Neural Networks and their applications

Garrison W. Cottrell

Computer Science & Engineering Department

and

Institute for Neural Computation

University of California, San Diego

Outline of the talk

I. Motivation: Why neural nets?

II. Human-style computation: What’s it like?

III. Neural nets: What are they like?

IV. Example applications

- A. NETTalk
- B. Face and emotion recognition

V. Conclusions and some things to ponder

Introduction

- Why are people smarter than machines?

- Our take on the problem:

    - Basic differences in architecture

- Our research program:

Explore brain-like computational models

Mutual Constraints

People are able to combine *lots* of different kinds of knowledge *quickly* in understanding English -

For example, in understanding the relationships given in a sentence:

Syntax (structure) gives us some information:

The boy kissed the girl.

But usually we need semantics (meaning) too:

I saw the grand canyon *flying to New York*.

I saw the sheep *grazing in the field*.

Ditto for pronoun reference:

The city council refused the demonstrators a permit because *they* were communists. [Russia or China?]

Again, things that people do well involve integrating information from multiple sources.

E.g., word sense disambiguation:

1. Discourse Context: "I’m taking the **car**"

2. Grammar: "The carpenter **saw** the wood"

3. Meaning frequency: "Bob **threw** a **ball**"

4. Semantics:
   - Associations between word senses
     "dog’s bark" "deep pit"

   - The fit between roles and role fillers:
     "Bob **threw** the fight"

5. Pragmatic:
   "The man walked on the **deck**"
   "Nadia swung the hammer at the nail and the **head** flew off"(Hirst 83)

Computers are different ...

The boy the girl the dog bit liked cried.

U CN REED THIIS CANDT YU?

Constraints on a brain model

• The 100 step program constraint (Feldman, 1985)

- Neurons are slow:

  respond in a few milliseconds

- Yet mental events only take half a second

This implies:

- parallelism

- simple steps

Constraints on a brain model

	Computer	Brain
speed	fast	slow
order	serial	parallel
component reliablity	good	poor
fault tolerance	poor	no degradation
signals	precise symbolic	imprecise, terse
programming	needs it	does it

Parallel Distributed Processing Models

- Networks of simple processing "units"

- connected by positive and negative links

- spread *activation* and *inhibition*
  to other units

- The "solution" is a *stable state* -

  when nothing changes

An Example: The Necker Cube

- Units represent *hypotheses* about the inputs
- Connections encode *constraints* between hypotheses
- The network *relaxes* to a stable state: the "interpretation"

Learning

A bit of history:

Rosenblatt (1962) discovered the *perceptron convergence procedure*

- Guaranteed to learn anything computable

  (by a two-layer perceptron)

- Unfortunately, not everything was computable

  (Minsky & Papert, 1969)

Learning

Rumelhart, Hinton and Williams (1985) have (re-)discovered a learning technique for multiple layered nets, called *back-propagation*.

Here's how it works:

- present an input, let activation propagate through the network

- give a *teaching signal*

- propagate the error back through the network (hence the name *back propagation*)

- change the connections according to the error

Learning

Networks using back-propagation have learned:

- exclusive-or
- to compress images
- to retrieve properties of animals and plants from partial descriptions
- to read aloud from text
- to recognize faces
- to recognize emotion in faces
- to drive a Chevy Van at 55 MPH
- to predict financial markets

- to detect credit card fraud

NETtalk
(Terrence Sejnowski & Charles Rosenberg, 1986)

• 7 groups of input units for seven letters

• Job is to produce phoneme corresponding to middle letter

• output is fed to DECTALK speech synthesizer

NETtalk

Training corpus

Applications: Face and Emotion Recognition

Recognizing faces is:

hard! (And recalling names is worse...)

Recognizing emotions is harder.

seems easy for us but:

easy to make mistakes

a lot of cortex is dedicated to it (~10% of temporal cortex)

a subsystem: Prosopagnosia

Useful: Identifying criminals, political dissidents...

Computer Interfaces: feedback to adaptive systems

Nuclear plant operator monitoring - awake?

Applications: Face Recognition

The Network:

The left side network is an *auto-encoder*:

It is trained to simply *reproduce* its input.

The right hand network does classification using the features extracted by the left hand network.

Conclusions

- Neural nets are doing some amazing things!
  - Especially good at pattern recognition
  - Good at prediction
- Neural nets feature:
  - Uniform representation of multiple knowledge sources
  - "Automatic" generalization
  - Adaptability
  - Learning
  - Development of its own representations