional/dynamical mapping between these bin counts and the hand position. In this paper, we follow the same approach.

An important consideration in designing BMI is the feasibility of the approach taken. The target applications necessitate real-time implementations with minimal computational and hardware requirements. On one hand, linear models are usually the best in terms of their computational requirements. On the other hand, a simple linear model is often insufficient to accurately capture the complex input–output relationships between neural activity and hand position. Recently, our group has conducted a performance comparison between linear and nonlinear modeling approaches, and the latter was found to be favorable (Sanchez et al., 2002).

In this paper, our aim is to demonstrate that the target input–output mapping can be discovered using a *divide-and-conquer* approach. In this approach, we combine the simplicity of training linear models with the performance boost that can be achieved by nonlinear methods.

* Corresponding author.

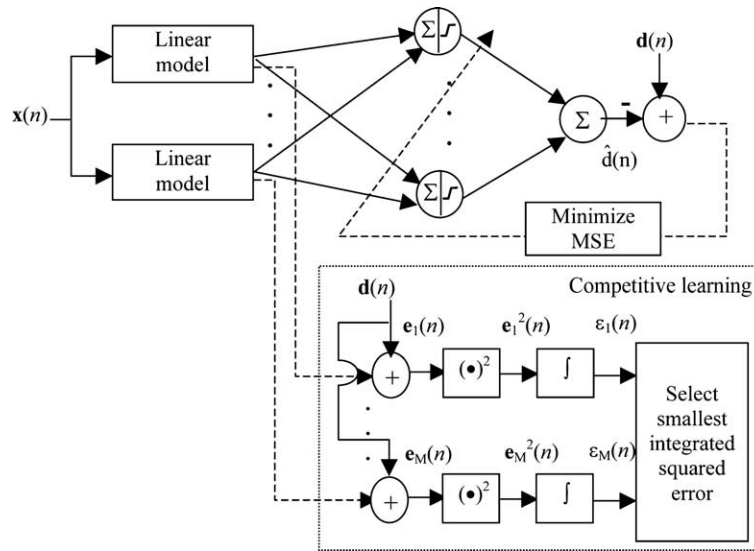


Fig. 1. An overall architecture of the proposed model. The box illustrates the selection of the winner using integrated squared errors from each linear model. Outputs from M trained linear models are fed to a MLP, which is trained using the conjugate gradient algorithm. $d(n)$ denotes the desired response.

Specifically, a two-stage structure is used where the first stage consists of a bank of competitively trained linear filters and the second stage consists of a single hidden layer MLP (see Fig. 1). Performance comparisons with simple linear, TDNN, and RMLP approaches are provided.

2. The models

In this section, we describe the modeling approaches that will be considered in the experimental comparisons of study. The main focus will be on the nonlinear mixture of competitive linear models (NMCLM). However, a brief description of the other three topologies will also be provided for completeness.

2.1. Nonlinear mixture of competitive linear models

The overall architecture is identical to a single nonlinear layer TDNN. However, the training procedure undertaken here is significantly different. This modeling method uses the divide-and-conquer approach. Our reasoning is that, a complex nonlinear modeling task can be elucidated by dividing it into simpler linear modeling tasks and combining them properly (Farmer & Sidorowich, 1987). Previously, this approach was successfully applied to nonstationary signal segmentation, assuming that a nonstationary signal is a combination of piecewise stationary signals (Fancourt & Principe, 1996). Hypothesizing that the neural activity will demonstrate varying characteristics for different localities in the space of the hand trajectories, we expect the multiple model approach, in which each linear model specializes in a local region, to provide a better overall input–output mapping. However, here the problem is different since

the goal is not to segment a signal but to segment the joint input/desired signal space.

The local linear models can be conceived as a committee of experts each specializing in one segment of the hand trajectory space. The MLP is introduced to the system in order to nonlinearly combine the predictions generated by all the linear models. For BMI applications, it has the further advantage that it does not require a selection scheme in testing, which can only be accurately done using the desired output. For example, in a prosthetic arm application, the desired hand position is not available in practice.

The topology allows a two-stage training procedure that can be performed sequentially in off-line training: first, competitive learning for the local linear models and then error backpropagation learning for the MLP. It is important to note that in this scheme, both the linear models and the MLP are trained to approximate the same desired response, which is the hand trajectory of the primate.

The training of the multiple linear models is accomplished by competitively (hard or soft) updating their weights in accordance with previous approaches using the normalized-least-mean-squares (NLMS) algorithm (Haykin, 1996). The winning model is determined by comparing the (leaky) integrated squared errors of all competing models and selecting the model that exhibits the least integrated error for the corresponding input (Fancourt & Principe, 1996). In competitive training, the leaky integrated squared error for the i th model is given by

$$\varepsilon_i(n) = (1 - \mu)\varepsilon_i(n-1) + \mu e_i^2(n), \quad i = 1, \dots, M \quad (1)$$

where M is the number of models, and μ is the time constant of the leaky integrator. If hard competition is employed, then only the weight vector of the winning model is updated. Specifically, the hard competition update rule for the weight

vector of the winning model is

$$\mathbf{w}_{\text{winner}}(n+1) = \mathbf{w}_{\text{winner}}(n) + \frac{\eta e_{\text{winner}}(n) \mathbf{x}(n)}{\gamma + \|\mathbf{x}(n)\|^2} \quad (2)$$

where $\mathbf{w}_{\text{winner}}$ is the weight vector, $\mathbf{x}(n)$ is the current input, $e_{\text{winner}}(n)$ is the instantaneous error of the winning model, η is the learning rate and γ is the small positive constant. If soft competition is used, a Gaussian weighting function centered at the winning model is applied to all competing models.

Every model is then updated proportional to the weight assigned to that model by this Gaussian weighting function

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \frac{\eta(n) A_{i,j}(n) e_i(n) \mathbf{x}(n)}{\gamma + \|\mathbf{x}(n)\|^2}, \quad (3)$$

$$i = 1, \dots, M$$

where \mathbf{w}_i is the weight vector of the i th model, j th model is the winner, and $A_{i,j}(n)$ is the weighting function

$$A_{i,j}(n) = \exp\left(-\frac{d_{i,j}^2}{2\sigma^2(n)}\right) \quad (4)$$

where $d_{i,j}$ is the Euclidean distance between index i and j , which is equal to $|j - i|$, $\eta(n)$ is the annealed learning rate, and $\sigma^2(n)$ is the kernel width, which decreases exponentially as n increases. The learning rate also decreases exponentially with time.

Soft competition preserves the topology of the input space, updating the models neighboring the winner; thus it is expected to result in smoother transitions between models specializing in topologically neighboring regions (of the state space). However, in the experimental results that will be provided in Section 3, we will utilize the hard competition rule since comparisons on the data sets utilized in these experiments did not show any significant difference in generalization performance (possibly due to the nature of the data set used in these experiments).

The competitive training of the first layer of linear models to match the hand trajectory using the neural activity creates a set of basis signals from which the following single hidden layer MLP can generate a more accurate overall prediction than any one of the individual competing linear models (thus the name nonlinear mixture of competitive linear models).

2.2. Alternative modeling approaches

The rich variety of alternative modeling approaches for BMI design is evident from the relevant literature and the works of independent groups. In this paper, results are compared with the following three approaches: single moving average (MA) model, single hidden layer TDNN, single recursive hidden layer RMLP. Below, a brief description of each topology is provided for the sake of completeness.

Single moving average model. This approach tries to model the mapping from neural activity to hand position using a linear combiner, which is structurally equivalent to each one of the multiple local linear models used in the competitive framework of Section 2.1. It creates a prediction of the hand position by a product of its weight matrix \mathbf{W} and its input vector $\mathbf{x}(n)$ at time n , i.e. $\hat{\mathbf{d}}(n) = \mathbf{W}^T \mathbf{x}(n)$. The input vector consists of L lags of the bin counts for each of the neurons that measurements are taken from. The desired hand trajectory $\mathbf{d}(n)$, depending on the experiment, has one, two, or three dimensions. Optimal weights are determined either from the Wiener-Hopf equations or by gradient descent learning (Haykin, 1996).

Time-delay neural network. This approach models the mapping from neural activity to hand position using a nonlinear combination of delayed bin counts from every neuron. The memory is preset to the length of the tap-delay lines used in the input layer. The single hidden layer uses sigmoid nonlinearities. The output of the TDNN is given by $\hat{\mathbf{d}}(n) = \mathbf{W}_2 f(\mathbf{W}_1^T \mathbf{x}(n) + \mathbf{b}_1) + \mathbf{b}_2$, where \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 , \mathbf{b}_2 are the weights of the TDNN that are trained with error backpropagation (Haykin, 1994).

Recurrent multilayer perceptron. This network is built from a hidden layer of nonlinear processing elements (PEs) with state feedback and linear outputs. Different from a TDNN, the input vector $\mathbf{x}(n)$ of this network consists only of the instantaneous values of the bin counts from the neurons. In addition, its hidden layer has a fully connected feedback to itself with a feedback weight matrix \mathbf{W}_f , which is capable of adjusting its time-resolution to various levels depending on the requirements of the optimization criterion. The output of the RMLP is obtained by the following recursive equation: $\hat{\mathbf{d}}(n) = \mathbf{W}_2 f(\mathbf{W}_1 \mathbf{x}(n) + \mathbf{W}_f \mathbf{y}_1(n-1) + \mathbf{b}_1) + \mathbf{b}_2$. The hidden layer state vector is denoted by \mathbf{y}_1 . This network is trained using the backpropagation through time algorithm (Principe, Euliano, & Lefebvre, 2000).

3. Simulations and analysis

In this section, experimental results obtained with the multiple models trained with hard competition are presented and the performance is compared with the single linear model, TDNN, and RMLP.

3.1. Data collection

Synchronous, multichannel neuronal spike trains were collected at Duke University using owl monkeys (*Aotus trivirgatus*). Microwire electrodes were implanted in cortical regions where motor associations are known (Wessberg et al., 2000). The firing times of individual neurons were recorded while the primate performed a 3D reaching task that consists of a reach to food and followed by placing it in the mouth. The primate's hand position was also recorded (with a shared time clock) and digitized with

a 200 Hz sampling rate. The neuronal firings were binned in nonoverlapping windows of 100 ms, which represents the local firing rate for each neuron. These spike counts were directly used as inputs to the model to estimate hand positions. The digitized hand position signal was down-sampled to 10 Hz to synchronize with spike counts in time. In order to take the reaction time into account, the spike trains were delayed by 0.23 s with respect to the hand position.

3.2. Experimental environments

The spike counts from 104 neurons, a set of 20,000 consecutive time bins (2000 s), were used to train the models. After training, the weights of all models are fixed to their *optimal values* identified from the training set and the performance of the models are evaluated on an independent test set of 3000 consecutive bins (300 s).

Single moving average model. The MA model accepts an input vector that consists of 10 delayed versions of the bin counts of 104 neurons, and has three outputs (x, y, z). The 10 tap memory depth (1 s) is based on the suggestion of Wessberg et al. (2000). The model is trained with NLMS with a step size of 0.01.

Nonlinear mixture of competitive linear models. This topology consists of 10 competing linear models for each coordinate (x, y, z) and a single hidden layer MLP with 30 inputs (three coordinates \times 10 models), 30 hidden PEs with tanh nonlinearity, and three linear output PEs to predict the x, y , and z coordinates of the primate's hand. The number of multiple linear models and the number of hidden PEs were chosen empirically by examining the prediction performance on the test set. Each linear model is the same as the MA model. In the competition, the leaky integrator used a forgetting factor of $\mu = 0.1$. This corresponds to an approximate memory depth of 1 s, which is consistent with the memory depth of the linear models (Fancourt & Principe, 1996). The model was trained using NLMS for the local linear models and backpropagation for the MLP, both using a step size of 0.01. Training of the MLP is repeated with 100 random initial conditions and the minimum mean square error (MSE) solution is accepted.

Time-delay neural network. In the experiments, a TDNN (1040:30:3) is trained with the 104 neurons and their 10 delay bin counts as the input. The 30 hidden PEs use tanh nonlinearities and the output have three linear PEs. Effectively, this is the same topology as the competitive linear models and MLP combination, except it is trained completely based on error backpropagation with MSE criterion.

Recursive multilayer perceptron. The RMLP used here is a 104:5:3 network whose input consists of 104 instantaneous bin counts from all neurons. The hidden layer uses nonlinear PEs with tanh activation. Training is performed using Neurosolutions®, where 3 s trajectories are employed for gradient calculations.

3.3. Results

The accuracy of these estimations are evaluated using several measures: the correlation coefficient (CC) is used to quantify the linear relationship between the estimated and actual trajectories; the signal to error ratio (SER) between actual and estimated hand trajectories is used to account for the deficiency of CC in identifying the bias and scaling errors in trajectories; the cumulative error metric (CEM) is used to provide a global statistical picture of how well the models perform in terms of following the actual trajectory; the number of movement *hits* and *misses* a model obtains.

Specifically, the SER is defined as the ratio of the powers of the actual hand trajectory and the error of the estimated trajectory (Sanchez et al., 2002). The CC and SER are computed using 4 s overlapping windows (each reaching movement takes approximately 4 s). The CEM is defined as the probability that the L_2 -norm (or radius in 3D space) of the error vector is less than or equal to some r : $CEM(r) = P(\|e\|_2 \leq r)$ (Sanchez et al., 2003). This is, in fact, the empirical cumulative distribution function of the error radius. During a reaching movement, a model is said to hit the movement if 70% of the time, the instantaneous error norm is less than half of the norm of the hand position vector. Otherwise, the model is said to miss the movement.

The performances of all models are summarized in Table 1 using these measures. In the test set, there are a total of 10 reaching movements. The hit and miss counts are provided in the first two rows. The SER and CC values are segmented in movement and rest. The average and standard deviation of SER and CC for moving and resting portions are reported separately. For a fair comparison, this averaging is performed over only the six common movements that the TDNN, RMLP and multiple models hit.

The z -coordinates of the estimated hand positions with each model for the test set are shown along with the actual hand position in Fig. 2a. Also, in Fig. 2b, we present the changes in SER over time. It is observed that the SER increases when the hand is moving, and decreases when the hand is at rest, for all models. The multiple models achieve

Table 1

Performances of the four models compared (number in parenthesis are the free parameters). The average and the standard deviation were computed for the correlation coefficient (CC) and the SER over samples and three dimensions

Measures	Single MA (3123)	TDNN (31,323)	RMLP (568)	NMCLM (32,253)
No. of hits	2	8	7	7
No. of misses	8	2	3	3
CC: move	0.83 \pm 0.07	0.86 \pm 0.10	0.88 \pm 0.11	0.89 \pm 0.07
SER: move (dB)	4.69 \pm 1.19	6.64 \pm 1.99	7.40 \pm 2.00	7.02 \pm 2.47
CC: rest	0.06 \pm 0.27	0.04 \pm 0.25	0.06 \pm 0.25	0.07 \pm 0.26
SER: rest (dB)	-0.19 \pm 4.27	2.95 \pm 5.69	7.13 \pm 4.01	4.53 \pm 4.99

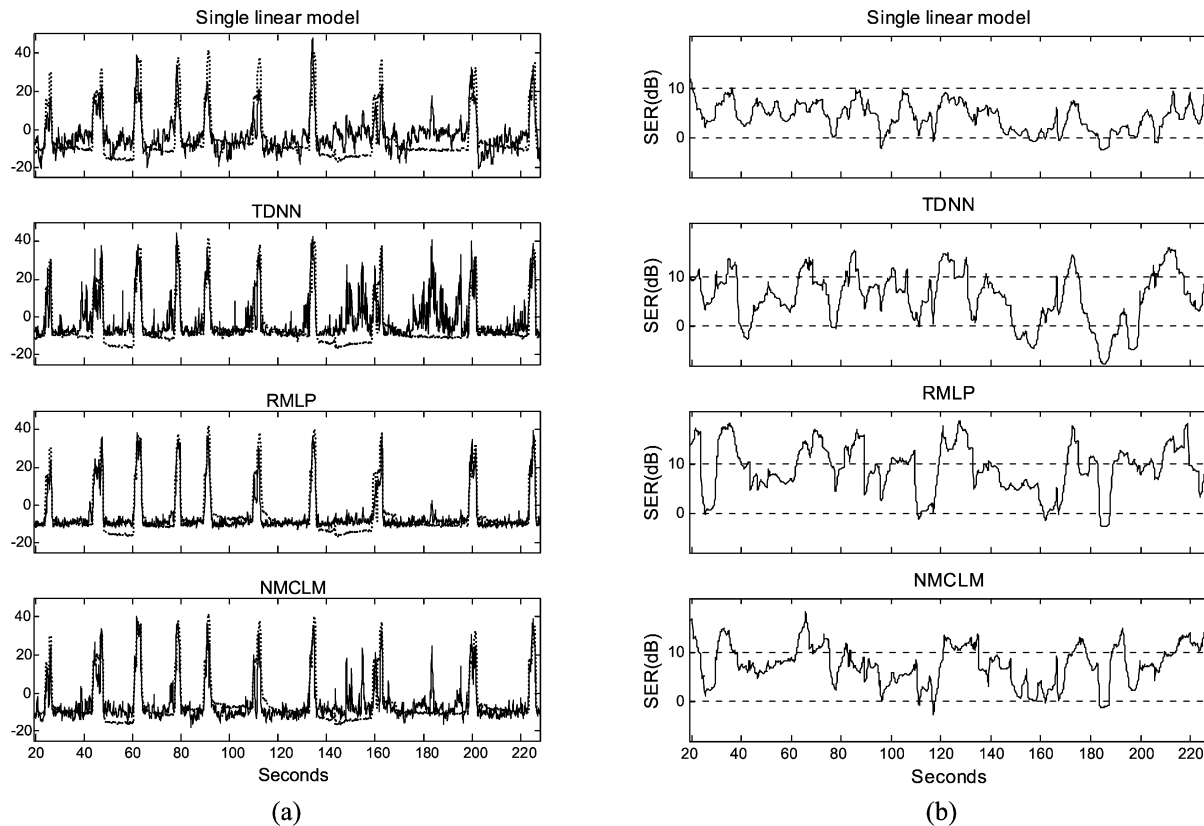


Fig. 2. (a) The z -coordinate of the actual hand trajectory (dotted) and that of the estimated hand trajectories (solid) on testing data for the single linear MA model, TDNN, RMLP, and the NMCLM, respectively. (b) The signal to error ratio (SER) for the z -coordinate in dB using a 4 s sliding window over time.

higher SER levels than the TDNN when the hand is at rest. The RMLP, however, performs better than the multiple models on average.

In Fig. 3, the CEM curves of all models are presented for the entire test set and for only the six common movements. Overall, the RMLP and the multiple models exhibit better performance in these curves compared to the TDNN and the single linear model. On the other hand, when we only consider movements, three models (RMLP, TDNN, and multiple models) produce similar performance.

We further investigate the localization properties of the multiple linear models in the hand trajectory space. In order to demonstrate this, two example reaching movements (taken from the training set) are shown in Fig. 4. For each trajectory, winners at every sample are indicated by different markers. It can be observed that the winning models localize in different regions of the movement. For instance, the 10th model specializes in the portion of the trajectory from mouth to the rest position, while the seventh model learns the trajectory from the rest position to the food.

3.4. Evaluation of training performance

The topology proposed in NMCLM is basically equivalent to a three-layer network: the first layer of weights

consists of the competitive model coefficients, the second and third layer of weights are simply the weights of the following MLP. In this topology, the first hidden layer and the output layer have linear PEs, whereas the second hidden layer has nonlinear PEs. In the NMCLM approach, the first layer weights are trained competitively to predict the desired signal, whereas the MLP is optimized using standard error backpropagation.

In order to quantify the performance of this training procedure from an information-theoretic point-of-view, we evaluate the mutual information (Cover & Thomas, 1991), $I(\mathbf{z}_C, \mathbf{d})$, between the outputs of the competitive models, \mathbf{z}_C , and the desired output, \mathbf{d} . Using a Parzen window estimator for mutual information (Erdogmus, 2002) on 10 segments of the hand trajectory (each of length 1000 samples), the average and standard deviation of $I(\mathbf{z}_C, \mathbf{d})$ is found to be 8.97 nats (± 1.21 nats). The maximum mutual information allowed by this model and data, obtained by estimating $I(\mathbf{z}_C, \hat{\mathbf{d}})$ is 9.83 nats (± 1.19 nats). Percentage-wise, the information contained in the competitive model outputs pertaining to the desired output is thus 92% ($\pm 6\%$). From this, we conclude that the information loss in the first layer is just 8% ($\pm 6\%$).

For comparison, a second network with the same topology is trained as follows: The MLP weights are

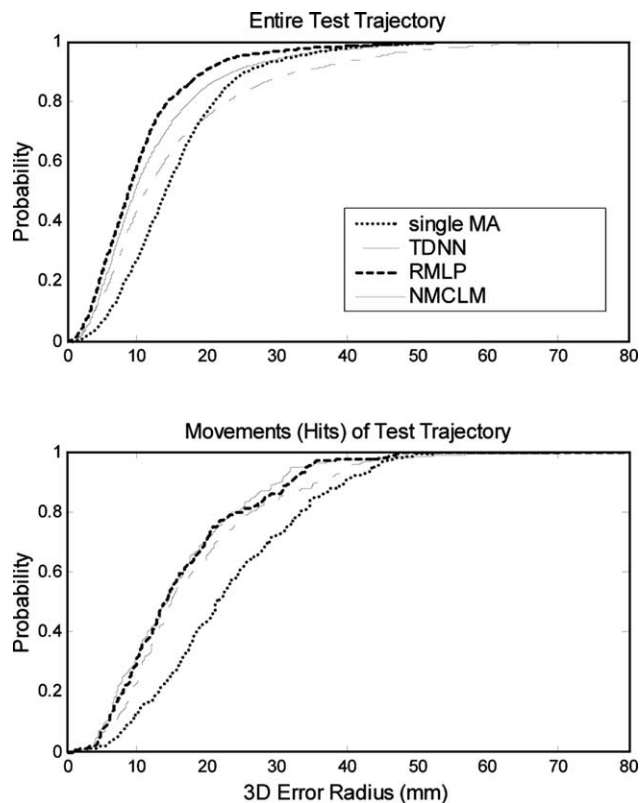


Fig. 3. Comparison of the CEM of the four models evaluated for the entire test set (top), and for the six common reaching movements (bottom).

borrowed from the one trained with competition (in order to ensure *identical* information loss at this stage). The first layer weights are then trained using standard backpropagation through these MLP weights, instead of using competition. This second network, therefore, uses the minimum MSE solution for the first layer weights. Similarly, the mutual information $I(\mathbf{z}_B, \mathbf{d})$ between the output of the first layer of this network \mathbf{z}_B , and the desired output \mathbf{d} is calculated to be 7.42 nats (± 1.35 nats). For this network, the maximum mutual information is 10.90 nats (± 0.40 nats). These correspond to an information-transfer percentage of 68% ($\pm 11\%$). Therefore, the information loss in the first layer of the second network is 32% ($\pm 11\%$).

5. Discussions

In this paper, we have investigated the performance of a nonlinear mixture of competitively trained multiple linear models in predicting the hand position of a primate using neural activity recorded from cortical areas related to arm movement. This approach possesses significant advantages compared to both the single MA model and the standard TDNN, although it is essentially a combination of both. The multiple models follow a divide-and-conquer approach to identify the nonlinear mapping from neuronal data to hand

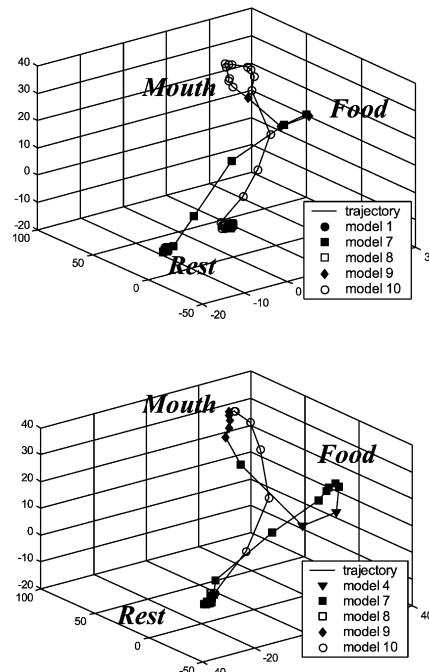


Fig. 4. Demonstration of the localization of multiple linear models for two different reaching movements. Each symbol represents a different winning model during training.

positions. Its performance did not exceed that of an RMLP, probably because the RMLP has the most parsimonious architecture (568 free parameters).

The biggest difficulty with any of these BMI models is the enormous set of free parameters produced by the large neural input vector (104 dimensions). For example, the TDNN and the NMCLM presented above had more than 30,000 weights. Training the first layer of the TDNN with backpropagation is a difficult task due to the attenuation of the error by the hidden layer PEs. By competitively training the first layer of linear models, on the other hand, the training complexity is made more practical, since standard LMS-type linear filter adaptation rules can be employed. However, the problem of large data requirements still remains, and we conclude that we did not have enough data to train these models sufficiently well. The performance of the single MA model is the worst of the group, which means that the constraint of linearity for this specific reaching task is taxing performance (in spite of having ‘only’ 10% of the TDNN weights). From the CEM of Fig. 3, we can conclude that the precision in tracking a trajectory needs further improvements to be useful.

The performance of the NMCLM can be improved further in several respects. First and foremost, pruning the linear model is a viable alternative because not all the neurons are important all the time. A simple test with weight decay on the single MA model shows that we are able to prune less important inputs significantly. For instance, the portion of weights whose magnitudes are less than 0.1,

increases from 22.4 to 36.3%. It consequently helps generalization in testing such that the SER increases to 5.56 dB (± 1.11 dB) (compared to 4.69 ± 1.19 dB) for the movement, and 2.88 dB (± 2.76 dB) for the rest (compared to -0.19 ± 4.27 dB).

In a purely competitive framework, one could use the output generated by only the winning linear model. However, the selection of the winner in the testing phase is not possible since the desired output is not available. Nevertheless, it is interesting to investigate how much performance loss occurs due to the use of the MLP as compared to the selection of the winner alone. To this end, we have calculated the CC and SER for the purely competitive prediction scheme for the same test set used in the experiments (assuming that the desired hand position is available for the selection). For the six reaching movements that have been considered throughout the experiments, the CC is 0.94 (± 0.03) (compared to 0.89 ± 0.07 with the mixture), and the SER is 9.79 dB (± 1.62 dB) (compared to 7.02 ± 2.47 dB with the mixture). Hence, we conclude that there is room for improvement in the selecting of the winner in testing.

From the models presented, the NMCLM is the only one that uses explicit competition to divide the trajectory among models. This is an advantage if we are interested in creating a system that models each piece of the trajectory, as may be required in the future for BMIs with a large repertoire of trajectories. The other approach to segmentation that we are testing is a Hidden Markov Model (HMM) (Darmanjian et al., 2003). For a two segment task (rest and movement), the segmentation performance of the HMM is high, but it still requires lots of data for training (1620 free parameters when the input data is quantized in 128 levels). Each HMM can then be coupled with an MA model and fitting performance can be increased up to 20% with respect to the single MA model (we are in the process of validating this approach). However, it is still unclear how to extend the technique to finer segmentation (e.g. rest to food, food to mouth, mouth to rest).

Overall, the multiple model approach produced better CC and SER than both the TDNN and the single linear filter at a reasonable additional cost in terms of training complexity. Additionally, it produced a CC profile identical to the RMLP. However, the SER of RMLP is on average slightly higher than the multiple models. This can be attributed to the additional error variance that the multiple models exhibit in predicting the hand position at rest. This

may motivate replacing the MLP in the topology with an RMLP in future work.

Acknowledgements

This work is supported by DARPA grant # N-66001-02-C-8022.

References

- Chapin, J. K., Moxon, R. S., Markowitz, M. A., & Nicolelis, M. A. (1999). Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience*, 2(7), 664–670.
- Cover, T., & Thomas, J. (1991). *Elements of information theory*. New York: Wiley.
- Darmanjian, S., Kim, S. P., Nechyba, M. C., Morrison, S., Principe, J. C., & Nicolelis, M. A. (2003). Bimodal brain-machine interface for motor control of robotic prosthetic. Submitted for publication.
- Erdogmus, D. (2002). *Information theoretic learning: Renyi's entropy and its applications to adaptive system training*. PhD Dissertation. Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL.
- Fancourt, C., & Principe, J. C. (1996). Temporal self-organization through competitive prediction. *Proceedings of ICASSP'96*, 3325–3329.
- Farmer, J. D., & Sidorowich, J. J. (1987). Predicting chaotic time series. *Physical Review Letters*, 50, 845–848.
- Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York: McMillan.
- Haykin, S. (1996). *Adaptive filter theory*. Upper Saddle River, NJ: Prentice-Hall.
- Moran, D. W., & Schwartz, A. B. (1999). Motor cortical activity during drawing movements: population representation during spiral tracing. *Journal of Neurophysiology*, 82(5), 2693–2704.
- Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (2000). *Neural and adaptive systems: Fundamentals through simulations*. New York: Wiley.
- Sanchez, J. C., Erdogmus, D., Rao, Y. N., Principe, J. C., Nicolelis, M. A., & Wessberg, J. (2003). Learning the contributions of the motor, premotor, and posterior parietal cortices for hand trajectory reconstruction in a brain machine interface. *IEEE EMBS Neural Engineering Conference*.
- Sanchez, J. C., Kim, S. P., Erdogmus, D., Rao, Y. N., Principe, J. C., Wessberg, J., & Nicolelis, M. A. (2002). Input–output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns. *Proceedings of NNSP'02*, 139–148.
- Serruya, M. S., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., & Donoghue, J. P. (2002). Brain-machine interface: instant neural control of a movement signal. *Nature*, 416, 141–142.
- Wessberg, J., Stambaugh, R., Kralik, J. F., Beck, P. D., Laubach, M., Chapin, J. K., Kim, J., Biggs, J., Srinivasan, M. A., & Nicolelis, M. A. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(6810), 361–365.