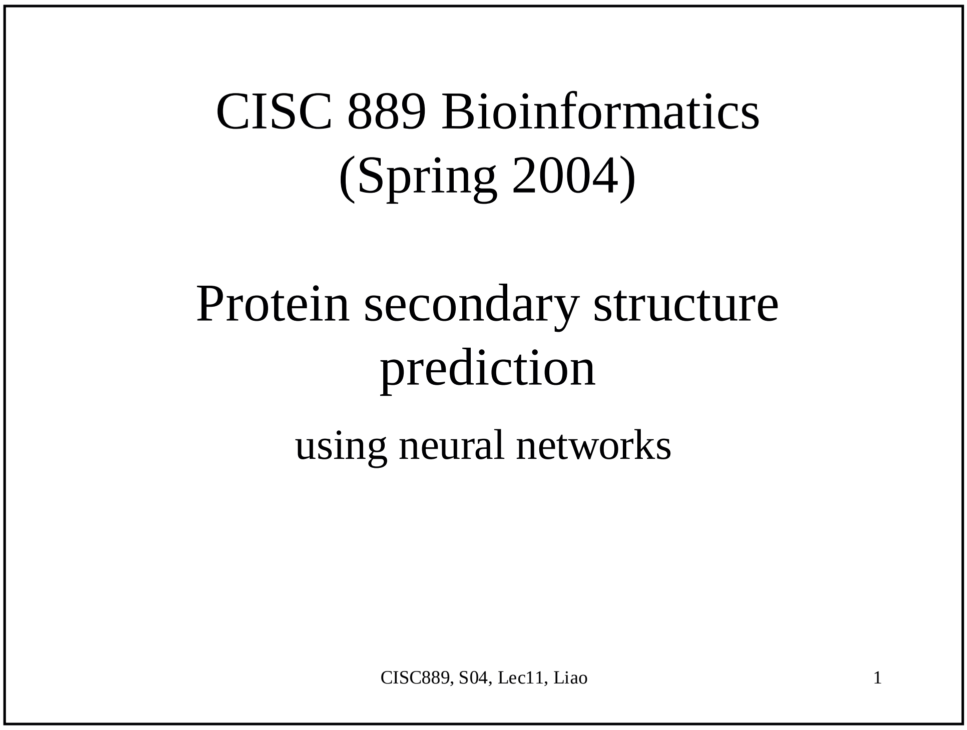
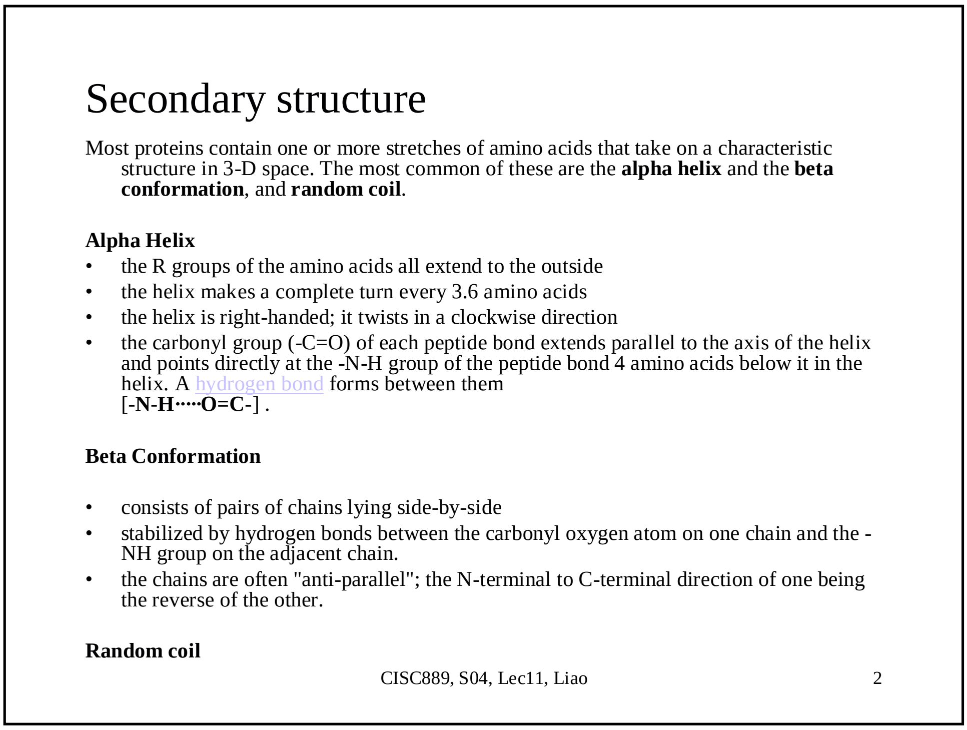


CISC 889 Bioinformatics (Spring 2004)

Protein secondary structure prediction

using neural networks



Secondary structure

Most proteins contain one or more stretches of amino acids that take on a characteristic structure in 3-D space. The most common of these are the **alpha helix** and the **beta conformation**, and **random coil**.

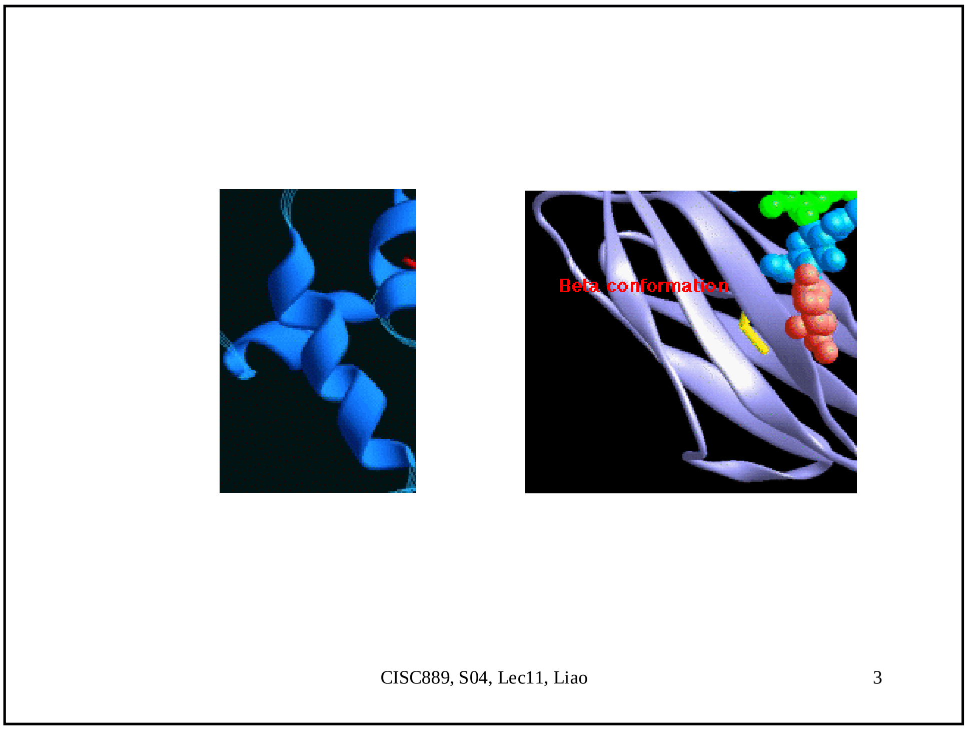
Alpha Helix

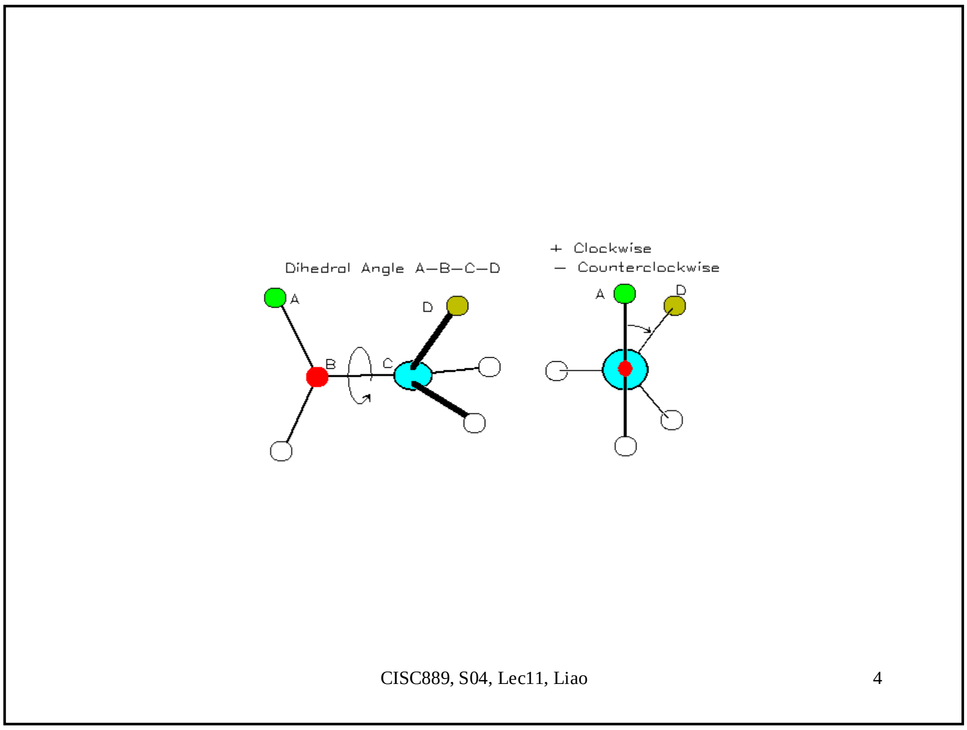
- the R groups of the amino acids all extend to the outside
- the helix makes a complete turn every 3.6 amino acids
- the helix is right-handed; it twists in a clockwise direction
- the carbonyl group (-C=O) of each peptide bond extends parallel to the axis of the helix and points directly at the -N-H group of the peptide bond 4 amino acids below it in the helix. A hydrogen bond forms between them [-**N-H·····O=C-**] .

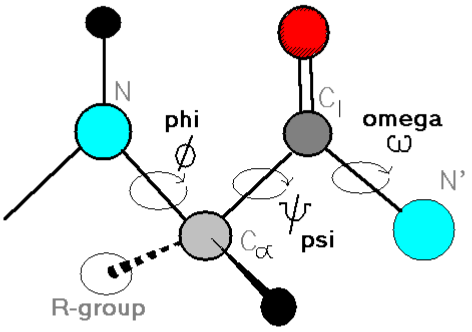
Beta Conformation

- consists of pairs of chains lying side-by-side
- stabilized by hydrogen bonds between the carbonyl oxygen atom on one chain and the -NH group on the adjacent chain.
- the chains are often "anti-parallel"; the N-terminal to C-terminal direction of one being the reverse of the other.

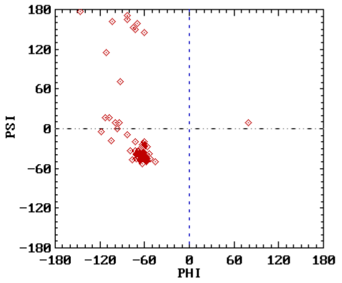
Random coil

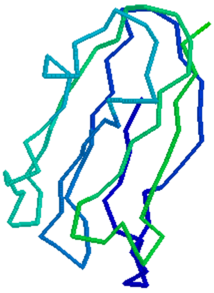

Beta conformation

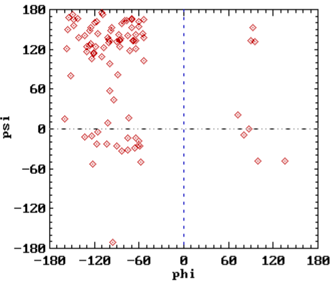

Dihedral Angle A—B—C—D
A
B
C
D
+ Clockwise
— Counterclockwise
A
D



Ramachandran Plot







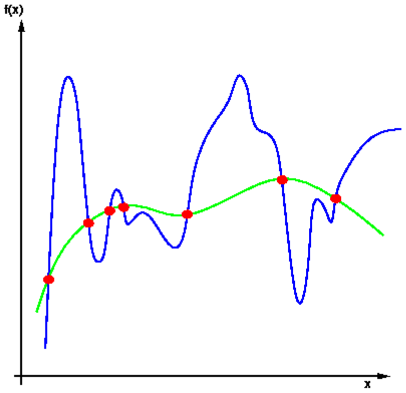
Task:

primary sequence → secondary structures

Approach: Machine learning

- Involve two steps: learning (hard, usually NP) & testing (easy, P)
- What is learning? To improve from experience **E** with respect to some tasks **T** and performance measure **P**.
- How to learn? Often as a search, to find the hypothesis that best fits the training examples.
- Hypotheses:
  - A function f: x → f(x, θ)
  - Issues
    - Expressiveness
    - Generalization (Occam’s razor, see Jeffreys and Berger, American Scientist 80(1992)64)

Which one should we pick?



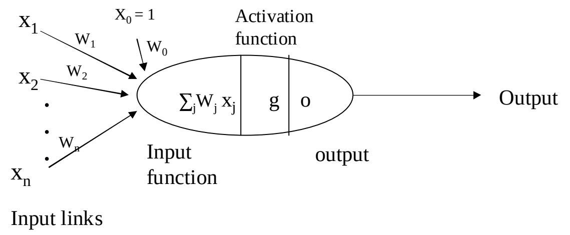
Credit:Alessandro Verri (MIT)

Artificial Neural Networks

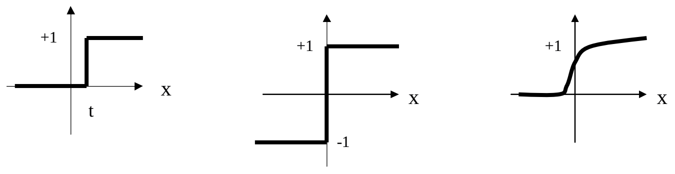
- Problems suitable for ANN to solve
  - Instances are represented by many attribute-value pairs
  - The target function output may be discrete-valued, real-valued, or a vector
  - Training examples may contain errors
  - Long training times are acceptable
  - Fast evaluation of the learned target function may be required
  - The ability of humans to understand the learned target function is not important

Artificial neural networks

- Perceptron $o(x_1, \dots, x_n) = g(\sum_j W_j x_j)$



- Activation functions g



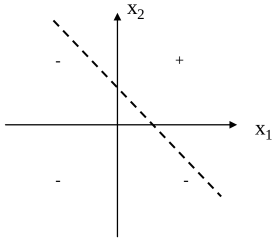
$$\text{Step}(x) = \begin{cases} 1 \text{ if } x \geq t \\ 0 \text{ otherwise} \end{cases} \qquad \text{Sign}(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ -1 \text{ otherwise} \end{cases} \qquad \text{Sigmoid}(x) = 1/(1+e^{-x})$$

- Example: AND function

O(x_1, x_2) = sign(w_0 + w_1 x_1 + w_2 x_2) will behave like AND

for w_0 = -0.8, w_1 = 0.5, and w_2 = 0.5

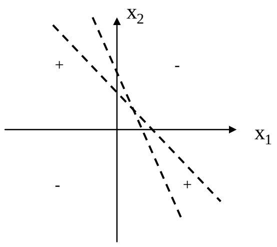
Note: the activation function g is a sign function, and input function is a linear function, which gives a straight line (dotted line in the figure below).



- Example: XOR function

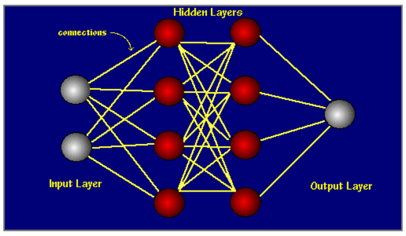
O(x_1, x_2) = sign(w_0 + w_1 x_1 + w_2 x_2)

There are no straight line that can separate + from -



Note: single perceptrons have limited expressive power.

- Neural Networks



Note: multi-layer networks can simulate any function

- For any continuous function

$$y = f(x),\ x \in [0,1]$$

we can construct a neural net h(x) that can approximate f(x) within any error range e. There exist n such that

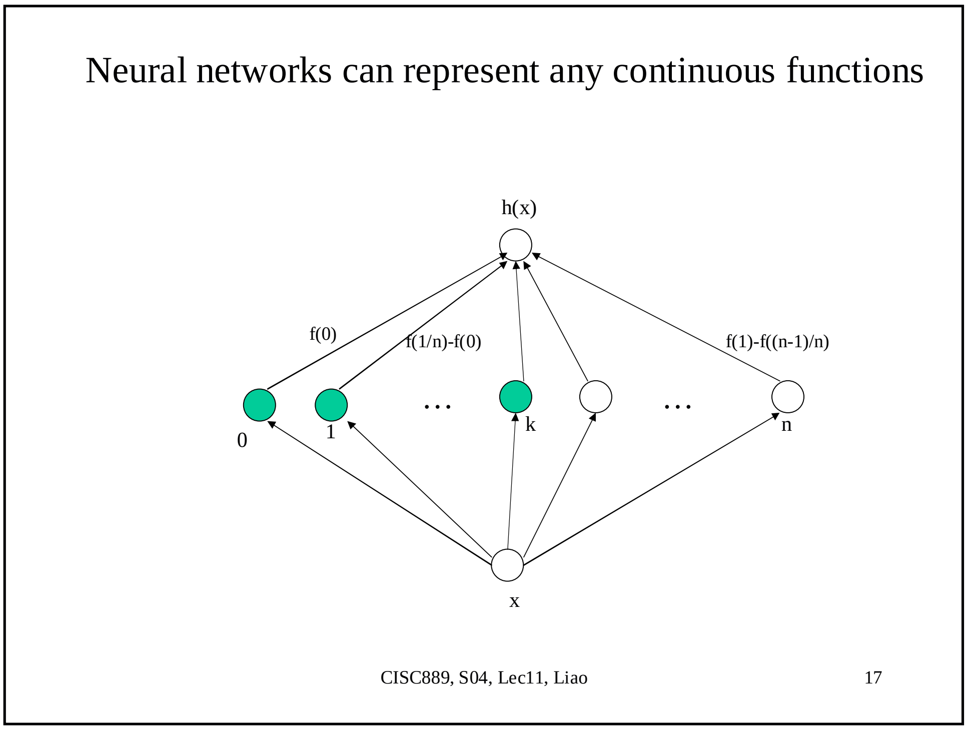
$$|f(x_2) - f(x_1)| \leq e \leftrightarrow |x_2 - x_1| \leq 1/n.$$

- Net:
  - one input unit for x, n+1 hidden nodes, and one output unit for y
  - All weights from input to n hidden units are set to 1
  - K-th hidden unit has threshold value at (k-1)/n.
  - Weight for kth hidden node to output unit is f(k/n) – f((k-1)/n).
  - Output unit is identity function with 0 threshold.
  - For any x, it must fall in an interval [(k-1)/n, k/n] for some k, then only the first k hidden nodes are turned on, therefore

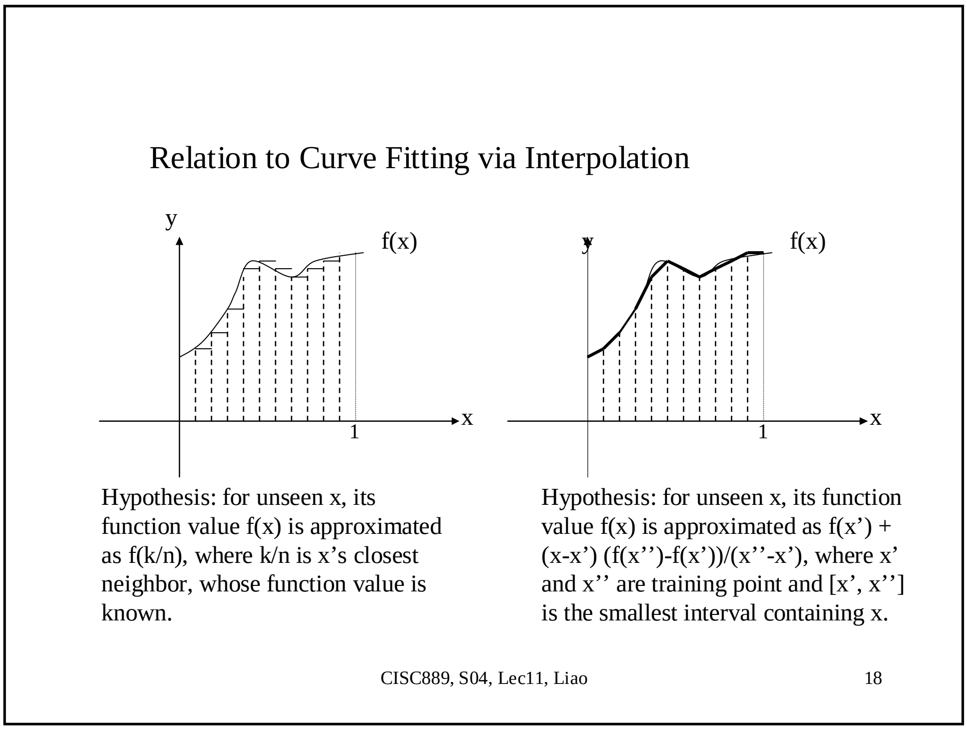
$$h(x) = f(0) + \sum j = 1\ k\ (f(j/n) - f((j-1)/n) = f(k/n)$$

and

$$|f(x) - h(x)| = |f(x) - f(k/n)|$$
$$\leq e$$



Neural networks can represent any continuous functions



Relation to Curve Fitting via Interpolation

Hypothesis: for unseen x, its function value f(x) is approximated as f(k/n), where k/n is x's closest neighbor, whose function value is known.

Hypothesis: for unseen x, its function value f(x) is approximated as f(x') + (x-x') (f(x'')-f(x'))/(x''-x'), where x' and x'' are training point and [x', x''] is the smallest interval containing x.

- Inductive:

  lacking any further information, it is assumed that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data.

  – This should remind you of the maximum likelihood method. We will seen maximum posterior probability approach when discuss bayesian.

  – Bias (rote learner) v.s. generalization

- Learning: to determine weights and thresholds for all nodes (neurons) so that the net can approximate the training data within error range.

  – Back-propagation algorithm

  - Feedforward from Input to output
  - Calculate the error (which is the difference between the network output and the target output):

$$E = (1/2)\ \Sigma_{d \in D} (t_d - o_d)^2,$$

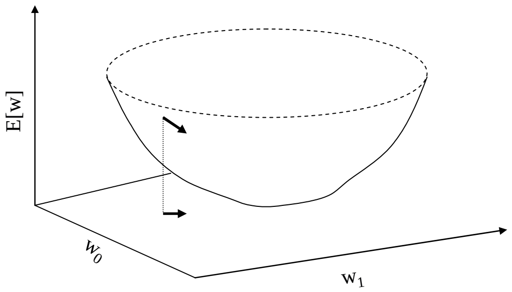
where D stands for set of all training data, and for data d, t_d and o_d are the target output and network output respectively.

  - E is a function of all weights w in the network. Adjust w (by *gradient descent*) to decrease the error. This is done layer-to-layer backwards in the network, called back-propogation.

Gradient descent

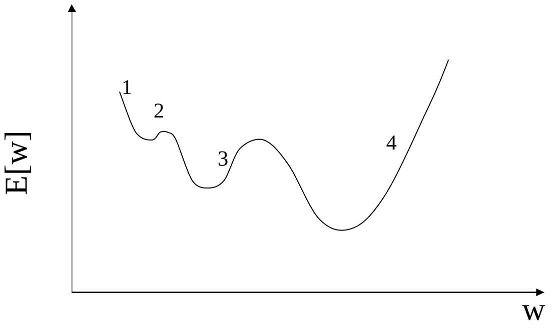
$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} - r\,[\partial E/\partial \mathbf{w}]$

where r is a positive constant called learning rate, which determines the step size for the weights to be altered in the steepest descent direction along the error surface.



- Issues with ANNs
  - Network architecture
    - FeedForward (fully connected vs sparsely connected)
    - Recurrent
    - Number of hidden layers, number of hidden units within a layer
  - Network parameters
    - Learning rate
    - Momentum term
  - Input/output encoding
    - One of the most significant factors for good performance
    - Extract maximal info
    - Similar instances are encoded to “closer” vectors

- Avoid Overfitting (early stop)
  - Data set is split into three parts: training set, validation set, and prediction set.
  - Training continues as long as the performance on the validation set keeps improving, and stops otherwise.
- Avoid local optima
  - Add momentum term
  - Use stochastic gradient descent (e.g., *simulated annealing*)
  - Train multiple networks (initializing each with different random weights)



-Initialize weights randomly to have chance to start from differently locations, e.g., 1, 3 and 4.

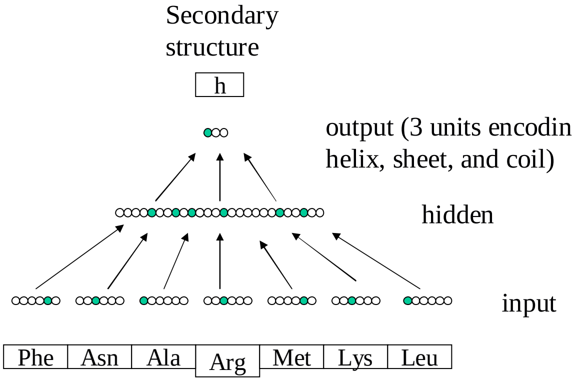
-Add momentum term to help get over little bumps like location 2.

-Simulated annealing: even when a new location will increase ΔE, there is still a chance $e^{-\Delta E / T}$ to take this new location. This is how to avoid being trapped in a local minimum.

Application for sequence analysis

- Input/output encoding
  - Direct sequence encoding
    - BIN4:
      A →1000; T → 0100; G → 0010; C → 0001; - → 0000
    - BIN2:
      A →00; T → 01; G → 10; C → 11
    - For amino acids: each amino acid → a vector of 21 bits (This is called BIN21)
    - Other properties of amino acids, such as hydrophobicity.
  - Indirect sequence encoding
    Sequence features and information content can be extracted by various scoring mechanisms.
    - Residue frequency
  - Input trimming
    Reduce dimensions and condense information content
    - Decision trees
    - Singular value decomposition (SVD)
    - Principle component analysis (PCA)

Qian & Sejnowski, JMB 202(1988)865-884



Sequence of amino acid processed as sliding windows of fixed-length (7 to 17 aa) segments. The central residues are then classified by a three-state (helix, sheet, or coil) prediction.

- Evaluation of performance
  - Success rate Q_3

    $$Q_3 = (P_a + P_b + P_c)/N$$

    Where N is total number of predicted residues and

    P_a , P_b , and P_c are numbers of correctly predicted helix, sheet, and coil respectively.
  - Correlation coefficient

    $$C = TP \bullet TN - FP \bullet FN / \sqrt{(PP \bullet PN \bullet AP \bullet AN)}$$
  - Cross validation
    - In k-fold cross validation, data set is randomly split into two exclusive parts, training and testing, with ratio k to1.

- Performance
  - ceiling at about 65% for direct encoding
    - Local encoding schemes present limited correlation information between residues
    - Little or no improvement using multiple hidden layers.
  - Surpassing 70% by
    - Including evolutionary information (contained in multiple alignment)
    - Using cascaded neural networks
    - Incorporating global information (e.g., position specific conservation weights)

Cathy Wu, Computers Chem. 21(1997)237-256

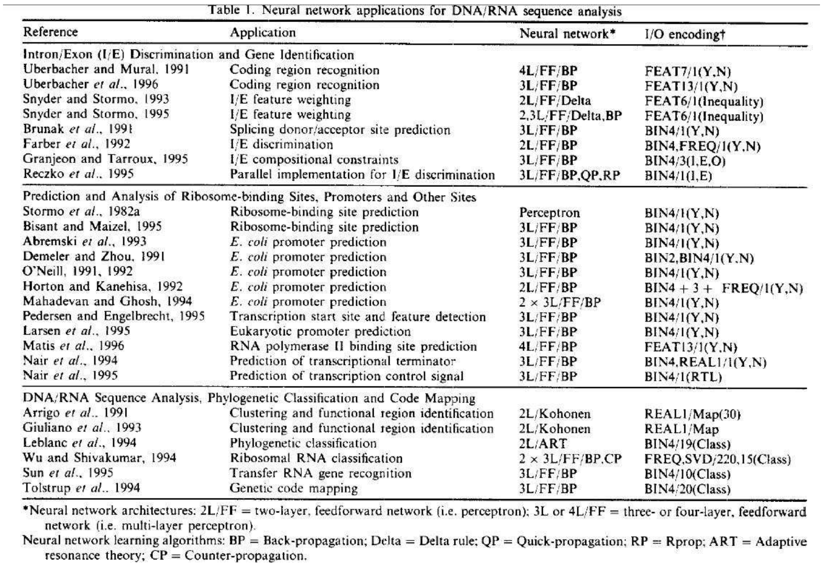


Table 1. Neural network applications for DNA/RNA sequence analysis

Reference	Application	Neural network*	I/O encoding†
Intron/Exon (I/E) Discrimination and Gene Identification			
Uberbacher and Mural, 1991	Coding region recognition	4L/FF/BP	FEAT7/1(Y,N)
Uberbacher *et al*., 1996	Coding region recognition	3L/FF/BP	FEAT13/1(Y,N)
Snyder and Stormo, 1993	I/E feature weighting	2L/FF/Delta	FEAT6/1(Inequality)
Snyder and Stormo, 1995	I/E feature weighting	2,3L/FF/Delta,BP	FEAT6/1(Inequality)
Brunak *et al*., 1991	Splicing donor/acceptor site prediction	3L/FF/BP	BIN4/1(Y,N)
Farber *et al*., 1992	I/E discrimination	2L/FF/BP	BIN4,FREQ/1(Y,N)
Granjeon and Tarroux, 1995	I/E compositional constraints	3L/FF/BP	BIN4/3(I,E,O)
Reczko *et al*., 1995	Parallel implementation for I/E discrimination	3L/FF/BP,QP,RP	BIN4/1(I,E)
Prediction and Analysis of Ribosome-binding Sites, Promoters and Other Sites			
Stormo *et al*., 1982a	Ribosome-binding site prediction	Perceptron	BIN4/1(Y,N)
Bisant and Maizel, 1995	Ribosome-binding site prediction	3L/FF/BP	BIN4/1(Y,N)
Abremski *et al*., 1993	*E. coli* promoter prediction	3L/FF/BP	BIN4/1(Y,N)
Demeler and Zhou, 1991	*E. coli* promoter prediction	3L/FF/BP	BIN2,BIN4/1(Y,N)
O'Neill, 1991, 1992	*E. coli* promoter prediction	3L/FF/BP	BIN4/1(Y,N)
Horton and Kanehisa, 1992	*E. coli* promoter prediction	2L/FF/BP	BIN4 + 3 + FREQ/1(Y,N)
Mahadevan and Ghosh, 1994	*E. coli* promoter prediction	2 × 3L/FF/BP	BIN4/1(Y,N)
Pedersen and Engelbrecht, 1995	Transcription start site and feature detection	3L/FF/BP	BIN4/1(Y,N)
Larsen *et al*., 1995	Eukaryotic promoter prediction	3L/FF/BP	BIN4/1(Y,N)
Matis *et al*., 1996	RNA polymerase II binding site prediction	4L/FF/BP	FEAT13/1(Y,N)
Nair *et al*., 1994	Prediction of transcriptional terminator	3L/FF/BP	BIN4,REAL1/1(Y,N)
Nair *et al*., 1995	Prediction of transcription control signal	3L/FF/BP	BIN4/1(RTL)
DNA/RNA Sequence Analysis, Phylogenetic Classification and Code Mapping			
Arrigo *et al*., 1991	Clustering and functional region identification	2L/Kohonen	REAL1/Map(30)
Giuliano *et al*., 1993	Clustering and functional region identification	2L/Kohonen	REAL1/Map
Leblanc *et al*., 1994	Phylogenetic classification	2L/ART	BIN4/19(Class)
Wu and Shivakumar, 1994	Ribosomal RNA classification	2 × 3L/FF/BP,CP	FREQ,SVD/220,15(Class)
Sun *et al*., 1995	Transfer RNA gene recognition	3L/FF/BP	BIN4/10(Class)
Tolstrup *et al*., 1994	Genetic code mapping	3L/FF/BP	BIN4/20(Class)

*Neural network architectures: 2L/FF = two-layer, feedforward network (i.e. perceptron); 3L or 4L/FF = three- or four-layer, feedforward network (i.e. multi-layer perceptron).

Neural network learning algorithms: BP = Back-propagation; Delta = Delta rule; QP = Quick-propagation; RP = Rprop; ART = Adaptive resonance theory; CP = Counter-propagation.

- http://saturn.med.nyu.edu/searching/SSpred/queryss.html
- http://www.cmpharm.ucsf.edu/~nomi/nnpredict.html

Secondary Structure Prediction Results

Top line is your sequence. Second line is the Sstructure prediction. Third line is the pre-residue confidence of the prediction (0=low confidence, 9=high confidence)

```
MLMPKKNRIAIYELLFKEGVMVAKKDVHMPKHPELADKNVPNLHVMKAM
________HHHHHHHHH___EEEE___________________HHHHHH
0089007099979990069566755998099099999899988570900
SLKSRGYVKEQFAWRHFYWYLTNEGIQYLRDYLHLPPEIVPATLRRSRP
HHH___HHHHHHHHHHHHH______HHHHHHH_______HHHHHH____
0966066999996677880057097077050050000006666766009
TGRPRPKGPEGERPARFTRGEADRDTYRRSAVPPGADKKAEAGAGSATE
____________________________________HHHHH________
9000000000000760078880070560070909000006777909865
QFRGGFGRGRGQPPQ _______________ 000699999990000
```