Adaptive mixture of local experts

Robert A. Jacobes
Michael I. Jordan
Steven J. Nowlan
Geoffrey E. Hinton

By:
Negin Yousefpour

April 2010

Contents

- Introduction
- Local Experts
- Motivation
- Architecture
- ME Learning
- Vowel recognition
- Summary
- References

Introduction

- **Modular Neural Networks** proposed by Jacobs et al. (1991), consist of a group of networks ➡ **Local Experts**
- Experts competes among themselves to learn the different characteristics of input patterns
- Competition is regulated by another network ➡ **Gate Network**
- Gate network assigns different space areas of the inputs to the different local experts

Local Experts

- They are feed-forward Neural Networks.
- The experts are local, means:
  - The weights in one expert are decoupled from the weights in other expert
  - Each expert will be allocated to only a small local region of the space of possible input vectors
- A gating network could decides which of the experts should be used for each training case.

Motivation

a set of training cases naturally divided into subsets that correspond to distinct subtasks

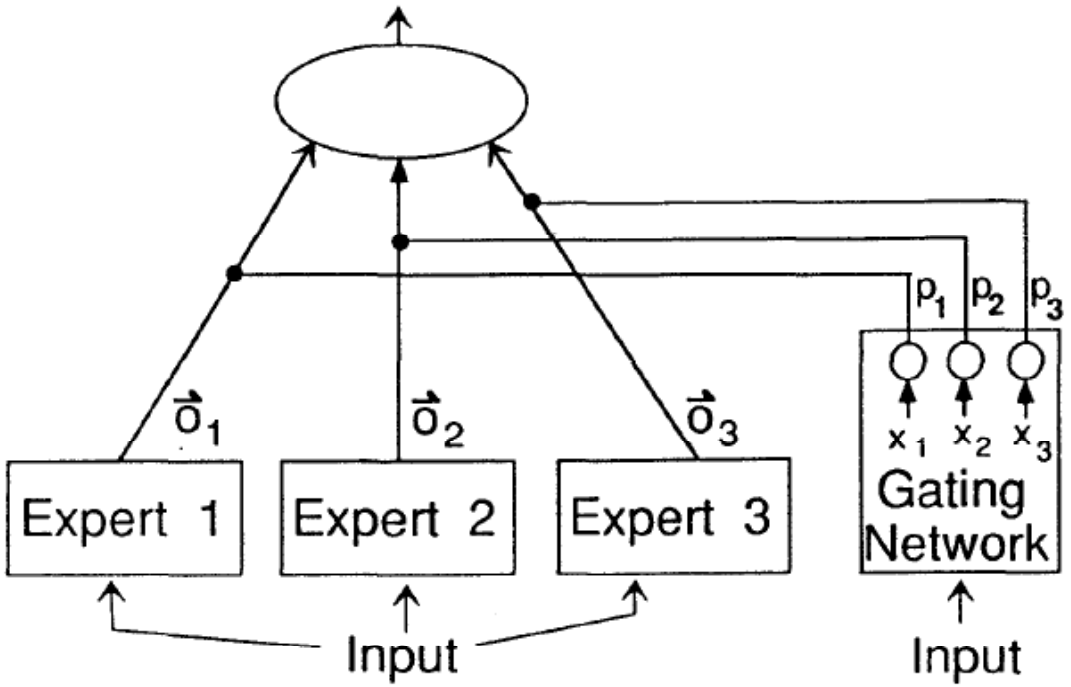
using a system composed of several different "expert" networks plus a gating network

less interference to perform different subtasks comparing to BP Networks

faster learning and Better generalization

Architecture

$$o = \sum_{i=1}^{L} p_i . o_i$$



$$p_i = \frac{e^{S_i}}{\sum_{i=1}^{L} e^{S_i}}$$

Output number =L

Number of local experts=L

Learning

Hampshire and Waibel (1989)

$$E^c = \|\mathbf{d}^c - \sum_i p_i^c \mathbf{o}_i^c\|^2$$

- When the weights in one expert change, the residual error changes, and so the error derivatives for all the other local experts change.

Jacob et.al. (1991)

$$E^c = \langle \|\mathbf{d}^c - \mathbf{o}_i^c\|^2 \rangle = \sum_i p_i^c \|\mathbf{d}^c - \mathbf{o}_i^c\|^2$$

$$E^c = -\log \sum_i p_i^c e^{-\frac{1}{2}\|\mathbf{d}^c - \mathbf{o}_i^c\|^2}$$

Learning

- A gradient decent method is used to find minimum for:

$$E^C = -log\left(\sum_i p_i{}^C e^{-\frac{\left\|d^C - o_i{}^C\right\|^2}{2\sigma^2}}\right)$$

E^c: error on training case c
p_i : output of the gating network for expert i
d^c: the desired output vector
o_i: the output vector of expert i
σ: variance is constant

- This is the negative log probability of generating the desired output vector under a mixture of Gaussians model of the probability distribution
- Each expert is required to produce the whole of the output vector rather than a residual.
- System tends to devote a single expert to each training case.

Learning

Gradient decent has two effect

raises the mixing proportion of experts that do better than average

makes each expert better for those cases for which it has a high mixing proportion

mixing proportion near **1** to one expert on each case

each expert can focus on modeling the cases it is good at

Associative Competitive Learning

The data vectors used in competitive learning	→	The output vectors of an associative network
The competitive network	→	Inputless stochastic generator of output vectors
The competitive learning	→	Generate output vectors with a distribution that matches the distribution of the "data" vectors.
The weight vector of each competitive hidden unit(Local expert)	→	The mean of a multidimensional Gaussian distribution

Associative Competitive Learning

- The log probability of generating any particular output vector:

$$\log P^C = -log(\sum_i p_i{}^C k e^{-\frac{1}{2}\left\|\mu_i{}^C - o^C\right\|^2})$$

$\mathbf{p_i}$: The probability of picking hidden unit i

k : A normalizing constant

$\boldsymbol{\mu_i}$: "weight" vector of the hidden unit

Soft / Hard competitive learning

- "Soft" competitive learning
  - Modifies weights in all hidden units
  - objective is to increase the product of the probabilities of generating the output vectors in the training set
  - Like GTM
- "Hard" competitive learning
  - It is assumed that only the out put vector must be generated by the hidden unit with the closest weight vector to input vector.
  - only this weight vector needs to be modified to increase the probability of generating the data vector.
  - Like ME

Vowel recognition

- The mixture of experts model was evaluated on a speaker independent, four-class, vowel discrimination problem .
- The data is came from the work by two linguistics experts Peterson and Barney(1952)
- They conducted experiments where listeners were asked to identify vowels
- The general purpose of these tests was to obtain an aural classification of each vowel
- Different researches tried to do the same thing by ANN

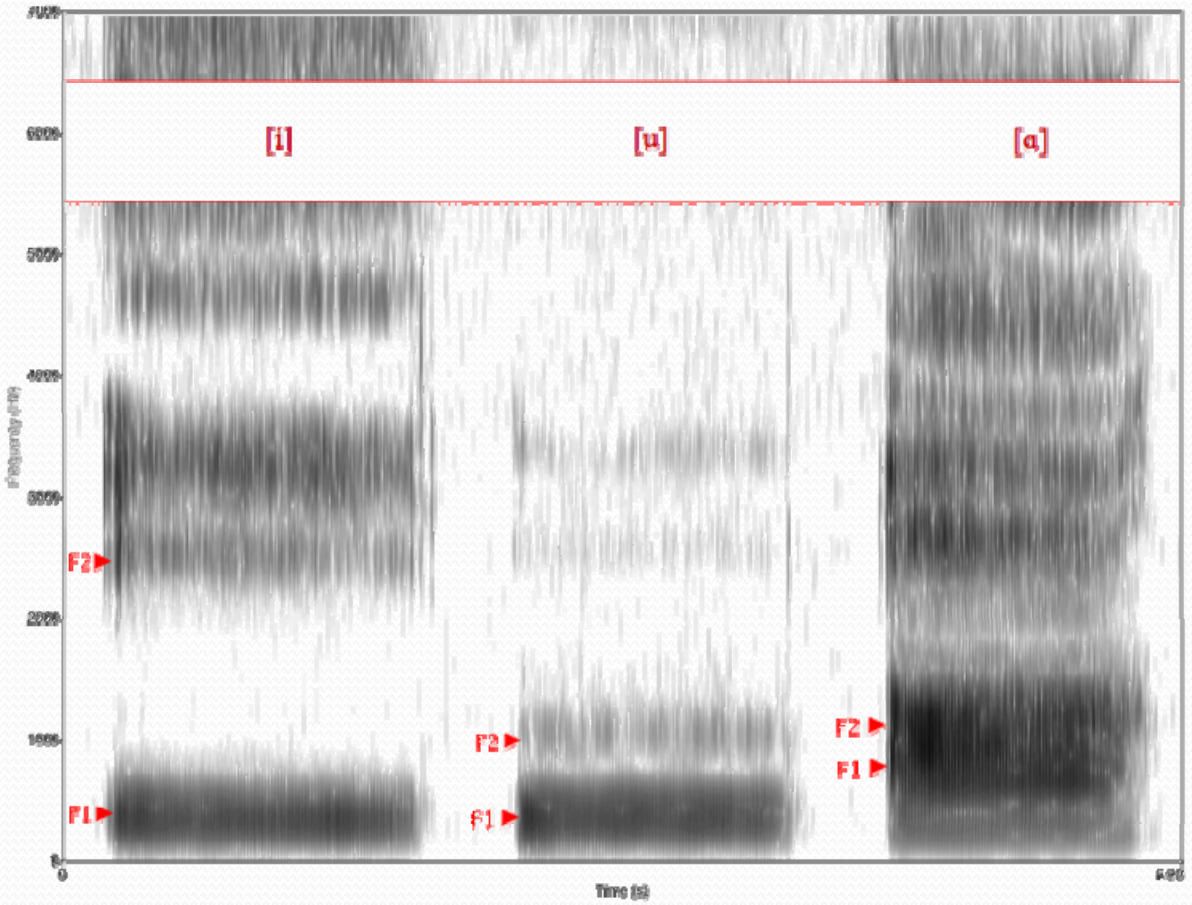
Vowel recognition

- The vowels were: **heed, hid, head, had, hud, hod, hawed, hood, who'd, heard**.
- Generally, Vowels can be organized according to the tongue's position
- The spectral peaks of the sound spectrum are called Formants
- The first formant F1 can be related to how far the tongue is raised and F2 to which part of the tongue is raised

Vowels as pronounced by **Bruce Hayes (Department of Linguistics, UCLA)**

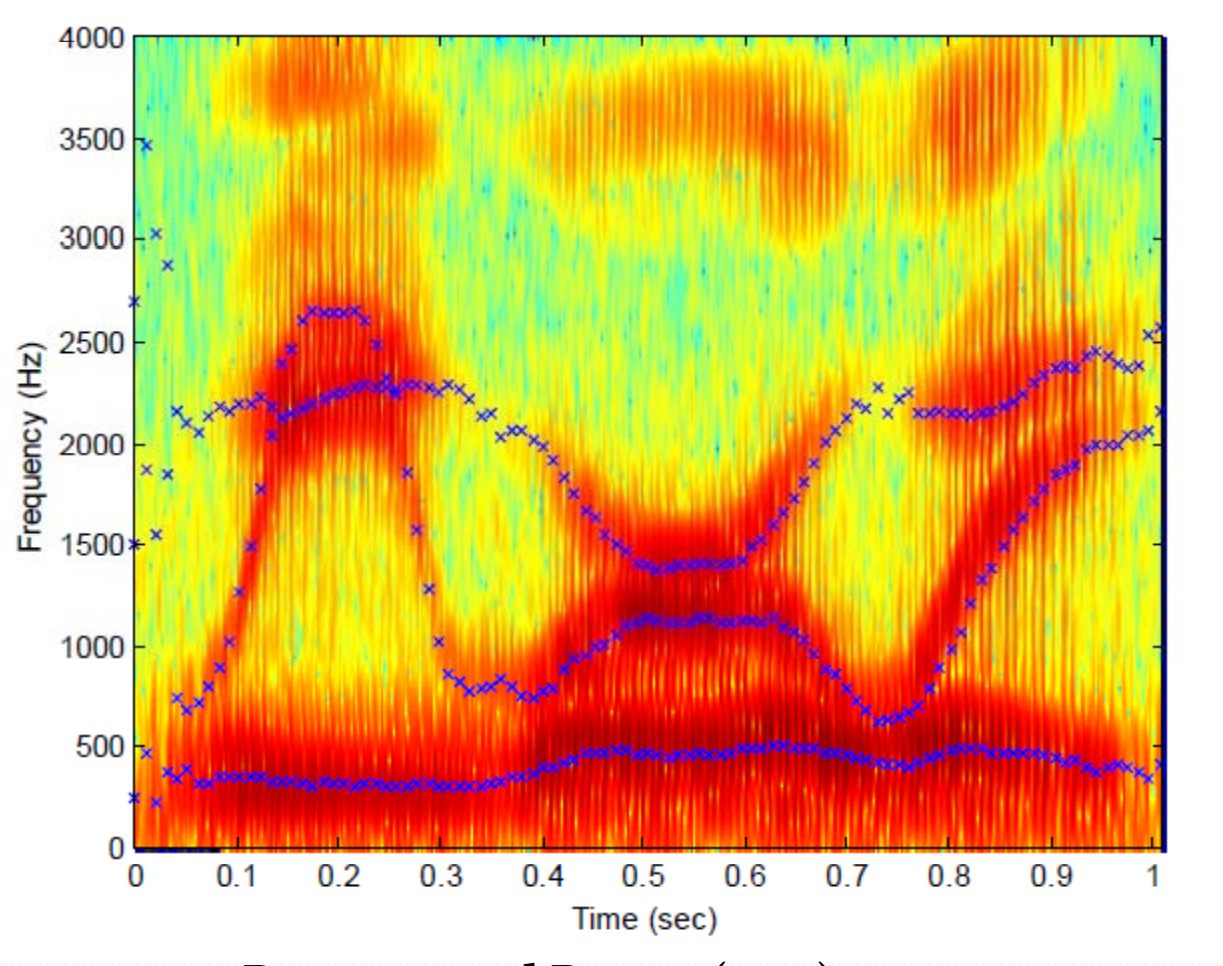
Word used in [1]	IPA symbol for the vowel
heed	i
hid	ɪ
head	ɛ
had	æ
hod	ɑ
hawed	ɔ
hood	ʊ
who'd	u
hud	ʌ
heard	ɜ

Vowel recognition

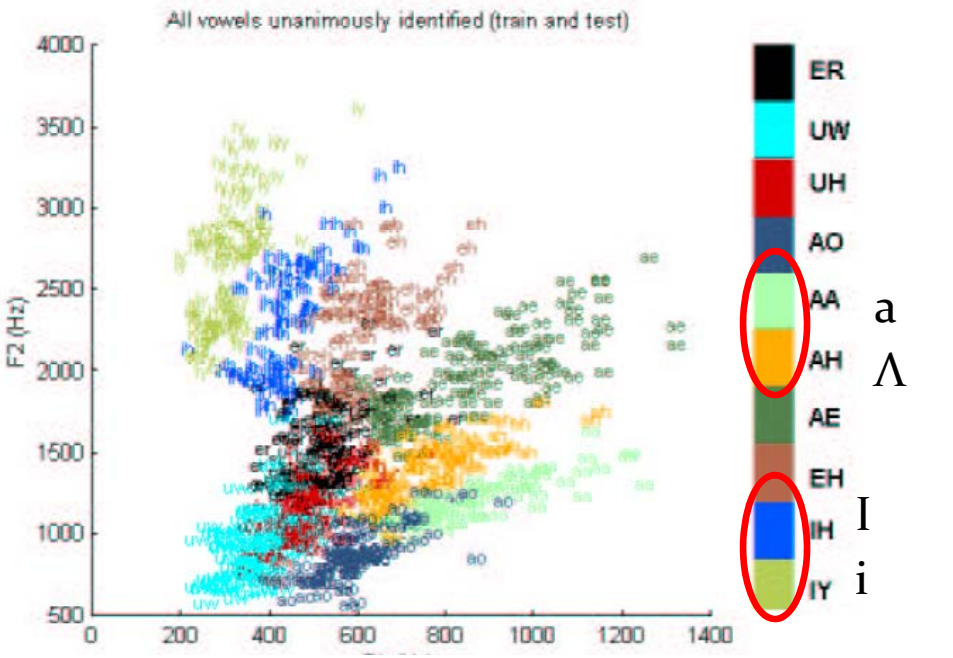


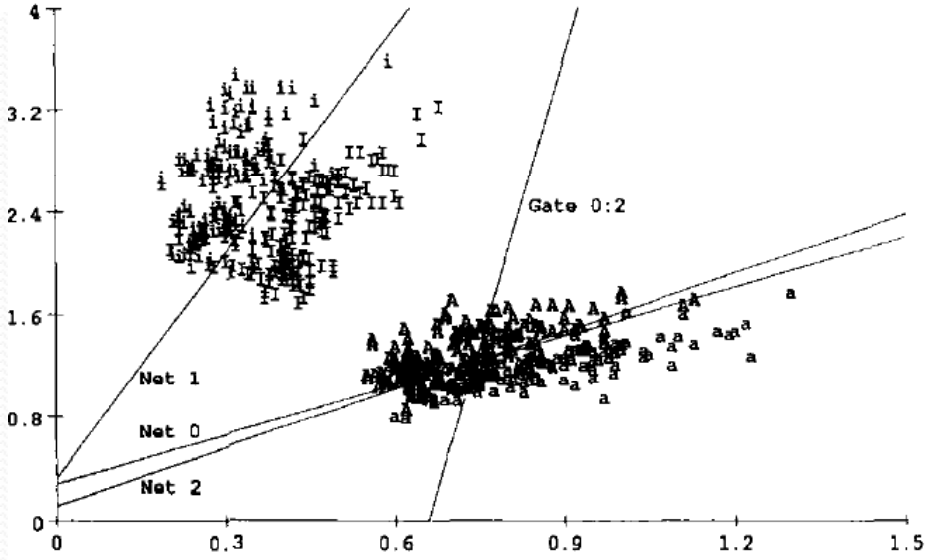
Wiki Pedia

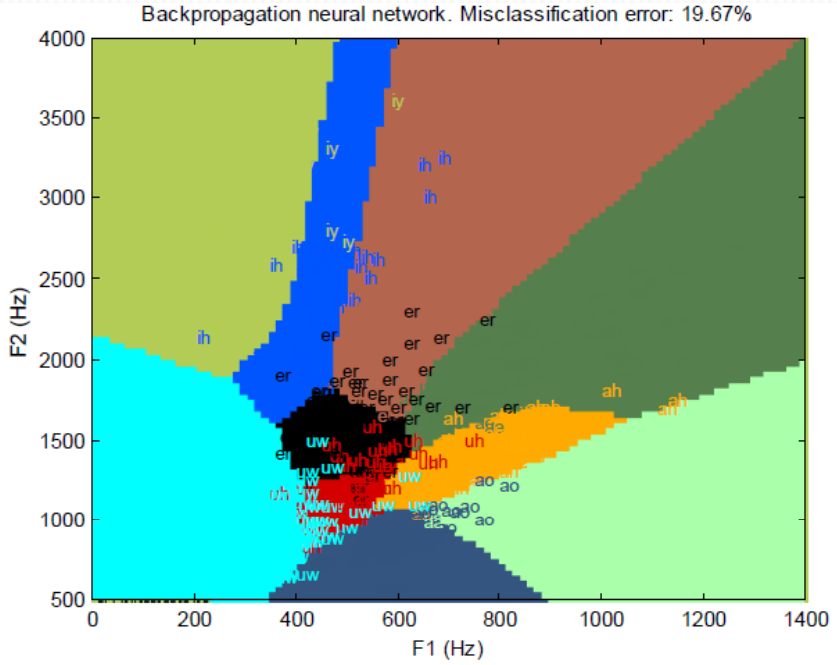
Vowel recognition



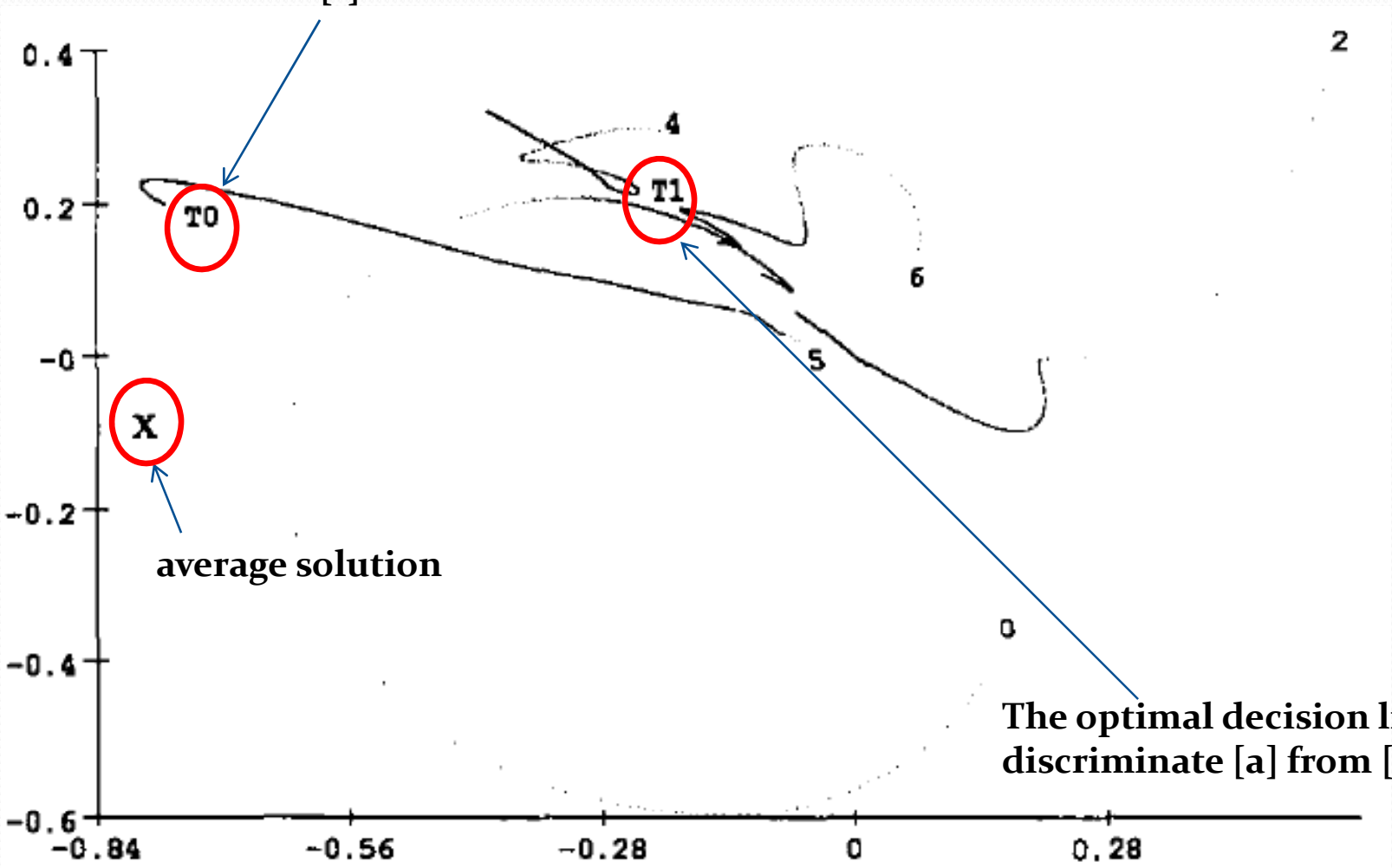
Peterson and Barney(1952)







A. Klautau, 2009



The trajectories of the decision lines of some experts

Comparing ME with BP network

System	Train % correct	Test % correct	Average number of epochs	SD
4 Experts	88	90	1124	23
8 Experts	88	90	1083	12
BP 6 Hid	88	90	2209	83
BP 12 Hid	88	90	2435	124

- No. of parameters for the BP = No. of parameters for mixture models
- The ME reach the error criterion significantly faster than the BP networks
- Learning time for the ME decrease when the number of experts increase, but in BP it is vice versa

Summary

- The mixture of experts outperform single back-propagation networks
- ME show much better generalization properties when dealing with relatively small training sets
- 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 .

Reference

- http://www.bcs.rochester.edu/people/robbie/jacobslab/cheat_sheets.html
- Klautau, A. Classification of Peterson & Barney's vowels using Weka, 2002
- Matera, F. Modular Neural Networks, Semeion Research Center, 1998
- Nowlan, S. J. Hinton, G. E. Evaluation of Adaptive Mixtures of Competing Experts, Neural Computation, 1990
- Peterson, G. E. Barney, H. L. Control methods used in a study of the vowels, 1952