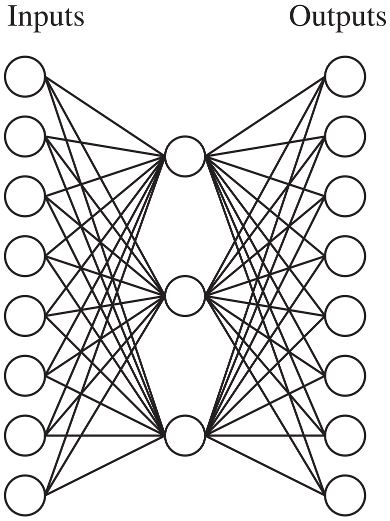
Learning Hidden Layer Representations

A network:



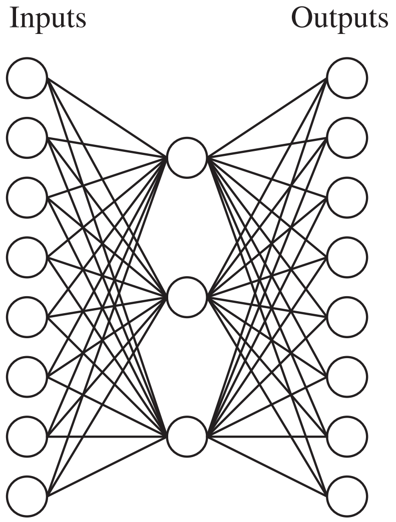
A target function:

Input		Output
10000000	$\rightarrow$	10000000
01000000	$\rightarrow$	01000000
00100000	$\rightarrow$	00100000
00010000	$\rightarrow$	00010000
00001000	$\rightarrow$	00001000
00000100	$\rightarrow$	00000100
00000010	$\rightarrow$	00000010
00000001	$\rightarrow$	00000001

Can this be learned??

Learning Hidden Layer Representations

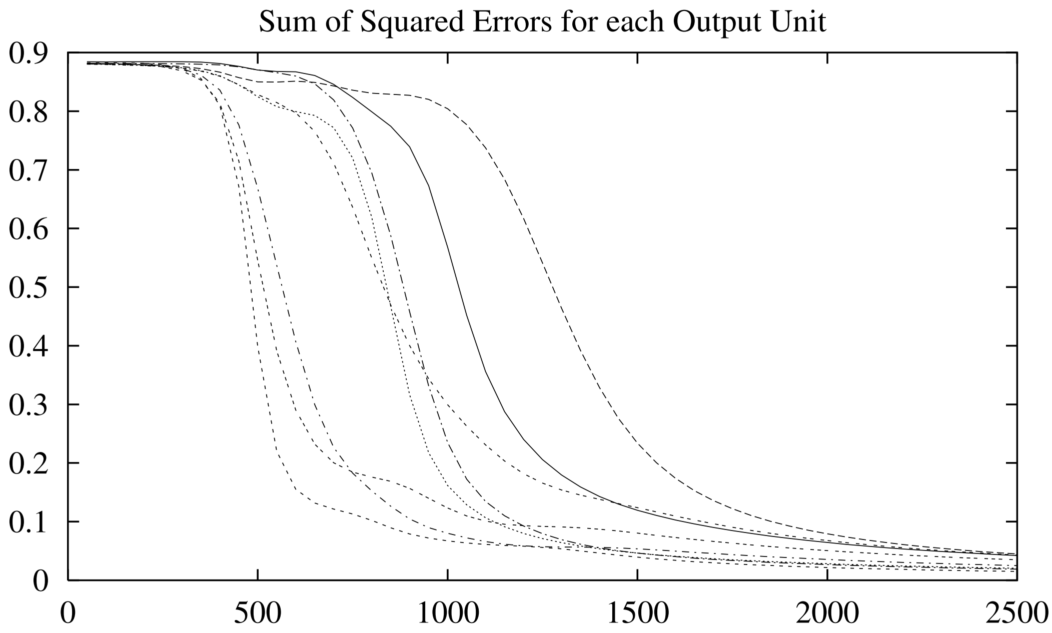
A network:

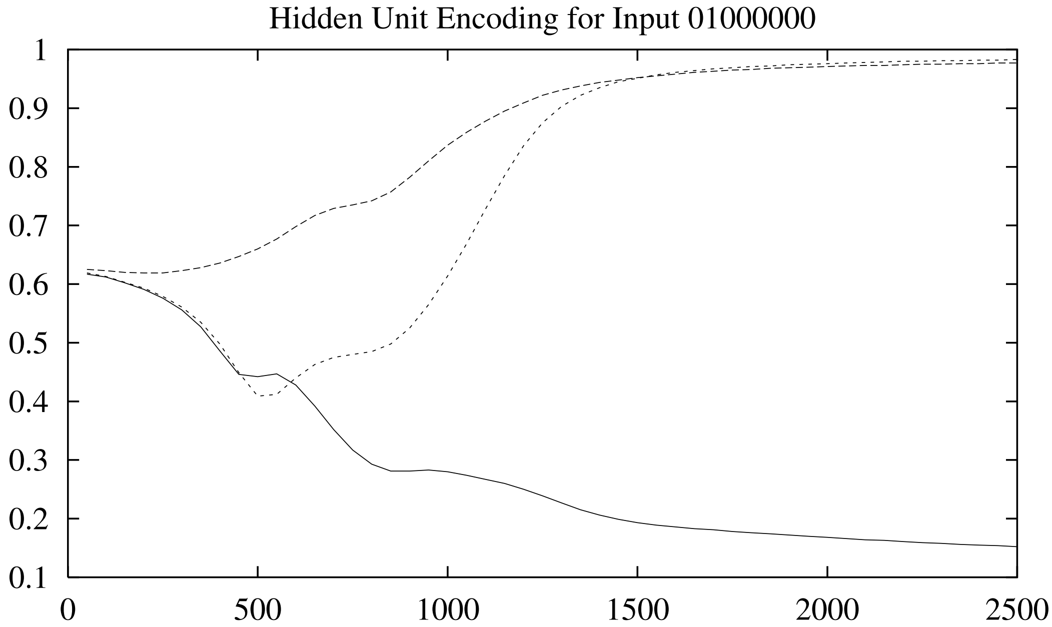


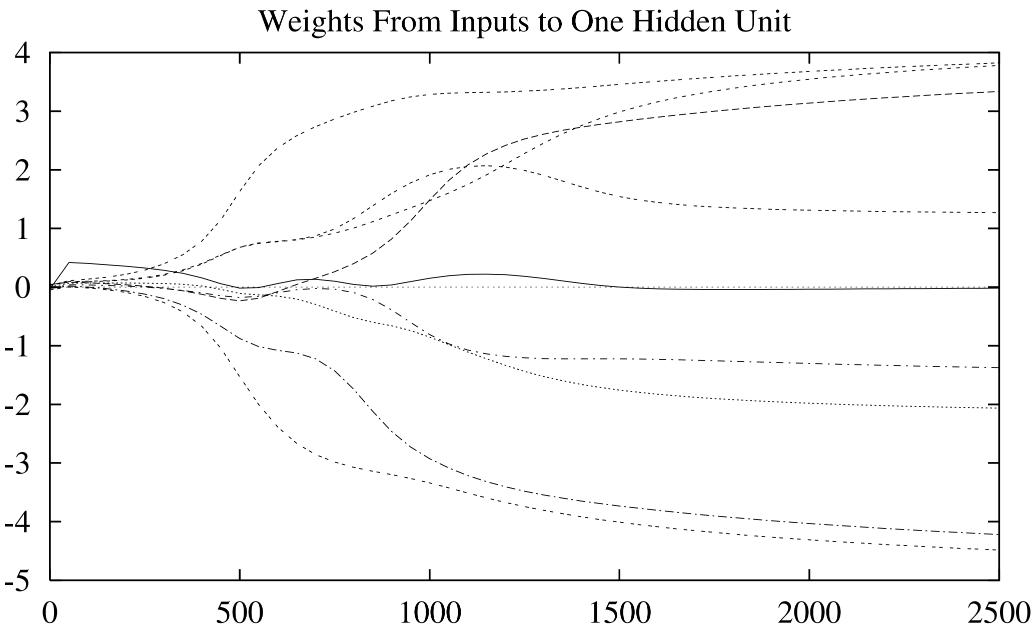
Learned hidden layer representation:

Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.01	.11	.88	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.22	.99	.99	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

i.e., binary code for $1, 2, \ldots 8$!


Sum of Squared Errors for each Output Unit
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500


Hidden Unit Encoding for Input 01000000
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
500
1000
1500
2000
2500


Weights From Inputs to One Hidden Unit
4
3
2
1
0
-1
-2
-3
-4
-5
0
500
1000
1500
2000
2500

Expressive Capabilities of Networks

Boolean functions:

- Every boolean function representable by some feedforward network with just one hidden layer
- But might require exponential (in number of inputs) hidden units

Continuous functions:

- Every *continous* function can be approximated arbitrarily closely by some feedforward network with one hidden layer [Cybenko, 1989], [Hornik et al., 1989]
- might require arbitrarily wide hidden layer

Arbitrary functions:

- *Any* function can be approximated arbitrarily closely using two hidden layers [Cybenko, 1988]

How many hidden units/layers should one include?

Hypothesis space search in Backpropagation

Hypothesis space

- Continuous rather than discrete
- Gradient descent a general search method for continuous H

Inductive bias (assuming cross-validation for early stopping)

- Interpolation, smoothing
- Preference for nets with small weights

Midterm Summary: Key Ideas

1. Learning as search through hypothesis space H
   - Boolean fns represented by conjunctions, disjunctions, ...
   - Discrete-valued functions represented by decision trees
   - Real-valued functions represented by ANNs
   - ...
2. Search strategies follow from structure of H
   - General-to-specific ordering
   - Simple-to-complex ordering
   - Continuity, differentiability $\rightarrow$ gradient descent
   - ...

Midterm Summary: Key Ideas

3. Inductive bias is essential
   - Without it, no generalization
   - Restricted hypothesis space $H \subset 2^X$
   - Preference ordering over H

4. Statistical arguments allow bounding future error
   - PAC learning bounds
   - Chernov bounds
   - VC dimension, H, m, ϵ, δ

Some Topics We've Covered

INTRODUCTION
Well-posed learning problems
Design tradeoffs, choosing target fn, representation, algorithm, source of training examples, ...

CONCEPT LEARNING
Learning as function approximation
Hypothesis space, Instance space
General-to-specific ordering
Version space
Active generation of training queries
Inductive Bias
Futility of unbiased learning

DECISION TREE LEARNING
Representation
Entropy, Mutual information
ID3 / C4.5
Inductive bias, Occam's razor
Avoiding overfitting
Train and test data
Postpruning the tree

COMPUTATIONAL LEARNING THEORY
Error as probability of misclassification
PAC learning model
ϵ-exhausting the version space
Agnostic learning
Sample complexity bounds
VC dimension

NEURAL NETWORKS

- Perceptron
- Linear unit
- Sigmoid Unit
- Multilayer nets
- Gradient descent
- Learning hidden representations