Polyhedral Mixture of Linear Experts for Many-To-One Mapping Inversion

A. Karniel, R. Meir, G.F. Inbar

Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000 Israel

Feed-forward control schemes require an inverse mapping of the controlled system. In adaptive systems as well as in biological modeling this inverse mapping is learned from examples. The biological motor control is very redundant, as are many robotic systems, implying that the inverse problem is ill posed. In this work a new architecture and algorithm for learning multiple inverses is proposed, the polyhedral mixture of linear experts (PMLE). The PMLE keeps all the possible solutions available to the controller in real time. The PMLE is a modified mixture of experts architecture, where each expert is linear and more than a single expert may be assigned to the same input region. The learning is implemented by the hinging hyperplanes algorithm. The proposed architecture is described and its operation is illustrated for some simple cases.

1. Introduction

One of the salient characteristics of the biological motor control system is its apparent redundancy (Bernstein 1967). The controller has to act on a many to one (MTO) system and has to choose one of the many possible actions to obtain the same desired target. It was suggested that the nervous system contains an inverse model of the musculoskeletal system that is contextually being updated (see Inbar and Yafe 1976 for analysis of this idea, and Jordan 1996 for a review of recent modeling with artificial neural networks). The inverse problem is illustrated in Fig 1.

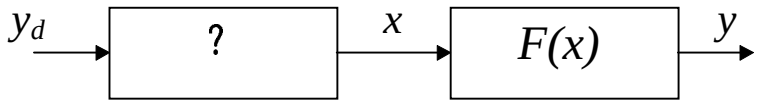

Fig 1. The inverse problem: Given a desired output y_d, find x such that $F(x)=y_d$.

This problem gets more complicated when the mapping $F(X)$ is unknown or uncertain. Then the inverse mapping should be learned from examples of input and outputs pairs (x^l,y^l). This description is appropriate for biological motor control learning, and can be also appropriate for certain robotics applications. Most of the proposed architectures and models to solve this problem choose an arbitrary solution that is the closest to the training set and to the initial conditions of the network (see for example Jordan 1996).

The Mixture of Experts (ME) architecture proposed by Jacobs et al. (1991) is a modular artificial neural network where each module is called an expert and is a parametric function of the inputs. An input dependent gate chooses the weights of each expert in the output of the mixture (see Fig 2a). The gate is also a parametric function, and all the parameters are learned from examples. In the case where each expert is a linear function and the gate chooses just one expert for a given input, the ME constructs a piecewise linear approximation of the learned mapping. We call this special architecture Polyhedral Mixture of Linear Experts (PMLE).

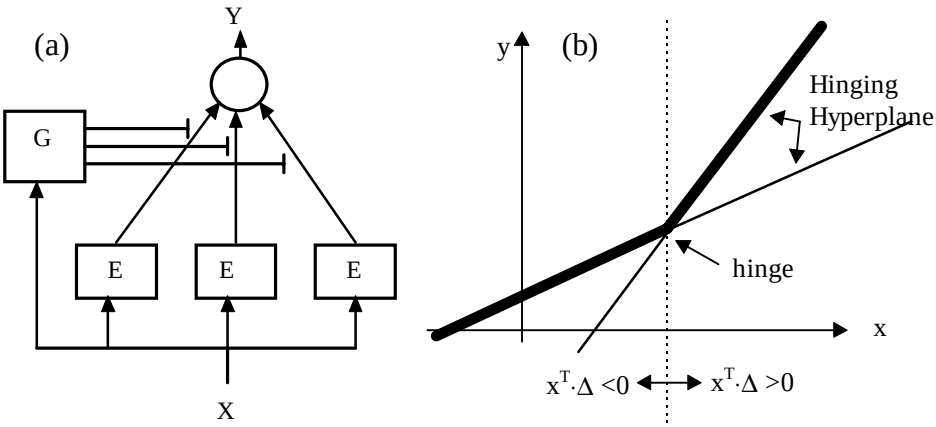

Fig 2. (a) The Mixture of Experts Architecture: Each expert (E) computes a function of the input. The gate (G) determine the weight of each expert in the output. (b) An illustration of Hinging Hyperplanes and Hinge function(Bold) in one dimension.

The Hinging Hyperplanes (HH) method proposed by Breiman (1993) is an elegant and efficient way of identifying piecewise linear models based on data collected from an unknown system. A hinge function $y=h(x)$ consists of two hyperplanes continuously joined together at a hinge. An illustration in one dimension is given in Fig 2b. In an M-dimensional space, taking $x_0=1$, that is $x=[1,x_1,...x_M]^T$, the two hyperplanes are given by $y=x^T\cdot\beta^+$ and $y=x^T\cdot\beta^-$, and are joined together on $\{x|x^T\cdot(\beta^+-\beta^-)=0\}$. The vector, $\Delta\equiv\beta^+-\beta^-$, or any multiple of Δ, is defined as the *hinge*. The explicit form of the hinge function is either $max(x^T\cdot\beta^+, x^T\cdot\beta^-)$ or $min(x^T\cdot\beta^+, x^T\cdot\beta^-)$. Given data from an unknown function $f(x)$ one can construct an approximation of this function as a sum of hinge functions. (see Breiman 1993 for the description of the algorithm)

$$\hat{f}(x)=\sum_{k=1}^{K} h_k(x) \qquad h_k(x)=\begin{cases} x^T\cdot\beta_k^+ & x^T\cdot\Delta_k \geq 0 \\ x^T\cdot\beta_k^- & x^T\cdot\Delta_k < 0 \end{cases} \qquad (1)$$

In this work the HH algorithm is used to estimate a piecewise linear model of the system, and then these estimated parameters are transformed to the PMLE parameters to enable multiple inverse model. In the next section the PMLE architecture is described, and proven to be capable of approximating any inverse function. In section 3, a simulation for a nonlinear functions is given and finally conclusions are drawn.

2. Polyhedral Mixture of Linear Experts

2. 1. The Architecture

The Mixture of Experts (ME) architecture of Jacobs et al. (1991) illustrated in Fig 2a can be formulated as follows, where f_i are the experts and g_i are the gate functions.

$$y = \sum_i g_i(x,\theta)\cdot f_i(x,w) \quad ; \quad \sum_i g_i(x,\theta) = 1 \quad ; \quad g_i(x,\theta) \geq 0 \qquad (2)$$

The Polyhedral Mixture of Linear Experts (PMLE) is a special case of the ME architecture where each expert is a linear function, and the gate function is an indicator function that separates the input space into a polyhedral partition and assigns to each polyhedron a unique linear expert, as in equation (3).

$$f_i(x,w) = x^T \cdot w \qquad g_i(x,\theta) \in \begin{cases} 1 & \text{if } x \in \text{polyhedron } i \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Where: $x=[1,x_1,...x_M]^T$, $w=[w_0,w_1,...w_M]^T$. A polyhedron is a subspace of R^n composed of the intersection of a finite number of half spaces. This architecture is actually an implementation of a piecewise linear mapping.

2. 2. The Ability to Approximate Inverse Functions

In this section it will be shown that the PMLE is able to approximate any inverse map. Let us recall the definition of the ability to approximate inverse function from Sontag (1992). The following is a property of the class of functions F_p^m, from R^p to R^m.

(INV) For any m and p, any continuous function $f:R^m \rightarrow R^p$, any compact subset $C \subseteq R^p$ included in the image of f, and any $\varepsilon>0$, there exist some $\Phi \in F_p^m$ so that $\left\| f(\Phi(x)) - x \right\| < \varepsilon$ for all $x \in C$.

Theorem:

The class of functions that are computable by the PMLE satisfies (INV), i.e., any inverse function can be approximate by the PMLE.

Proof:

Recall the following proposition and lemma from Sontag (1992)

Proposition 2.4: F_p^m, The set of maps computable by two-hidden-layer nets with processors of type *H*, satisfies (INV). (*H* is $H(x)=0$ if $x<0$ and $H(x)=1$ if $x \geq 0$)

Lemma 3.6: A function f is piecewise constant if and only if it is computable by a two-hidden-layer net with processors of type *H*.

Now all we have to add is the trivial observation that a piecewise constant function is a special case of a piecewise linear function, so that the PMLE can satisfy any piecewise constant function. Based on lemma 3.6 one can conclude that the PMLE can compute any function that is computable by a two-hidden-layer net with processors of type H. From this conclusion and Proposition 2.4 the theorem is proven.

2. 3. Parametrization via Hinging Hyperplanes

In this section the relationship between the parameters of the PMLE (see 2,3) and the HH function approximation (1) are derived, that is, given the number of hinge functions *K*, the hinges Δ_k, and the hyperplanes β_k^-, β_k^+, the parameters of the PMLE, θ and *w*, and the structure of the gate functions g_i are derived.
In order to make the description compact and readily programmable with MATLAB, the parameters are written in vector and matrix notation as follows: D is the hinges matrix where each row *k* is Δ_k, B^+ and B^- are the matrices af all β_k^-, β_k^+ respectively. These matrices are learned from examples by Breimans algorithm (1993). The gating function of the PMLE contains a vector Θ_i for each expert that describes its side for each hinge function, and each expert possesses a weight vector W_i as follows:

$$\Theta_i = [\theta_{i1} \quad \cdots \quad \theta_{iK}]$$

$\theta_{ik} = +1$ if expert *i* belongs to $x^T \cdot \Delta_k \geq 0$

$\theta_{ik} = -1$ if expert *i* belongs to $x^T \cdot \Delta_k < 0$

$$W_i = \begin{bmatrix} w_i(1) \\ \vdots \\ w_i(\mathrm{M}+1) \end{bmatrix}$$

For a given input x, the gate can decide which expert describes the function at that point.

$$g_i(x) = \delta_K\left(\Theta_i \cdot sign\left(D^T \cdot X\right)\right) \quad \text{where } \delta_K(u) = \begin{cases} 1 & u = K \\ 0 & \text{Otherwise} \end{cases}$$

The following algorithm transforms the parameters of the Hinging-Hyperplanes function approximation to the parameters of the PMLE. The algorithm uses linear programming (LP) to find the position of each hinge in relation to each expert. LP is chosen for its efficient algorithmic implementability. See Karniel et al. (1997) for a more detailed description of this algorithm.

Initiation: For the first hinge, $D=\Delta$, $B^+=\beta^+$, $B^-=\beta^-$.
Construct two experts as follows: $\Theta_1 = [+1]$ $\Theta_2 = [-1]$ $W_1 = [\beta^+]$ $W_2 = [\beta^-]$

For each new hinge, *k*:
  For each expert, *i*:
    Check the position of the hinge according to expert *i* by LP:
    Calculate *Mn* and *Mx* which are the maximum and minimup of $x^T \cdot \Delta_k$
    If *Mn*>0 and *Mx*>0 the hinge is in one side: $W_i = W_i + \beta^+$, $\Theta_{i,k} = +1$
    If *Mn*<0 and *Mx*<0 the hinge is in the other side: $W_i = W_i + \beta^-$, $\Theta_{i,k} = -1$
    else: the hinge goes through this expert, split to get two experts:
    $W_{N+1} = W_i + \beta^+$ $\Theta_{N+1,k} = +1$ $W_i = W_i + \beta^-$ $\Theta_{i,k} = -1$
  end (for expert *i*)
  Add the columns Δ_k, β^+ and β^- to the matrixes *D*, B^+,B^- respectively.
end (for hinge *k*)

2. 4. Constructing the Complete Inverse Approximation

One can use the PMLE in order to construct the complete inverse approximation. For the one-dimensional problem, one can invert each expert and get a candidate-solution which should be validated for being in that experts' range of operation. This can be done with the inverse PMLE and the regularization problem is now reduced to a problem of choosing one of the possible solutions. We can give each solution an identification number, call it *p*, and add this parameter as a regularization input. At this point the solution to the problem in Fig 1 can be illustrated as in Fig 3.

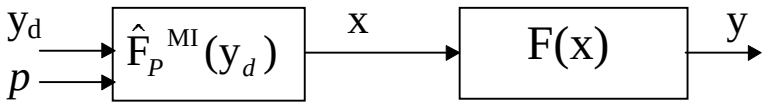

Fig 3. The proposed solution to the inverse problem. The multiple inverse (MI) approximation can be implemented by the IPMLE.

Further investigation is needed in order to describe the values of the parameter *p* and in choosing the appropriate solution, but this stage depends on the specific control problem, its constraints and goals.

3. Simulations

The first example demonstrates the construction of the complete inverse of a smooth function which is not injective. We have drawn 400 examples from the function $y=sin(x^2)$ Where x_i was uniformly distributed in the range [-2,2]. The results of the HH algorithm are given in Fig 4a.

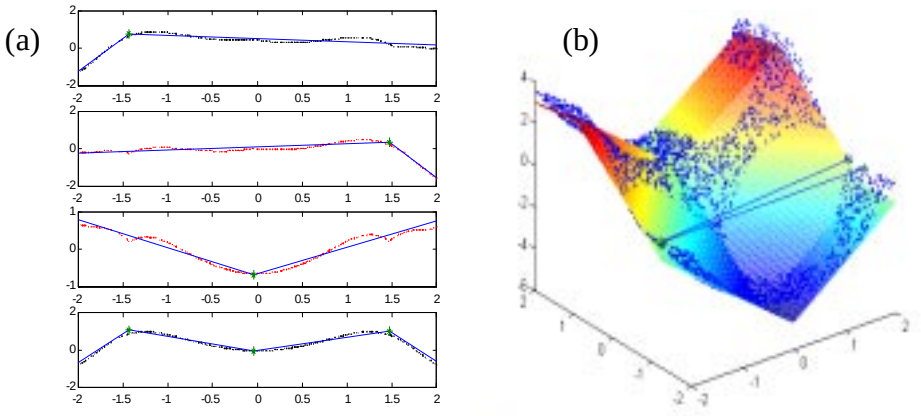

Fig 4. Simulation of the HFA. (a) Re-fitting the three hinges from to the examples from the function $y=\sin(x^2)$. The dots are the examples and the lines are the hinge functions. (b) The hinge functions over the examples of the 2d function.

Let us demonstrate the results of the IPMLE: The complete inverse of the target value 0.5 is the following 4 answers:

$$\text{» [X] = ipmle(D,W,teta,0.5)} \Rightarrow \text{X} = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ -1.6102 & -0.7285 & 1.6185 & 0.7485 \end{bmatrix}$$

And the complete inverse of the target value -0.5 is the following 2 answers:

$$\text{» [X] = ipmle(D,W,teta,-0.5)} \Rightarrow \text{X} = \begin{bmatrix} 1.0000 & 1.0000 \\ -1.9394 & 1.9478 \end{bmatrix}$$

In the second example we examine a two dimensional function. We have chosen to check the algorithm on the following function which can be presented as a typical control problem. We have drawn 4000 examples from the following function $y = 4 \cdot x_2 / \left(1 + x_2^2\right) + x_1^2 - 2$ Where x_1 and x_2 were uniformly distributed in the range [-2,2]. This function was taken from Lee and Lee (1995) who used it for a similar goal with another architecture. The results are presented in Fig 4b.

4. Conclusions

A new architecture for learning the inverse of a redundant system was proposed. The polyhedral mixture of linear experts (PMLE) can learn from examples a piecewise linear approximation of the system and can then be easily inverted. The structure of the architecture was presented, its ability to approximate any inverse function was proven, and an algorithm to learn its parameters from examples was demonstrated. The problem of choosing the proper solution from all the possible solutions will be the focus of future research.

References

[1] Bernstein, N. (1967) The Coordination and Regulation of Movements. Pergamon Press, Oxford.

[2] Breiman Leo. (1993) "Hinging Hyperplanes for Regression, Classification, and Function Approximation", IEEE Trans. on Information Theory 39:999-1013.

[3] Inbar GF, Yafe A (1976) Parameter and Signal Adaptation in the Stretch Reflex Loop. In: Homma S. (Ed), Progress in brain research. Vol 44:317-337. Elsevier.

[4] Jacobs R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E. (1991). "Adaptive mixture of local experts" Neural Computation. 3:79-87.

[5] Jordan MI (1996) Computational Aspects of Motor Control and Motor Learning. In Heuer H, Keele SW (eds.) Handbook of Perception and Action Vol 2.

[6] Karniel A, Meir R, Inbar GF (1997) Polyhedral Mixture of Linear Experts for Many-To-One Mapping Inversion. Department of Electrical Engineering EEPUB 1126, Technion Israel.

[7] Lee S, Lee JM. (1995) Nonlinear system control based on multi-resolution radial-basis competitive and cooperative networks. Neurocomputing 9:187-206.

[8] Sontag Eduardo D. (1992) "Feedback Stabilization Using Two-Hidden-Layer Nets", IEEE Transactions on neural networks, Vol. 3, No. 6.