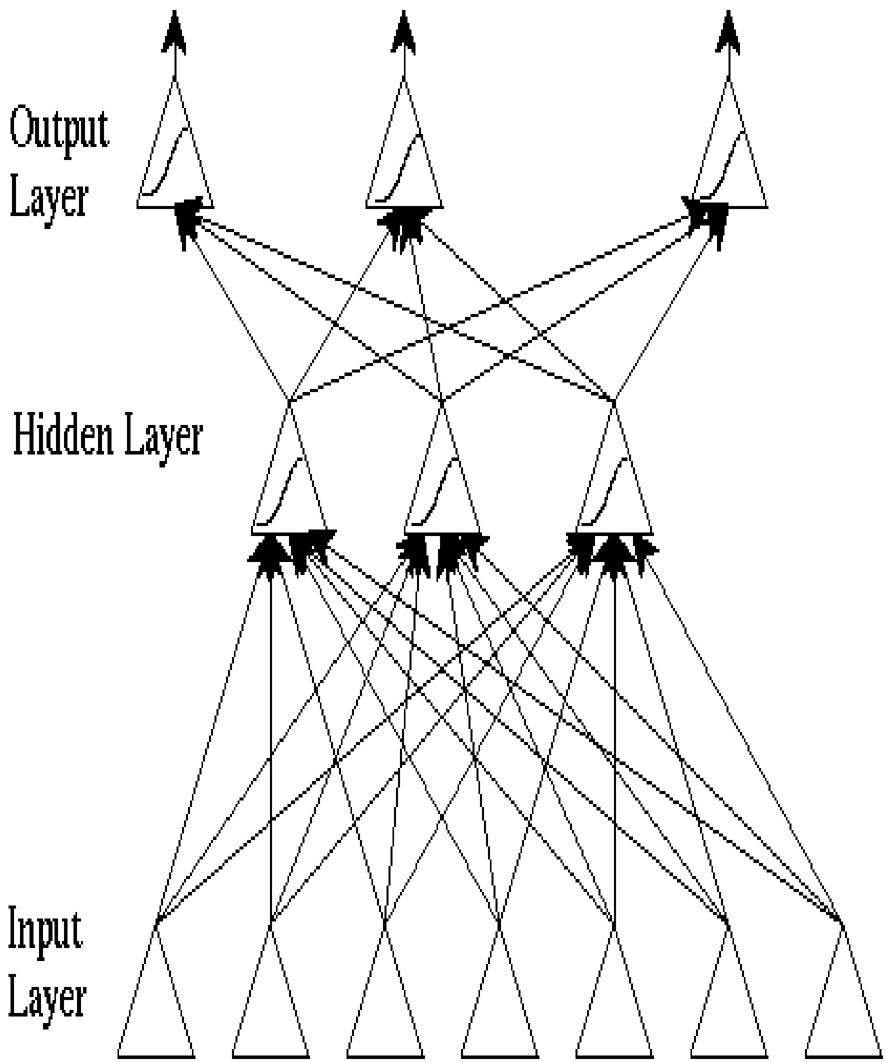
cs / philo 372
week 12

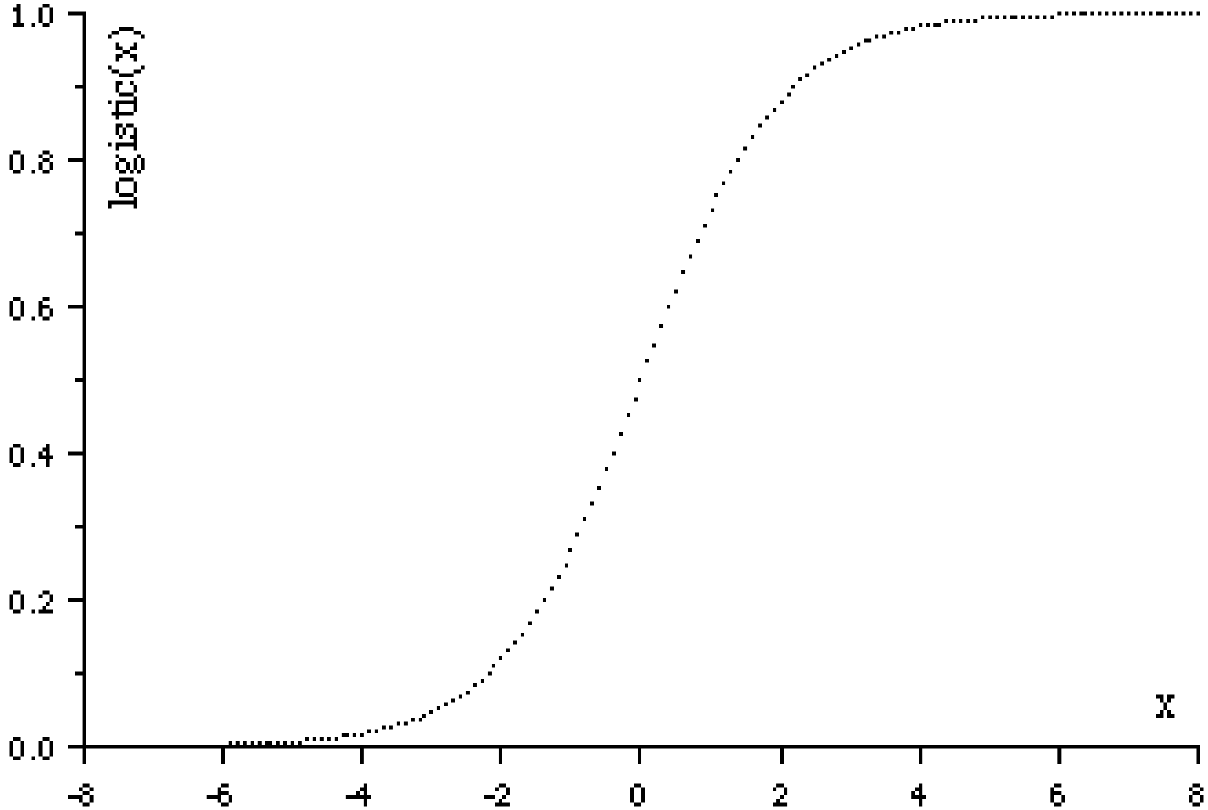
Neural Networks
Support Vector Machines

Neural Networks



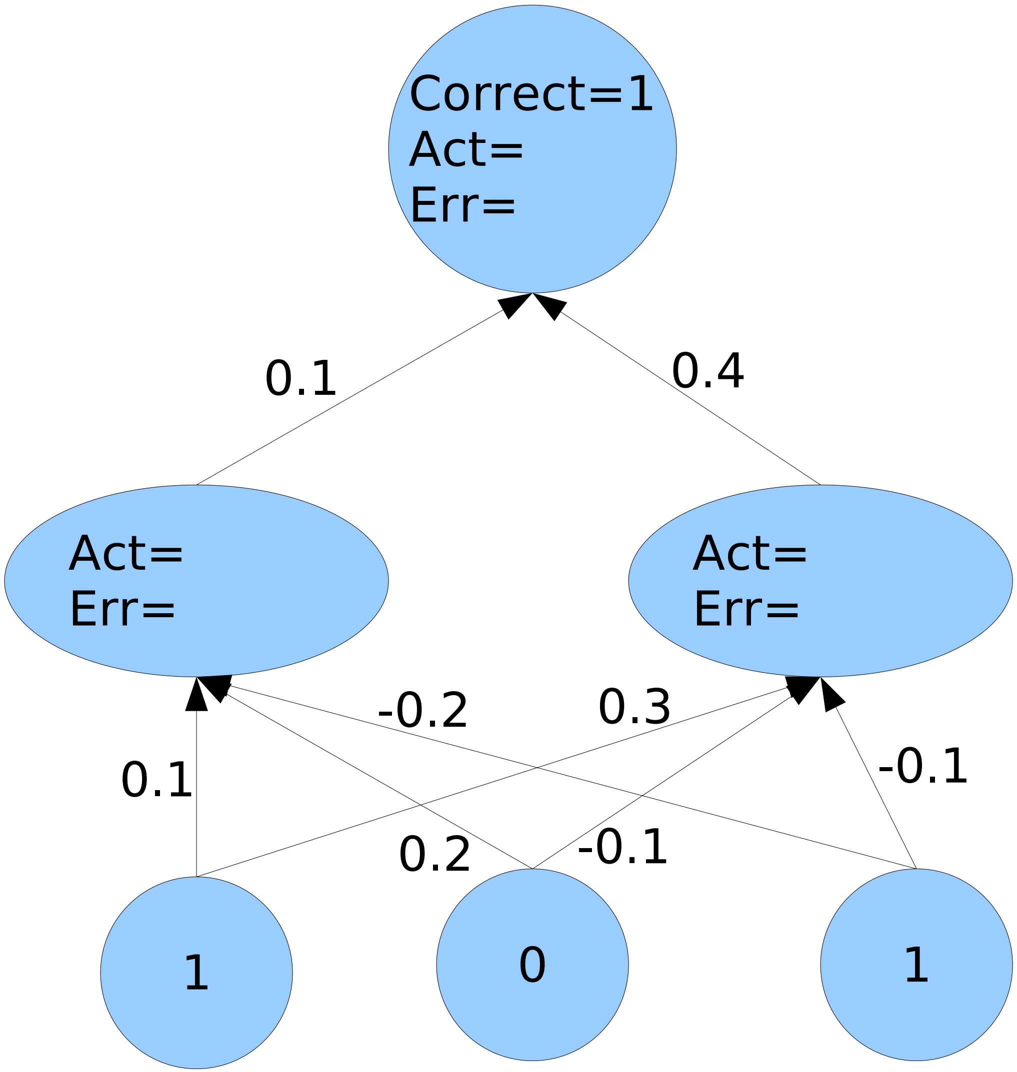
- Activation function:
  - Xj = 1 / (1+exp(-sum(Wij*Xi)))
- Derivative thereof:
  - Assuming E=sum(Yi-Xi)^2
  - For output units
    - Ei= (Yi-Xi)*[Xi*(1-Xi)]
  - For hidden units
    - Ei=sum(Wij*Ej)*[Xi*(1-Xi)]
- Weight change on link from unit i to unit j
  - Dij = Xi*Ej

Neural Networks – where do they learn best?



- Ei= (Y i-X i)* [X i* (1-X i)]

NN Example in practice



Summed input	Activation	Derivative
-2.25	0.1	0.09
-2	0.12	0.1
-1.75	0.15	0.13
-1.5	0.18	0.15
-1.25	0.22	0.17
-1	0.27	0.2
-0.9	0.29	0.21
-0.8	0.31	0.21
-0.7	0.33	0.22
-0.6	0.35	0.23
-0.5	0.38	0.24
-0.4	0.4	0.24
-0.3	0.43	0.24
-0.2	0.45	0.25
-0.1	0.48	0.25
0	0.5	0.25
0.1	0.52	0.25
0.2	0.55	0.25
0.3	0.57	0.24
0.4	0.6	0.24
0.5	0.62	0.24
0.6	0.65	0.23
0.7	0.67	0.22
0.8	0.69	0.21
0.9	0.71	0.21
1	0.73	0.73
1.25	0.78	0.78
1.5	0.82	0.82
1.75	0.85	0.85
2	0.88	0.88

Neural Networks Work Well

but they are rather slow

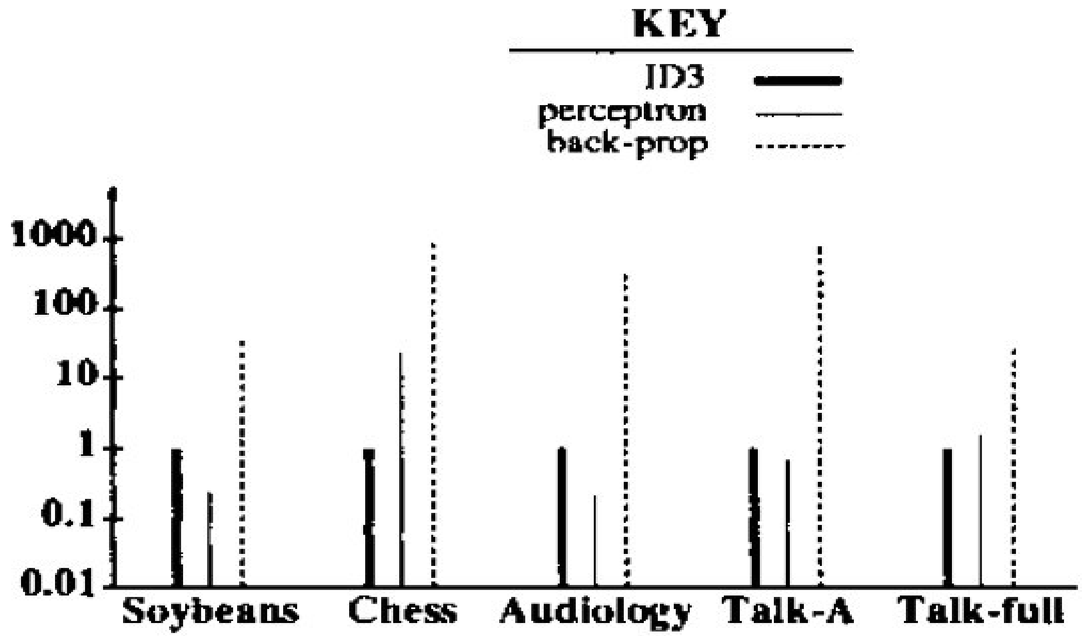
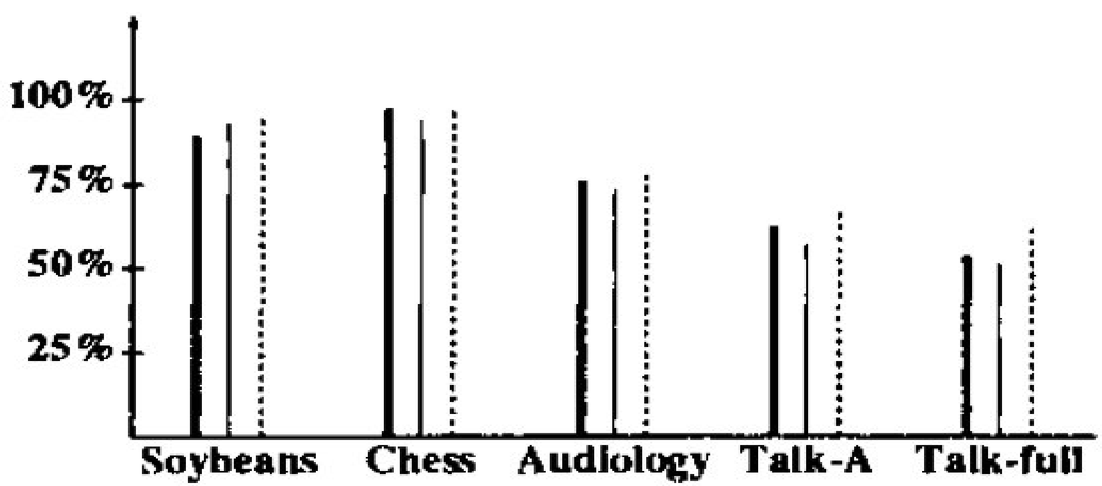


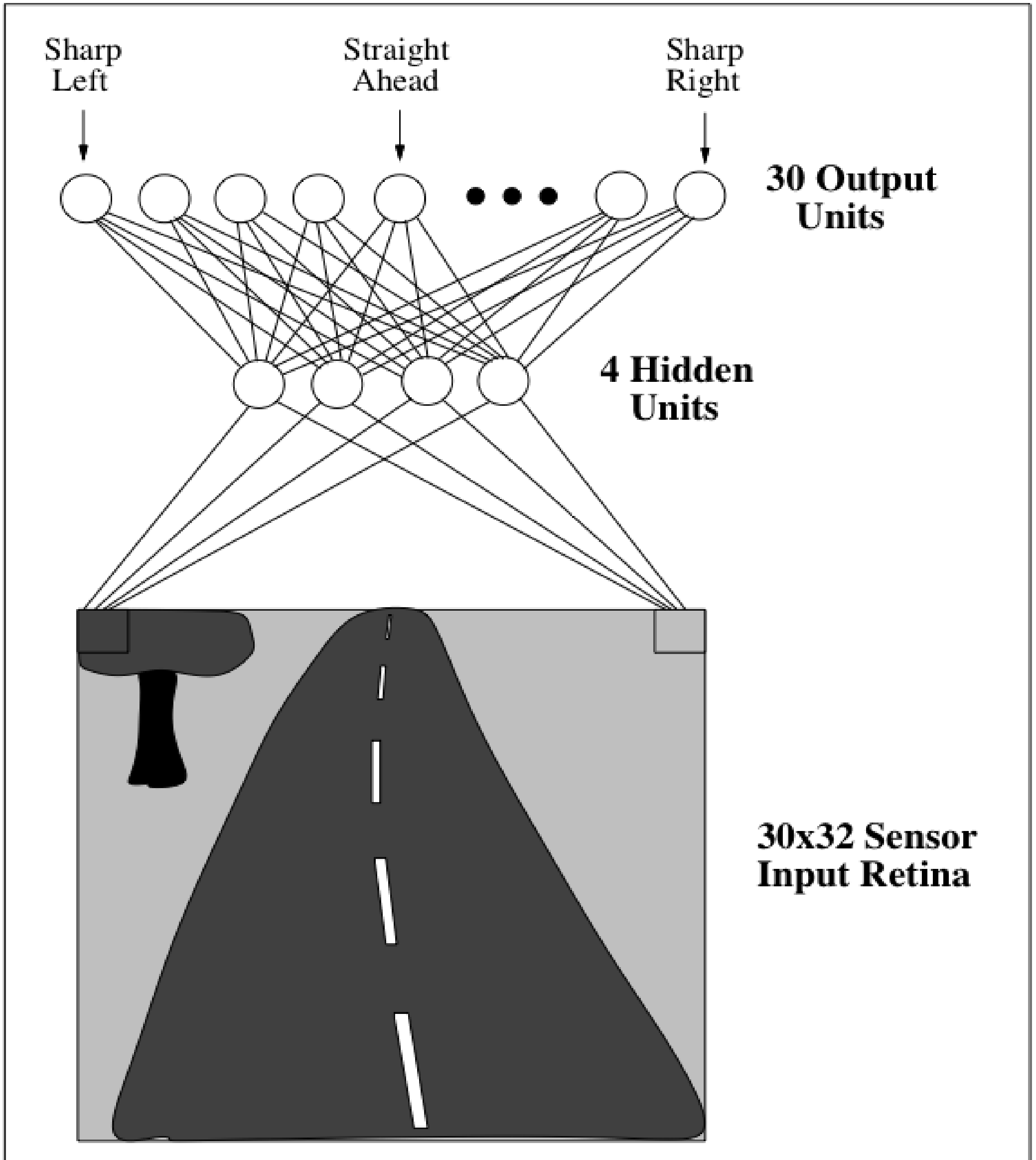
Figure 1. Relative Training Times of the Three Algorithms (scaled to ID3)



- Do a couple more example applications
- Include Hanson & Petsche Autoassociators
- Include Pomerleau's driving
- Then goto Caruana's multicategory analysis as an example of why they work well?
- Finally wrap up with SVM.

Neural Networks for driving cars

from Pomerleau 1991-1996



- 4 Hidden Units!!!
- Hands almost free across america
- Precursor of DARPA challenges
- Able to handle day, night, rain, ...
  - How?
- Q: why 30 outputs
  - 1 is sufficient?
- Use of simple recursion (not shown)

Anomaly Detection

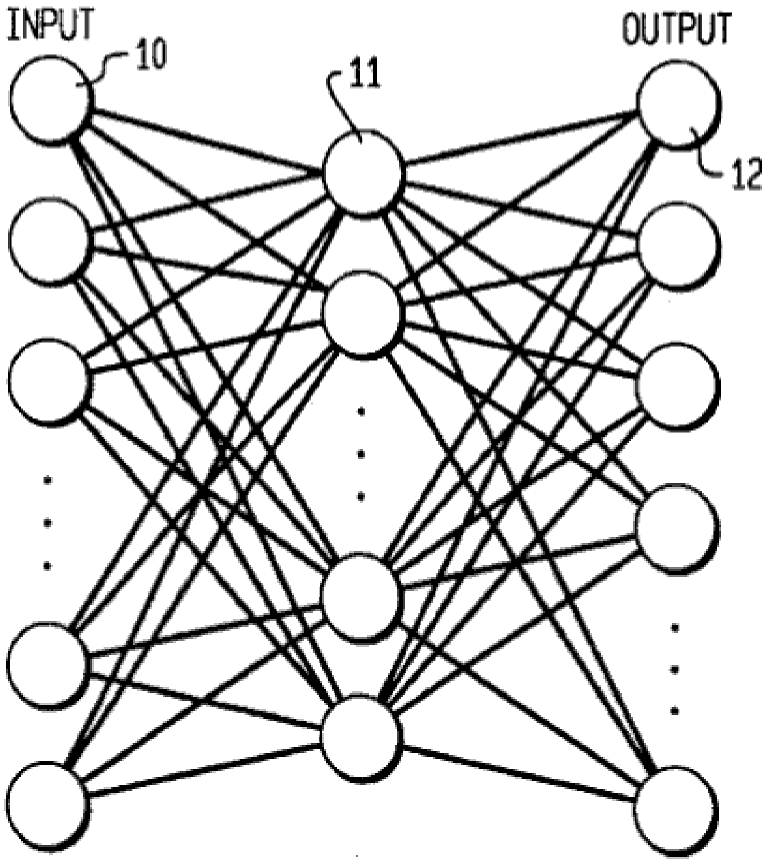
from Hanson & Petsche (1996-2000)

- Problem: determine when you do not know
  - For instance, if Pomerleau always trained during daytime, what happens at night
    - Bad case – network causes car to confidently drive off the road.
- Hanson & Petsche's problem – when is a motor about to fail?
  - Only ever see motor when it is behaving normally
  - Related to learning from positive only examples
  - H&P changed the question to “when am I seeing something I have never seen before?”

Anomaly Detection

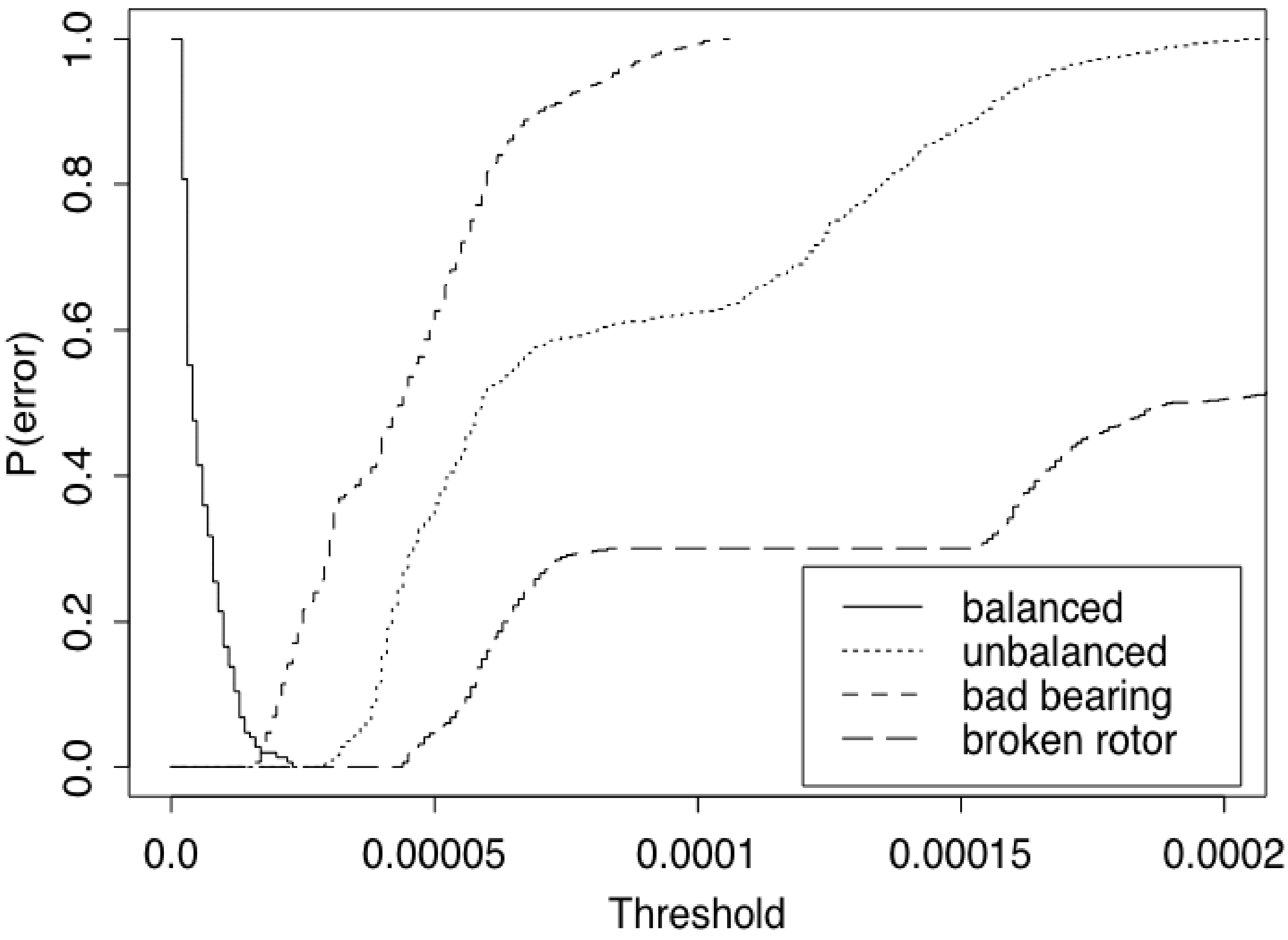
- Navy
  - Motors are most likely to fail immediately after repair, or after they have been running for a long time
  - So, should only repair when they are about to fail
  - When is that?

Autoassociators for AD



- Autoassociator is a NN that simply reproduces its input in its output
- Key observation:
  - Reproduction success indicates that similar input has been seen before
  - Failure indicates that input has not been seen
  - So, if cannot reproduce then motor is doing something new -- failing?

AD with Autoassociators

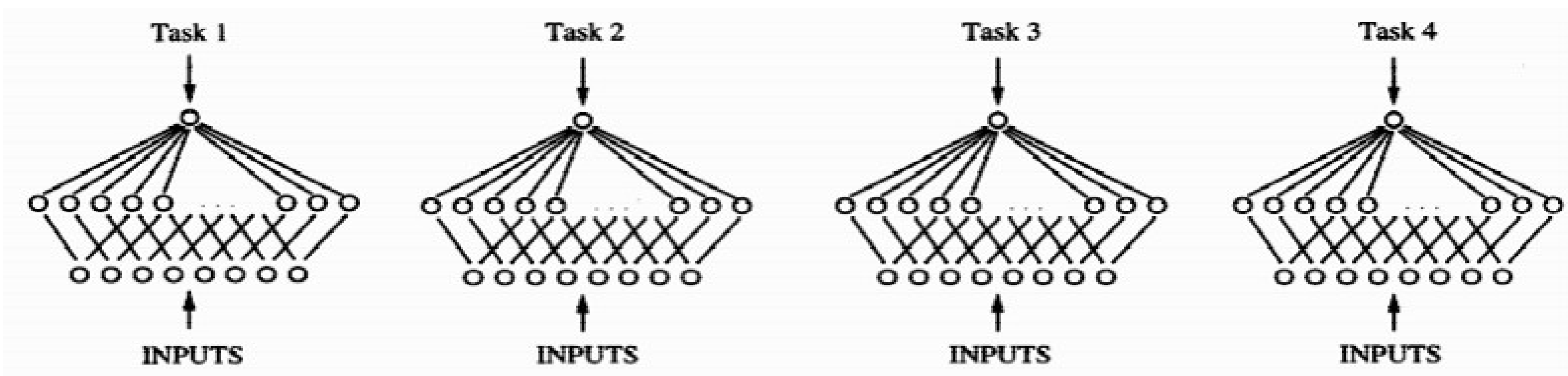


MultiCategory Learning

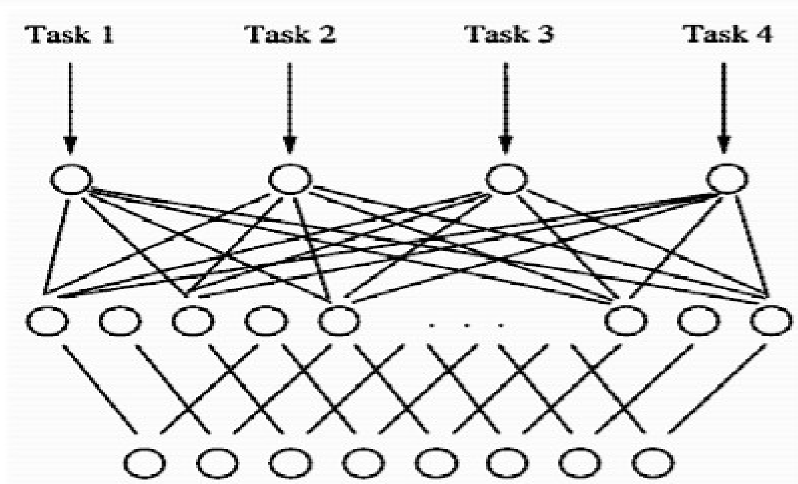
from Caruana (1997)

- Question, which of these works best?
  - Sum squared error across all 4 tasks
- Why?

System 1



System 2



MTL really works

	ROOT-MEAN SQUARED ERROR ON TEST SET				
TASK	**Single Task Backprop (STL)**			**MTL**	Change MTL
	6HU	24HU	96HU	120HU	to Best STL
Doorknob Loc	.085	.082	**.081**	**.062**	-23.5% *
Door Type	.129	**.086**	.096	**.059**	-31.4% *

- Tasks must be related
- Conceptually similar to autoassociator
  - Forcing system to compress data into hidden units. Extra tasks constrain hidden unit representation thereby making it more likely to get a good representation
- MTL does not require NN, but it is easy using NNs

NN Conclusions

- Wide range of problems and solution approaches
- Nns are not a single thing, but a suite of solution approaches
- Even within a single approach there are lots of variables
  - NN structure, etc.
- “NNs are often the second best approach to solving a problem” (Dietterich, pc, 1998)
  - So can spend months/years looking for best, or just use a NN

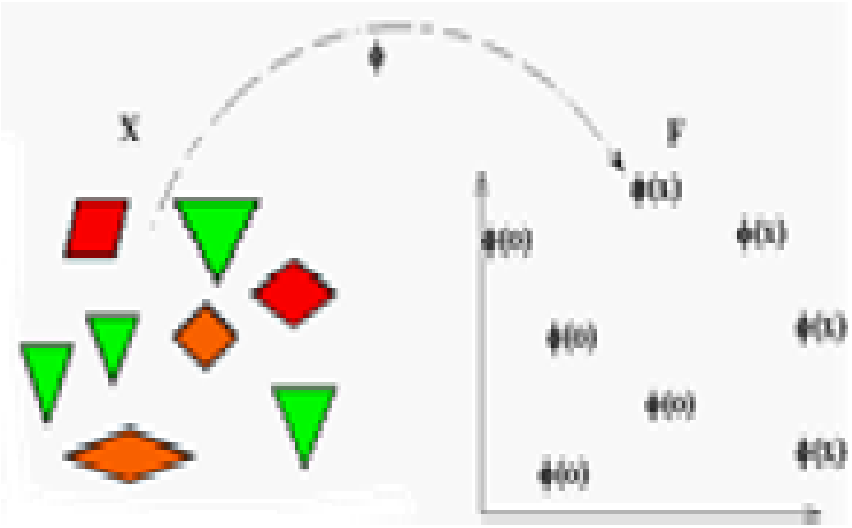
Support Vector Machines

- Perhaps the most interesting development in AI in past 15 years
- Largely attributed to Vapnik
  - Vapnik also shows up in AI / COLT in the Vapnik-Chervonenkis dimension
- Relates ML explicitly to compression, especially to wavelet compression

SVM – Main Idea

- Kernel Methods work by:

  1-embedding data
  in a vector space
  2-looking for
  (linear) relations in such
  space
- If map chosen suitably, complex relations can be simplified, and easily detected



SVM -- Observations

- 1- Much of the geometry of the data in the embedding space (relative positions) is contained in all pairwise inner products*

  We can work in that space by specifying an inner product function between points in it (rather than their coordinates)

- 2- In many cases, inner product in the embedding space very cheap to compute

<x1,x1>	.	<x1,x2>	<x1,xn>
<x2,x1>	.	<x2,x2>	<x2,xn>
.	.	.	.
<xn,x1>	.	<xn,x2>	<xn,xn>

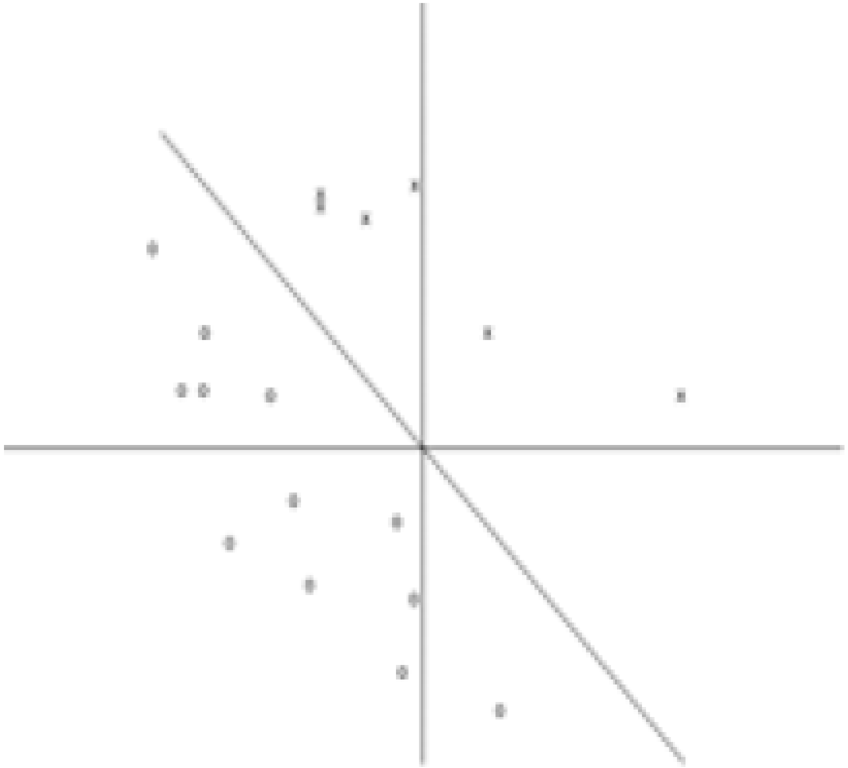
* Inner products matrix

Linear SVM

- Data $\{x_i\}$ in vector space X, divided into 2 classes $\{-1,+1\}$
- Find linear separation: a hyperplane

$$\langle w, x \rangle = 0$$

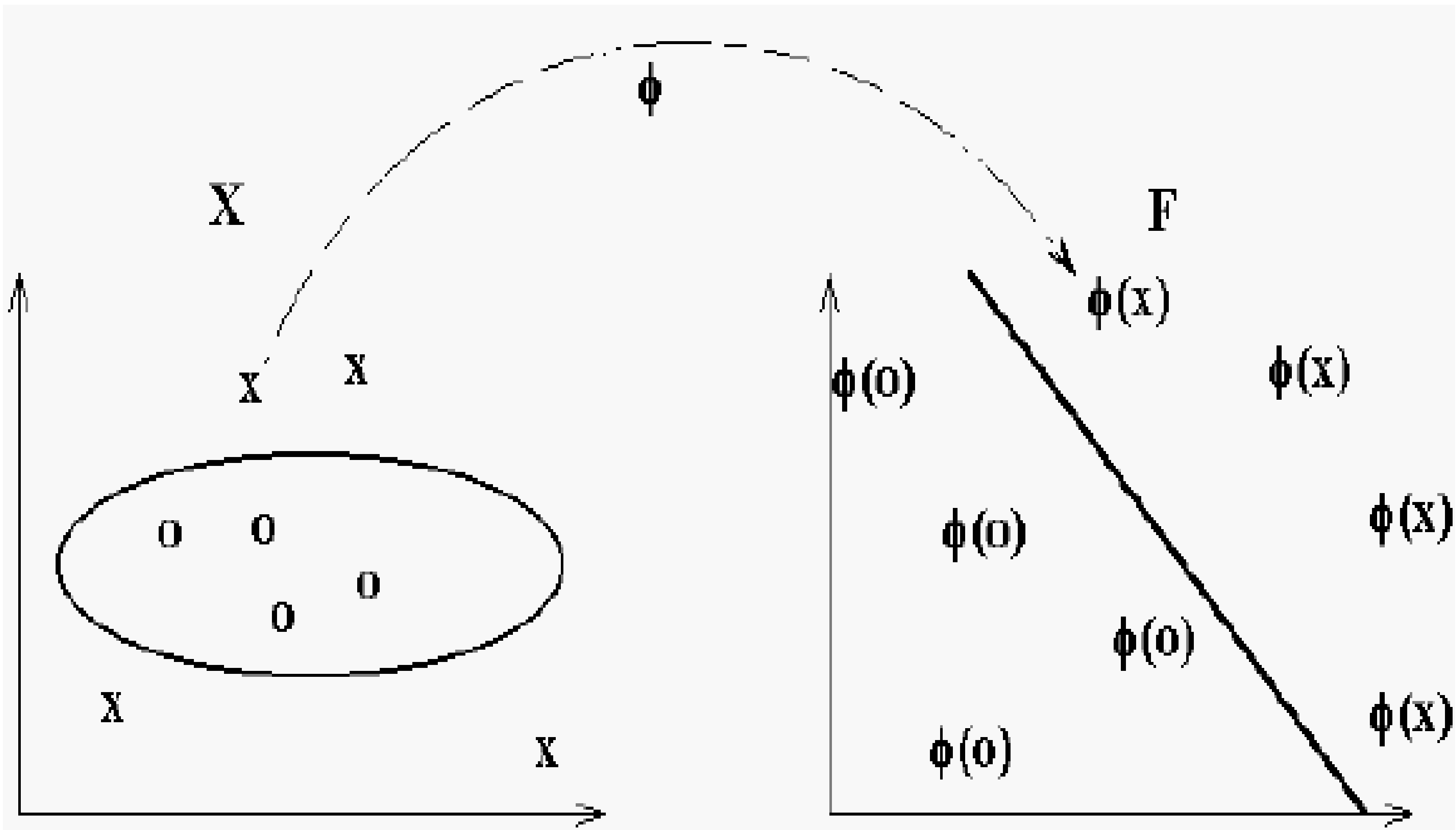
- (Eg: the perceptron)



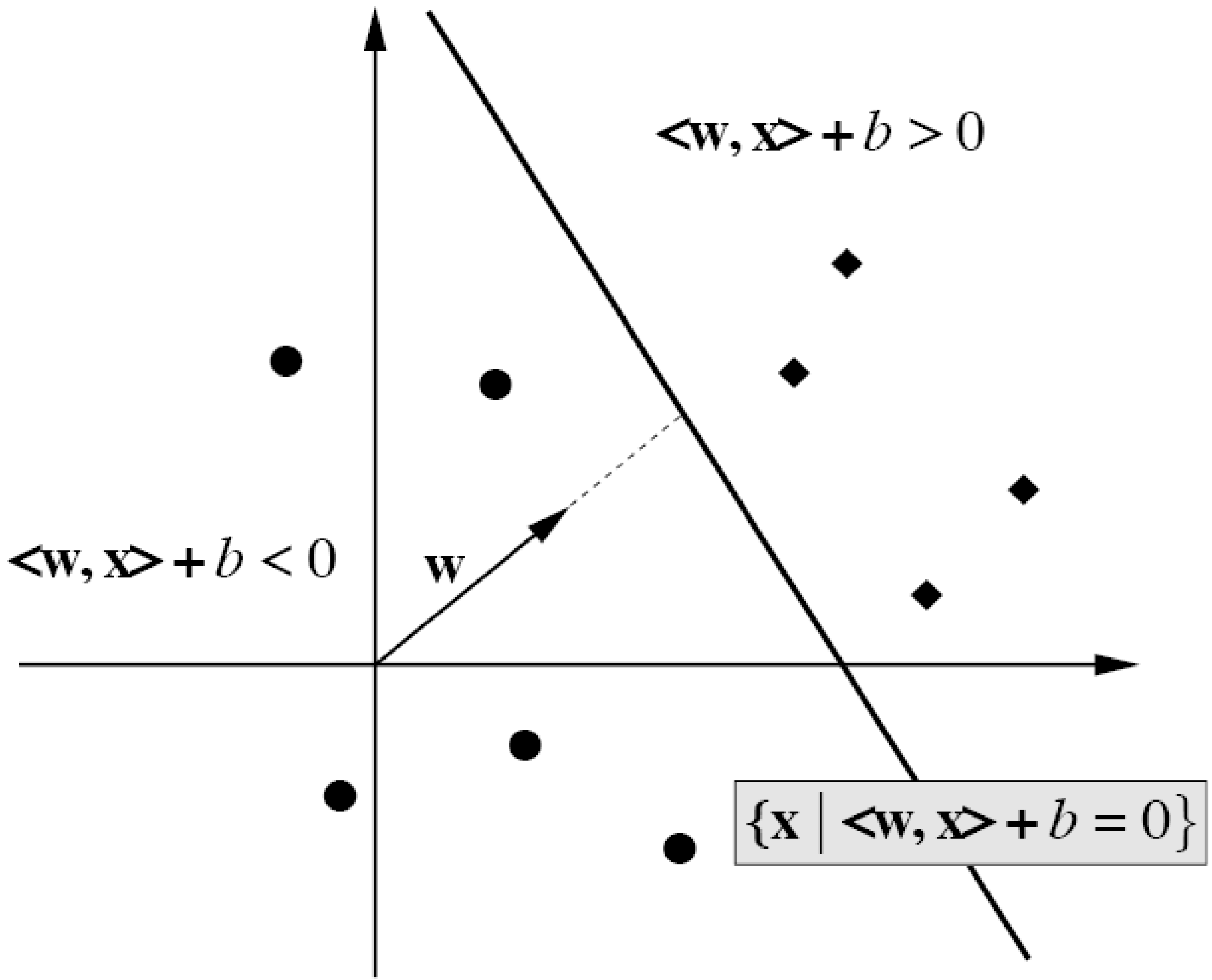
SVM -- extensions

- SVM originally developed in 1960's
- 1990's
  - apply svm to non-linear kernels
  - allow some errors in classification decisions

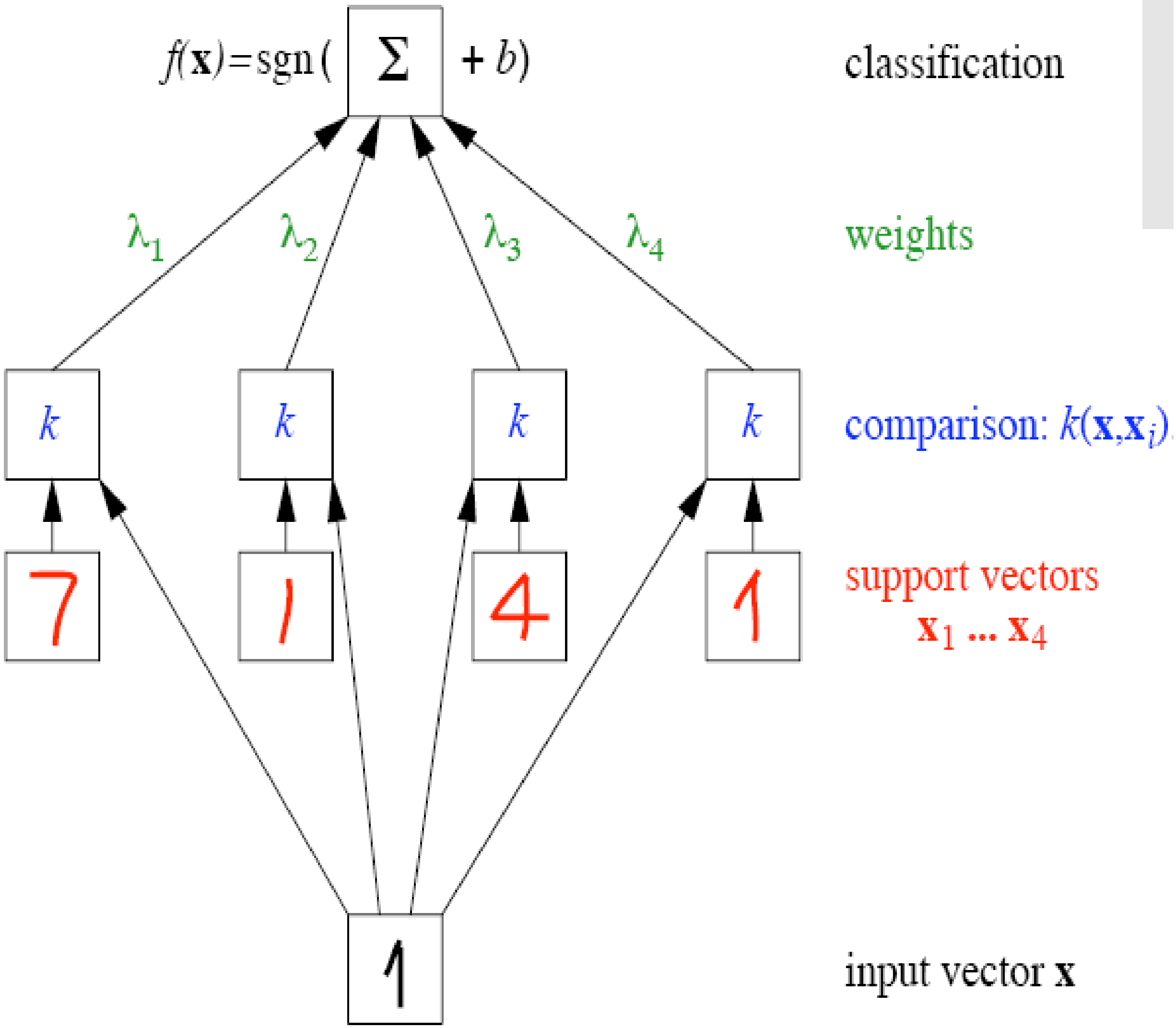
Non-linear SVM



Identifying optional discriminant



SVM – in use



Note similarity to adaboost in making class decision

SVM – in use

Classifier	test error
linear classifier	8.4%
3-nearest-neighbour	2.4%
SVM	1.4%
Tangent distance	1.1%
LeNet4	1.1%
Boosted LeNet4	0.7%
Translation invariant SVM	0.56%

- Problem is handwritten digit recognition
  - e.g. zip codes
- LeNet is a highly optimized NN

SVM -- commentary

- SVM has reputation of handling high dimensional spaces very well
  - because it finds low dimensional projections?
- Often applied to textual analysis
- my guess: many of the DARPA grand challenge cars use SVMs at their code