Lecture 20

Neural Computation

Thursday, November 4, 1999

William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.cis.ksu.edu/~bhsu

Readings:
Chapter 19, Russell and Norvig
Section 4.8, Mitchell
Section 4.1.3, Buchanan and Wilkins (Hinton)

Lecture Outline

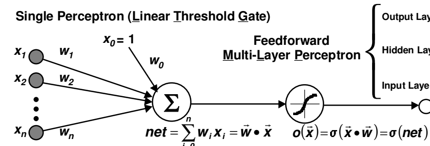
- Readings: Chapter 19, Russell and Norvig; Section 4.8, Mitchell
- Suggested Exercises: 4.6, Mitchell
- Paper Review: "Connectionist Learning Procedures" [Hinton, 1989]
- Review: Feedforward Artificial Neural Networks (ANNs)
- Advanced ANN Topics: *Survey*
  - Models
    - Associative memories
    - Simulated annealing and Boltzmann machines
    - Modular ANNs; temporal ANNs
  - Applications
    - Pattern recognition and scene analysis (image processing)
    - Signal processing (especially time series prediction)
    - Neural reinforcement learning
- Relation to Bayesian Networks
- Next Week: Combining Classifiers

Artificial Neural Networks

- Basic Neural Computation: Earlier
  - Linear threshold gate (LTG)
    - Model: single neural processing element
    - Training rules: perceptron, delta / LMS / Widrow-Hoff, winnow
  - Multi-layer perceptron (MLP)
    - Model: feedforward (FF) MLP
    - Temporal ANN: simple recurrent network (SRN), TDNN, Gamma memory
    - Training rules: error backpropagation, backprop with momentum, backprop through time (BPTT)
- Associative Memories
  - Application: robust pattern recognition
  - Boltzmann machines: constraint satisfaction networks that learn
- Current Issues and Topics in Neural Computation
  - Neural reinforcement learning: incorporating knowledge
  - Principled integration of ANN, BBN, GA models with symbolic models

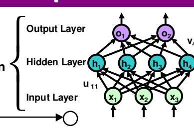
Quick Review: Feedforward Multi-Layer Perceptrons

Single Perceptron (Linear Threshold Gate)

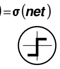


$net = \sum_{i=0}^{n} w_i x_i = \vec{w} \bullet \vec{x}$ $\quad o(\vec{x}) = \sigma(\vec{x} \bullet \vec{w}) = \sigma(net)$

Feedforward Multi-Layer Perceptron



- Sigmoid Activation Function
  - Linear threshold gate activation function: *sgn* ($w \bullet x$)
  - Nonlinear activation (*aka* transfer, squashing) function: generalization of *sgn*
  - σ is the sigmoid function $\sigma(net) = \frac{1}{1 + e^{-net}}$
  - Can derive gradient rules to train
    - One sigmoid unit
    - Multi-layer, feedforward networks of sigmoid units (using backpropagation)
- Hyperbolic Tangent Activation Function $\sigma(net) = \frac{\sinh(net)}{\cosh(net)} = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}$



Quick Review: Backpropagation of Error

- Recall: Backprop Training Rule Derived from Error Gradient Formula
- Algorithm *Train-by-Backprop* (*D*, *r*)
  - *r*: constant learning rate (e.g., 0.05)
  - Initialize all weights w_i to (small) random values
  - UNTIL the termination condition is met, DO

    FOR each <*x*, *t*(*x*)> in *D*, DO

    Input the instance *x* to the unit and compute the output $o(x) = \sigma(net(x))$

    FOR each output unit *k*, DO

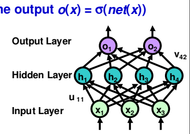
    FOR each hidden unit *j*, DO

    $\delta_j = h_j(x)(1 - h_j(x)) \sum v_{j,k} \delta_k$

    Update each $w = u_{i,j}$ ($a = h_j$) or $w = v_{j,k}$ ($a = o_k$)

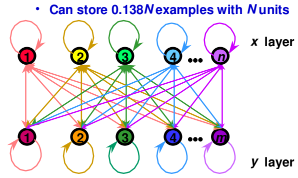
    $w_{start\text{-}layer,\ end\text{-}layer} \leftarrow w_{start\text{-}layer,\ end\text{-}layer} + \Delta w_{start\text{-}layer,\ end\text{-}layer}$

    $\Delta w_{start\text{-}layer,\ end\text{-}layer} \leftarrow r\, \delta_{end\text{-}layer}\, a_{end\text{-}layer}$
  - RETURN final *u*, *v*

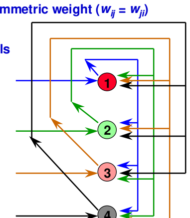


Associative Memory

- Intuitive Idea
  - Learning ANN: trained on a set *D* of examples x_i
  - New stimulus *x'* causes network to settle into activation pattern of closest *x*
- Bidirectional Associative Memory (19.2, Russell and Norvig)
  - Propagates information in either direction; symmetric weight ($w_{ij} = w_{ji}$)
  - Hopfield network
    - Recurrent; BAM with +1, -1 activation levels
    - Can store 0.138*N* examples with *N* units

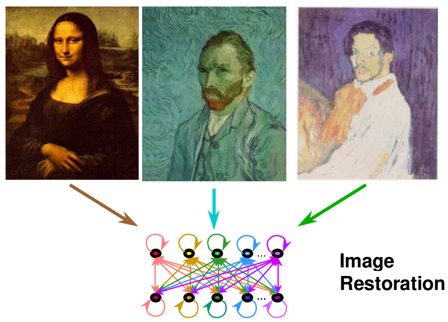


Bidirectional Associative Memory



Hopfield Network

Associative Memory and Robust Pattern Recognition



Simulated Annealing

- Intuitive Idea
  - Local search: susceptible to relative optima
  - Frequency ∝ deceptivity of search space
  - Solution approaches
    - Nonlocal search frontier (A^*)
    - Stochastic approximation of Bayes optimal criterion
- Interpretation as Search Method
  - Search: transitions from one point in state (hypothesis, policy) space to another
  - *Force* search out of local regions by accepting *suboptimal state transitions* with decreasing probability
- Statistical Mechanics Interpretation
  - See: [Kirkpatrick, Gelatt, and Vecchi, 1983; Ackley, Hinton, and Sejnowski, 1985]
  - Analogies
    - Real annealing: cooling molten material into solid form (versus quenching)
    - Finding relative minimum of potential energy (objects rolling downhill)

Boltzmann Machines

- Intuitive Idea
  - Synthesis of associative memory architecture with global optimization algorithm
  - Learning by *satisfying constraints* [Rumelhart and McClelland, 1986]
- Modifying Simple Associative Memories
  - Use BAM-style model (symmetric weights)
    - Difference vs. BAM architecture: have hidden units
    - Difference vs. Hopfield network training rule: stochastic activation function
  - Stochastic activation function: simulated annealing or other MCMC computation
- Constraint Satisfaction Interpretation
  - Hopfield network (+1, -1) activation function: simple *boolean constraints*
  - Formally identical to BBNs evaluated with MCMC algorithm [Neal, 1992]
- Applications
  - Gradient learning of BBNs to simulate ANNs (sigmoid networks [Neal, 1991])
  - Parallel simulation of Bayesian network CPT learning [Myllymaki, 1995]

ANNs and Reinforcement Learning

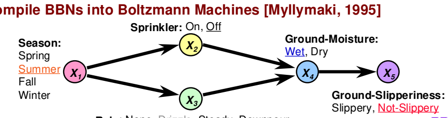
- Adaptive Dynamic Programming (ADP) Revisited
  - Learn value and state transition functions
  - Can substitute ANN for HMM
    - Neural learning architecture (e.g, TDNN) takes place of transition, utility tables
    - Neural learning algorithms (e.g., BPTT) take place of ADP
- Neural *Q*-Learning
  - Learn action-value function (Q : *state* × *action* → *value*)
  - Neural learning architecture takes place of Q tables
  - Approximate *Q*-Learning: neural TD
    - Neural learning algorithms (e.g., BPTT) take place of $TD(\lambda)$
    - *NB*: can do this even with implicit representations and save!
- Neural Reinforcement Learning: Course Online
  - Anderson, Spring 1999
  - http://www.cs.colostate.edu/~cs681

ANNs and Bayesian Networks

- BBNs as Case of ANNs
  - BBNs: another connectionist model (graphical model of mutable state)
    - Bayesian networks: state corresponds to beliefs (probabilities)

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

    - ANNs: state corresponds to activation in FF computation, weights in learning
    - *Local computation* (CPT tables; ANN unit activations, gradients)
  - Duality: asymmetric parallel Boltzmann machines ↔ BBNs
- Can Compile BBNs into Boltzmann Machines [Myllymaki, 1995]



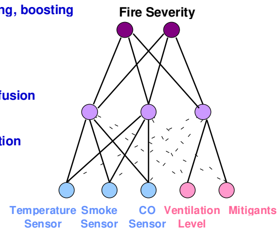
P(*Summer*, *Off*, *Drizzle*, *Wet*, *Not-Slippery*) = *P*(*S*) · *P*(*O* | *S*) · *P*(*D* | *S*) · *P*(*W* | *O*, *D*) · *P*(*N* | *W*)

ANNs and Genetic Algorithms

- Genetic Algorithms (GAs) and Simulated Annealing (SA)
  - Genetic algorithm: 3 basic components
    - Selection: propagation of fit individuals (proportionate reproduction, tournament selection)
    - Crossover: combine individuals to generate new ones
    - Mutation: stochastic, localized modification to individuals
  - Simulated annealing: can be defined as genetic algorithm
    - Selection, mutation only
    - Simple SA: single-point population (serial trajectory)
    - More on this next week
- Global Optimization: Common ANN/GA Issues
  - MCMC: *When is it practical?* e.g., scalable?
  - How to control high-level parameters (population size, hidden units; priors)?
  - How to incorporate knowledge, extract knowledge?

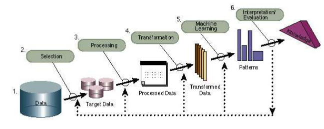
Advanced Topics

- Modular ANNs
  - Hierarchical Mixtures of Experts
    - Mixture model: combines outputs of simple neural processing units
    - Other combiners: bagging, stacking, boosting
    - More on combiners later
  - Modularity in neural systems
    - Important topic in neuroscience
    - Design choices: sensor and data fusion
- Bayesian Learning in ANNs
  - Simulated annealing: global optimization
  - Markov chain Monte Carlo (MCMC)
- Applied Neural Computation
  - Robust image recognition
  - Time series analysis, prediction
  - Dynamic information retrieval (IR), e.g., hierarchical indexing



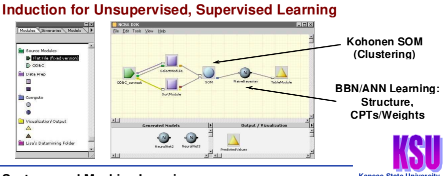
ANNs: Application to Data Mining

- Knowledge Discovery in Databases (KDD)



An Overview of the Steps That Compose the KDD Process

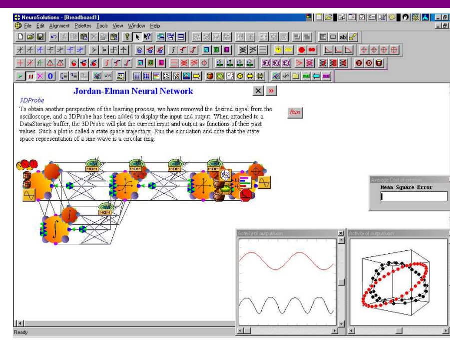
- Role of ANN Induction for Unsupervised, Supervised Learning



ANN Resources

- Simulation Tools
  - Open source
    - *Stuttgart Neural Network Simulator* (*SNNS*) for Linux
    - http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/
  - Commercial
    - *NeuroSolutions* for Windows NT
    - http://www.nd.com
- Resources Online
  - ANN FAQ: ftp://ftp.sas.com/pub/neural/FAQ.html
  - Meta-indices of ANN resources
    - PNL ANN archive: http://www.emsl.pnl.gov:2080/proj/neuron/neural
    - *Neuroprose* (tech reports): ftp://archive.cis.ohio-state.edu/pub/neuroprose
  - Discussion and review sites
    - ANNs and Computational Brain Theory (U. Illinois): http://anncbt.ai.uiuc.edu
    - *NeuroNet*: http://www.kcl.ac.uk/neuronet

NeuroSolutions Demo



Terminology

- Advanced ANN Models
  - Associative memory: system that can recall training examples given new stimuli
    - Bidirectional associative memory (BAM): clamp parts of training vector on both sides, present new stimulus to either
    - Hopfield network: type of recurrent BAM with +1, -1 activation
  - Simulated annealing: Markov chain Monte Carlo (MCMC) optimization method
  - Boltzmann machine: BAM with stochastic activation (cf. simulated annealing)
  - Hierarchical mixture of experts (HME): neural mixture model (modular ANN)
- Bayesian Networks and Genetic Algorithms
  - Connectionist model: graphical model of state and local computation (e.g., beliefs, belief revision)
  - Numerical (*aka* "subsymbolic") learning systems
    - BBNs (previously): probabilistic semantics; uncertainty
    - ANNs: network efficiently representable functions (NERFs)
    - GAs (next): *building blocks*

Summary Points

- Review: Feedforward Artificial Neural Networks (ANNs)
- Advanced ANN Topics
  - Models
    - Modular ANNs
    - Associative memories
    - Boltzmann machines
  - Applications
    - Pattern recognition and scene analysis (image processing)
    - Signal processing
    - Neural reinforcement learning
- Relation to Bayesian Networks and Genetic Algorithms (GAs)
  - Bayesian networks as a species of connectionist model
  - Simulated annealing and GAs: MCMC methods
  - Numerical ("subsymbolic") and symbolic AI systems: principled integration
- Next Week: Combining Classifiers (WM, Bagging, Stacking, Boosting)