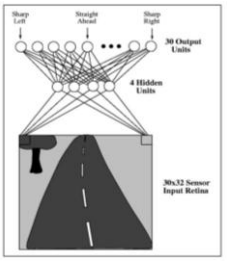
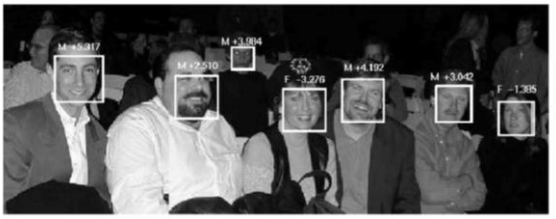
CSE 473

Lecture 27
(Chapter 18)

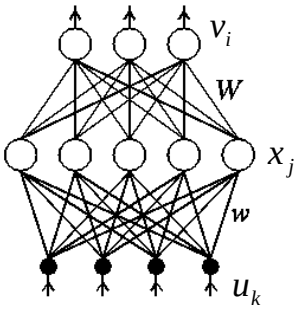
Neural Networks and Ensemble Learning





Recall: Learning in Multilayer Networks

$$v_i = g(\sum_j W_{ji}\, g(\sum_k w_{kj} u_k))$$



Start with random weights **W**, **w**

Given input vector **u**, network produces output vector **v**

Use gradient descent to find **W** and **w** that minimize total error over all output units (labeled *i*):

$$E(\mathbf{W},\mathbf{w}) = \frac{1}{2}\sum_i (d_i - v_i)^2$$

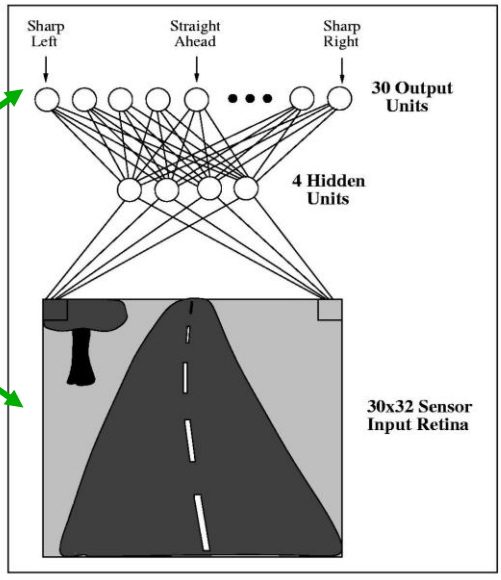
This leads to the famous "Backpropagation" learning rule

$$W_{ji} \rightarrow W_{ji} - \varepsilon \frac{dE}{dW_{ji}} \qquad w_{kj} \rightarrow w_{kj} - \varepsilon \frac{dE}{dw_{kj}} = w_{kj} - \varepsilon \frac{dE}{dx_j} \cdot \frac{dx_j}{dw_{kj}}$$

Example: Learning to Drive

How would you use a neural network to drive?

Example Network



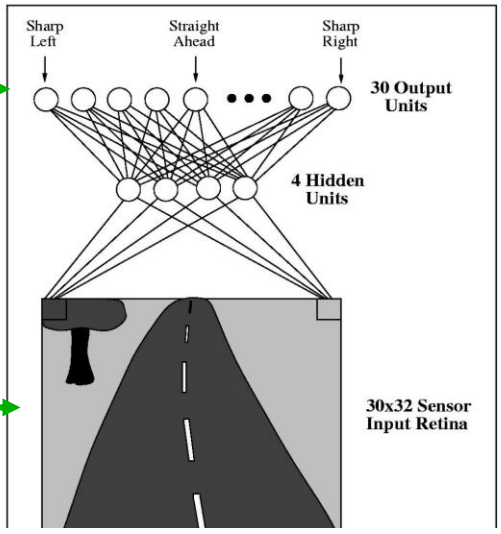
Where will the training inputs come from?

Example Network

Get steering angle from a human driver

Training Output:
$d = (d_1\ d_2\ \ldots\ d_{30})$

Get current camera image

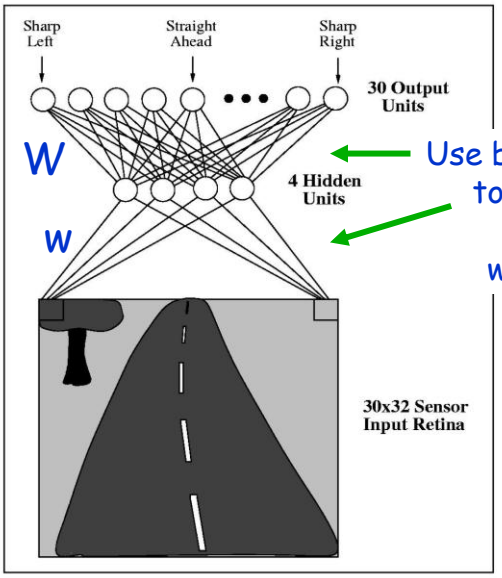


Training Input $u = (u_1\ u_2\ \ldots\ u_{960})$ = image pixels

5

Example Network

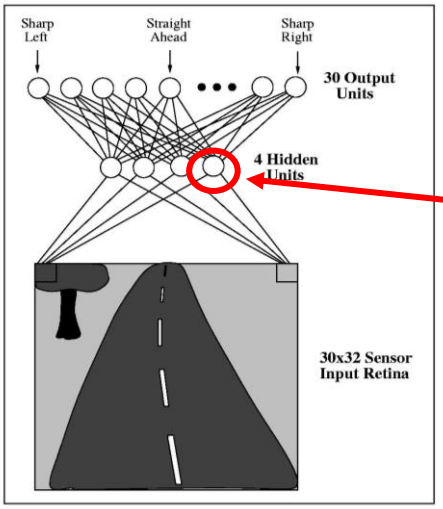
You are given training input-output pairs (**u**,**d**)



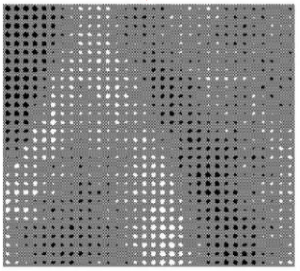
Use backprop to modify these weights

6

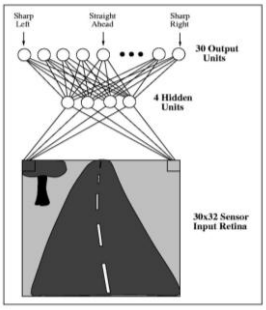
Results



One of the learned "road features" w_i



ALVINN (Autonomous Land Vehicle in a Neural Network)



CMU Navlab

Trained using human driver + camera images

After learning:

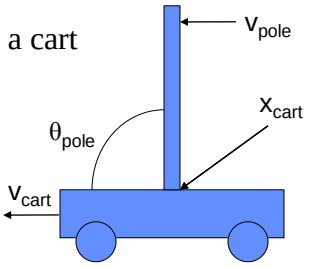
- Drove up to 70 mph on highway
- Up to 22 miles without intervention
- Drove cross-country largely autonomously

(Pomerleau, 1992)

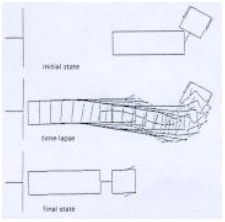
Demos: Pole Balancing and Backing up a Truck

(courtesy of Keith Grochow, CSE 599)

- Neural network learns to balance a pole on a cart
  - System:
    - 4 state variables: x_{cart}, v_{cart}, θ_{pole}, v_{pole}
    - 1 input: Force on cart
  - Backprop Network:
    - Input: State variables
    - Output: New force on cart



- NN learns to back a truck into a loading dock
  - System (Nyugen and Widrow, 1989):
    - State variables: x_{cab}, y_{cab}, θ_{cab}
    - 1 input: new $\theta_{steering}$
  - Backprop Network:
    - Input: State variables
    - Output: Steering angle $\theta_{steering}$



9

Ensemble Learning

Sometimes each learning technique yields a different "hypothesis" (function)

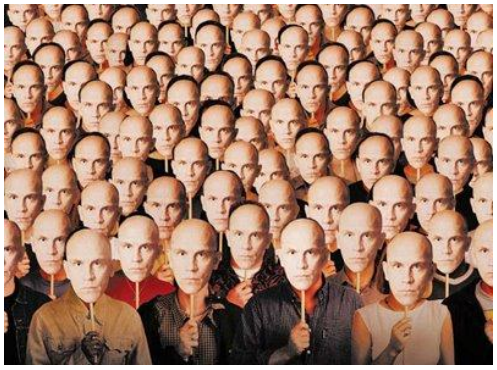
But no perfect hypothesis…

Could we combine several imperfect hypotheses to get a better hypothesis?

10

Ensemble Learning

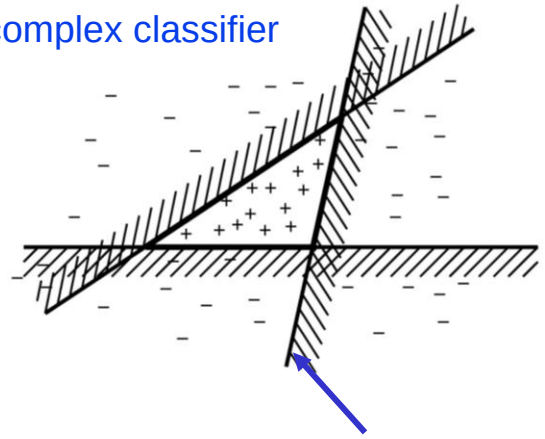
Wisdom of the Crowds



Example

Combine 3 linear classifiers

⇒ More complex classifier



This line is one simple classifier saying that
everything to the left is + and everything to the right is -

12

Ensemble Learning: Motivation

Analogies:

- Elections combine voters' choices to pick a good candidate (hopefully)
- Committees combine experts' opinions to make better decisions
- Students working together on a capstone project

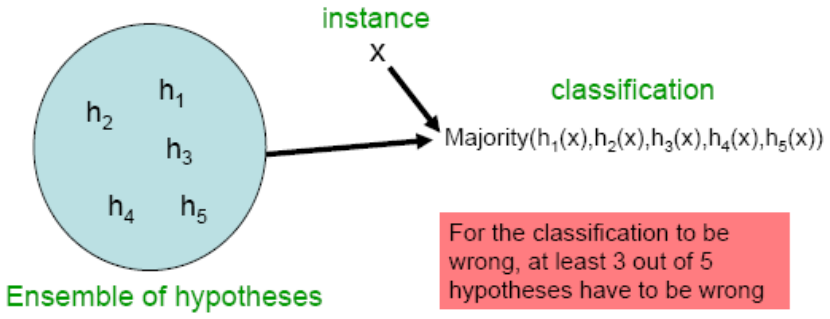
Intuitions:

Individuals make mistakes but the "majority" may be less likely to

Individuals often have partial knowledge; a committee can pool expertise to make better decisions

Ensemble Technique 1: Bagging

Combine hypotheses via majority voting



Bagging: Details

1. Generate *m* new training datasets by sampling with replacement from the given dataset
2. Train *m* classifiers $h_1,\ldots,h_m$ (e.g., decision trees), one from each newly generated dataset
3. Classify a new input by running it through the *m* classifiers and choosing the class that receives the most "votes"

Example: *Random forest* = Bagging with *m* decision tree classifiers, each tree constructed from random subset of attributes

Bagging: Analysis

- Assumptions:
  - Each h_i makes error with probability p
  - The hypotheses are independent
- Majority voting of n hypotheses:
  - k hypotheses make an error:
  - Majority makes an error: $\Sigma_{k>n/2} \binom{n}{k} p^k(1-p)^{n-k}$
  - With n=5, p=0.1 ➔ err(majority) < 0.01

Error probability went down from 0.1 to 0.01!

16

Weighted Majority Voting

In practice, hypotheses rarely independent

Some hypotheses have less errors than others ⇒ all votes are not equal!

Idea: Let's take a weighted majority

How do we compute the weights?

17

Ensemble Technique 2: Boosting

Operates on a weighted training set

- Each training example (instance) has a "weight"
- Best classifier (hypothesis) is one that has smallest total *weighted* classification error

Idea: when an input is misclassified by a hypothesis, increase its weight so that the ***next hypothesis*** is more likely to classify it correctly

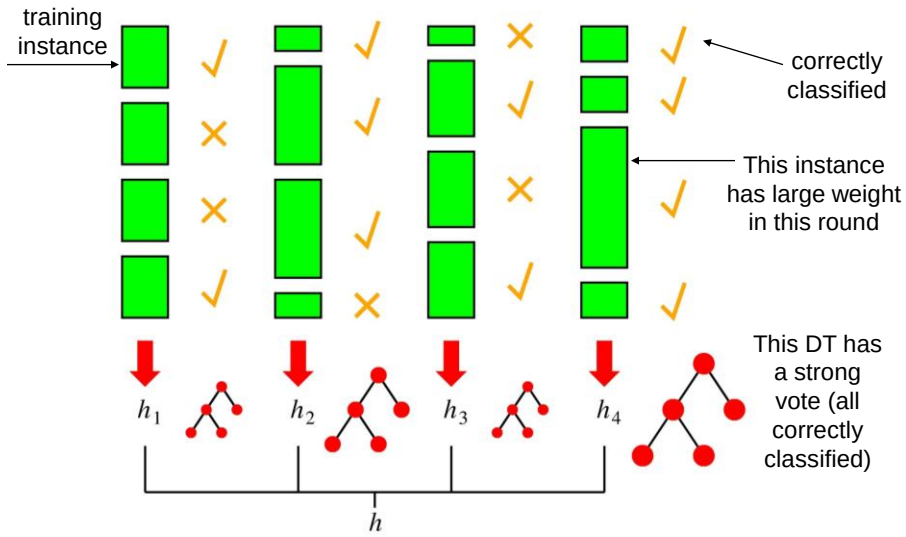
Output is weighted majority of all hypotheses

Why "boosting"?

Can "boost" performance of a "weak learner"

18

Example: Boosting with Decision Trees (DTs)



Output of h_{final} is weighted majority of outputs of $h_1,\ldots,h_4$

Next Time

- More on Boosting
- Survey of Applications of AI
- To Do:
  - Project 4 due tonight!
  - Finish Chapter 18