Artificial Neural Networks and Application to Thunderstorm Prediction

Waylon Collins
NOAA/National Weather Service
Philippe Tissot
Texas A&M University – Corpus Christi

Presentation Outline

I. Artificial Intelligence (AI) & Artificial Neural Networks (ANN)

II. Project #1: Thunderstorm Artificial Neural Network (TANN)

(a) Motivation

(b) Design/Framework

(c) Model Development/Optimization Strategy

(d) Most recent Results/Conclusions

(e) Operational TANN

III. Project #2: Improving TANN in connection with a NASA-funded project *Improved Convective Initiation Forecasting in the Gulf of Mexico Region*

Artificial Intelligence (AI) and Artificial Neural Networks (ANN)

The ANN is a form of Artificial Intelligence (AI)

What is Intelligence?

Intelligent systems tend to possess one or more of the following:

1. Sensory perception
2. Pattern Recognition
3. Learning and knowledge acquisition
4. Inference from incomplete information
5. Ability to deal with unfamiliar situations
6. Adaptability to new, yet related situations (through expectational knowledge)
7. Inductive Reasoning
8. Common Sense
9. Display of Emotions
10. Inventiveness

de Silva, C. W., 2000: *Intelligent Machines: Myths and Realities*. CRC Press LLC

Artificial Intelligence (AI) and Artificial Neural Networks (ANN)

The ANN is a form of Artificial Intelligence (AI)

What is AI?

1. **"The collective attributes of a computer, robot, or other mechanical device programmed to perform functions analogous to learning and decision making"**

Costello, R. B., 1992: *Random House Webster's College Dictionary*. New York: Random House, Inc.

2. **"Multidisciplinary field encompassing computer science, neuroscience, philosophy, psychology, robotics, and linguistics, and devoted to the reproduction of the methods or results of human reasoning and brain activity"**

Glickman, T. S., 2000: *Glossary of Meteorology*. 2nd Edition, American Meteorological Society, Boston. 855p

Artificial Intelligence (AI) and Artificial Neural Networks (ANN)

The ANN is a form of Artificial Intelligence (AI)

What is AI?

3. “Artificial intelligence is the science of making machines do things that would require intelligence if done by men”

de Silva, C. W., 2000: *Intelligent Machines: Myths and Realities*. CRC Press LLC

Artificial Neural Network (ANN)

"…A **neural network** is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge…"

Haykin, S., 1999: *Neural Networks: A Comprehensive Foundation, 2nd Edition*, Prentice-Hall, Inc.

Haykin adapted the foregoing ANN definition from Aleksander and Morton (1990) who viewed the ANN as an adaptive machine.

Aleksander, I., and H. Morton, 1990: *An Introduction to Neural Computing*, London: Chapman and Hall.

Artificial Neural Network (ANN)

What is Parallel Distributed Processing (PDP)?

"…information processing takes place through interactions of large numbers of simple processing elements called units, each sending excitatory and inhibitory signals to other units…"

Rumelhart, D. E., J. L. McClelland, and the PDP Research Group, 1986: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press: Cambridge, MA.

Artificial Neural Network (ANN)

What is Parallel Distributed Processing (PDP)?

PDP Model General Framework

1. A set of processing units
2. A state of activation
3. An output function for each unit
4. A pattern of connectivity among units
5. A propagation rule for propagating patterns of activities through the network of connectivities
6. An activation rule for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit
7. A learning rule whereby patterns of connectivity are modified by experience
8. An environment within which the system must operate

Rumelhart, D. E., J. L. McClelland, and the PDP Research Group, 1986: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press: Cambridge, MA.

Artificial Neural Network (ANN)

Parallel Distributed Processing in Biological Neural Networks

- Densely interconnected network of $\sim 10^{11}$ neurons, each connected, on average, to 10^4 others.
- The fastest neuron switching times: $\sim 10^{-3}$ seconds; computer switching speeds: $\sim 10^{-10}$ seconds.
- Humans → render complex decisions, surprisingly quickly. (e.g. $\sim 10^{-1}$ seconds required to visually recognize your mother.)
- Speculation based on foregoing → information-processing abilities of biological neural systems must follow from highly parallel processes operating on representations that are distributed over many neurons.
- A motivation for ANN systems → capture this kind of highly parallel computation based on distributed representations

Mitchell, T. M, 1997: *Machine Learning*, McGraw Hill Companies, Inc. (Section 4.1.1; page 82)

Artificial Neural Network (ANN)

Parallel distributed processing versus serial processing

Parallel distributed processing most efficient to resolve the following:

Serial processing most efficient to resolve the following:

$$\sum_{X=1}^{N} X^4$$

Beal R. and T. Jackson, 1990: Neural Computing: An Introduction, Institute of Physics Publishing.

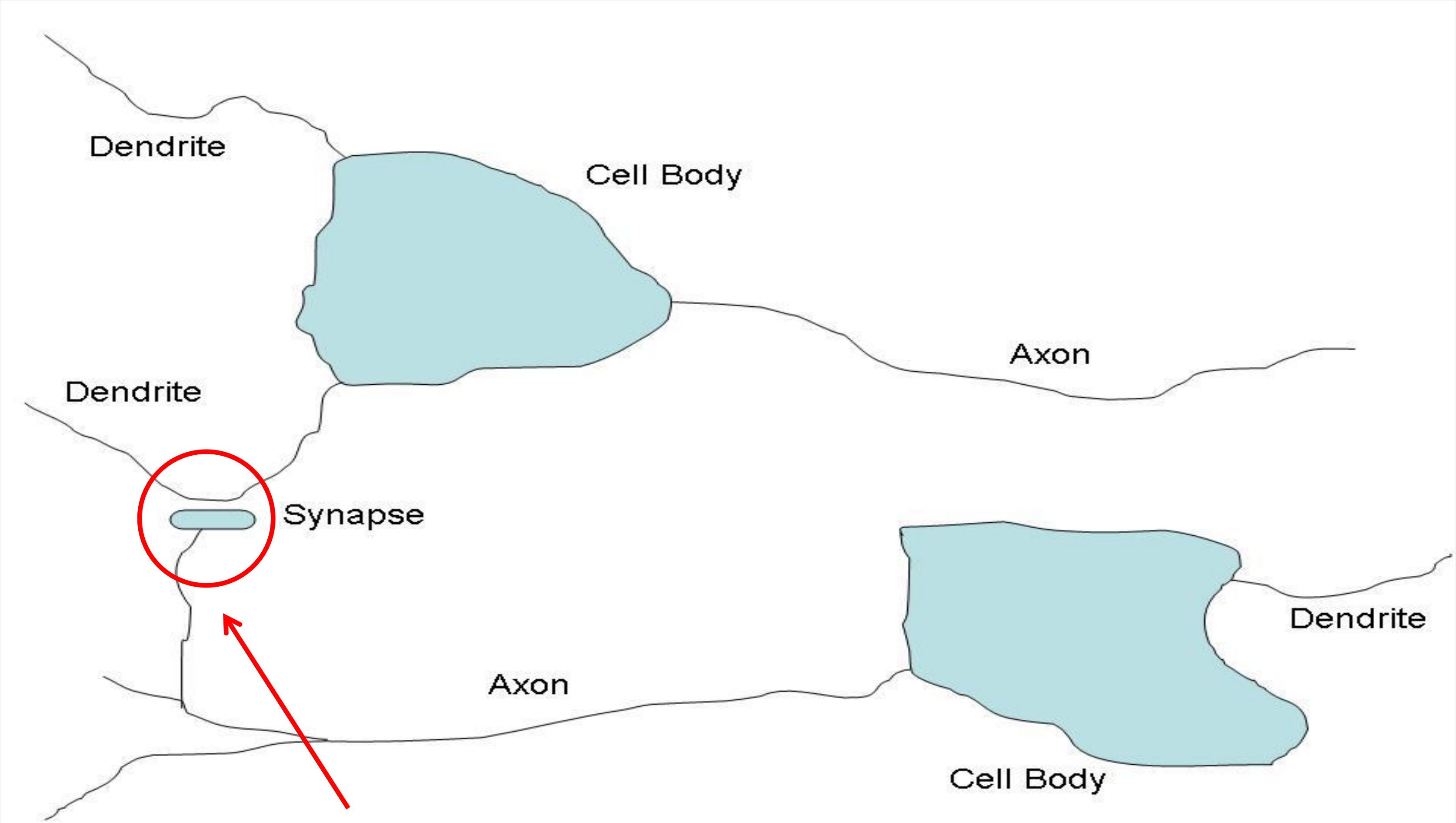
Artificial Neural Network (ANN)

Serial Processing Example: $\sum_{x=1}^{N} x^4$

```
cat speed-of-serial.ksh
#!/bin/ksh
# Initialize x
x=0
N=1000000
# Compute the sum of x after N iterations
time for ((i = 1; i <= $N; i += 1))
do
x=$((i**4))
done
# The answer
echo The sum of the 4th power of integers
1 through $N is $((x+1))
echo $?

./speed-of-serial.ksh

real    0m6.07s
user    0m6.04s
sys     0m0.02s
The sum of the 4th power of integers 1
through 1000000 is 9.99999999999999983e+23
0
```

Artificial Neural Network (ANN)

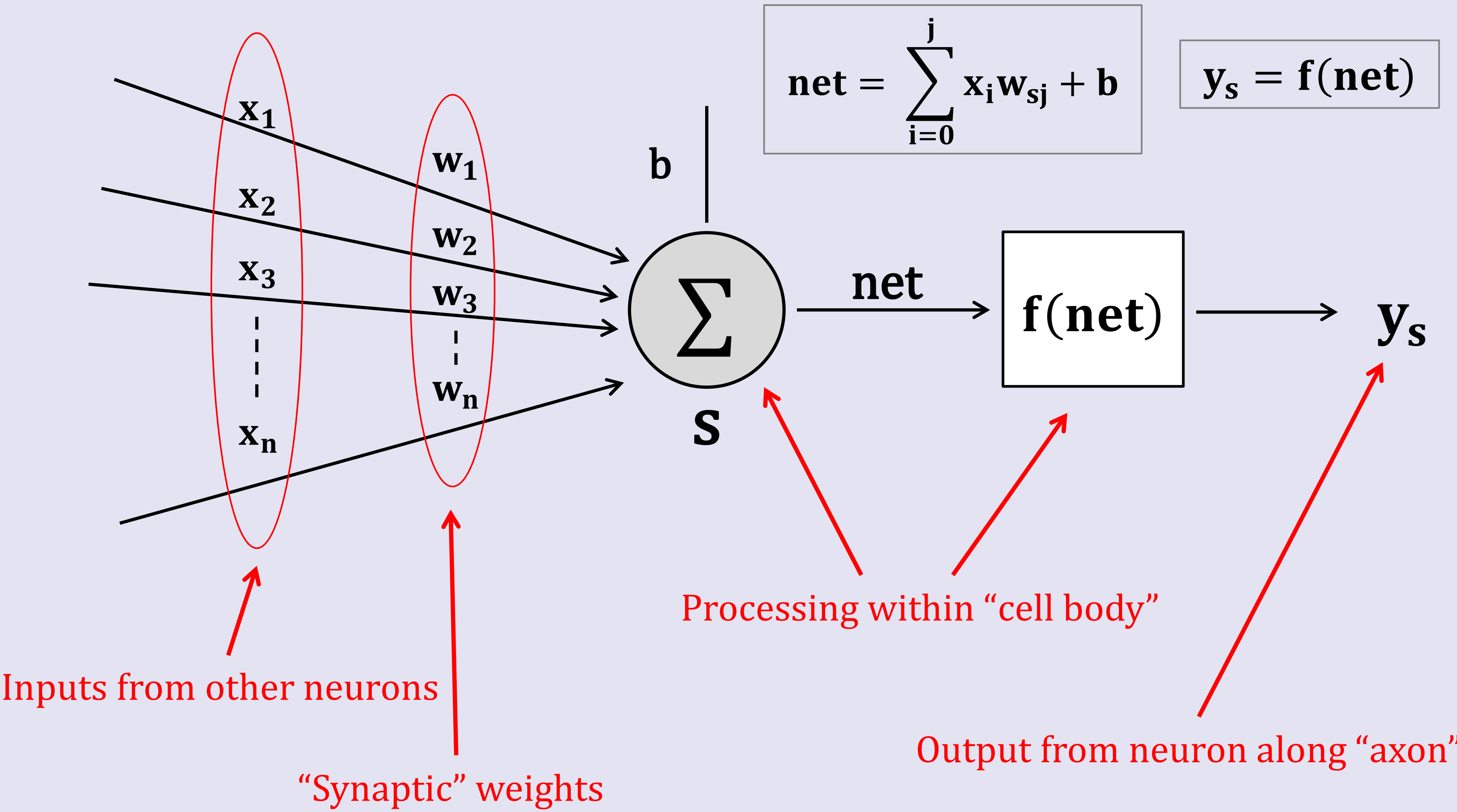
ANN versus the Biological Neural Network: **Biological Neuron Model**



Adjust synaptic weights → "Learning" (e.g. Hagan et. al. 1996)

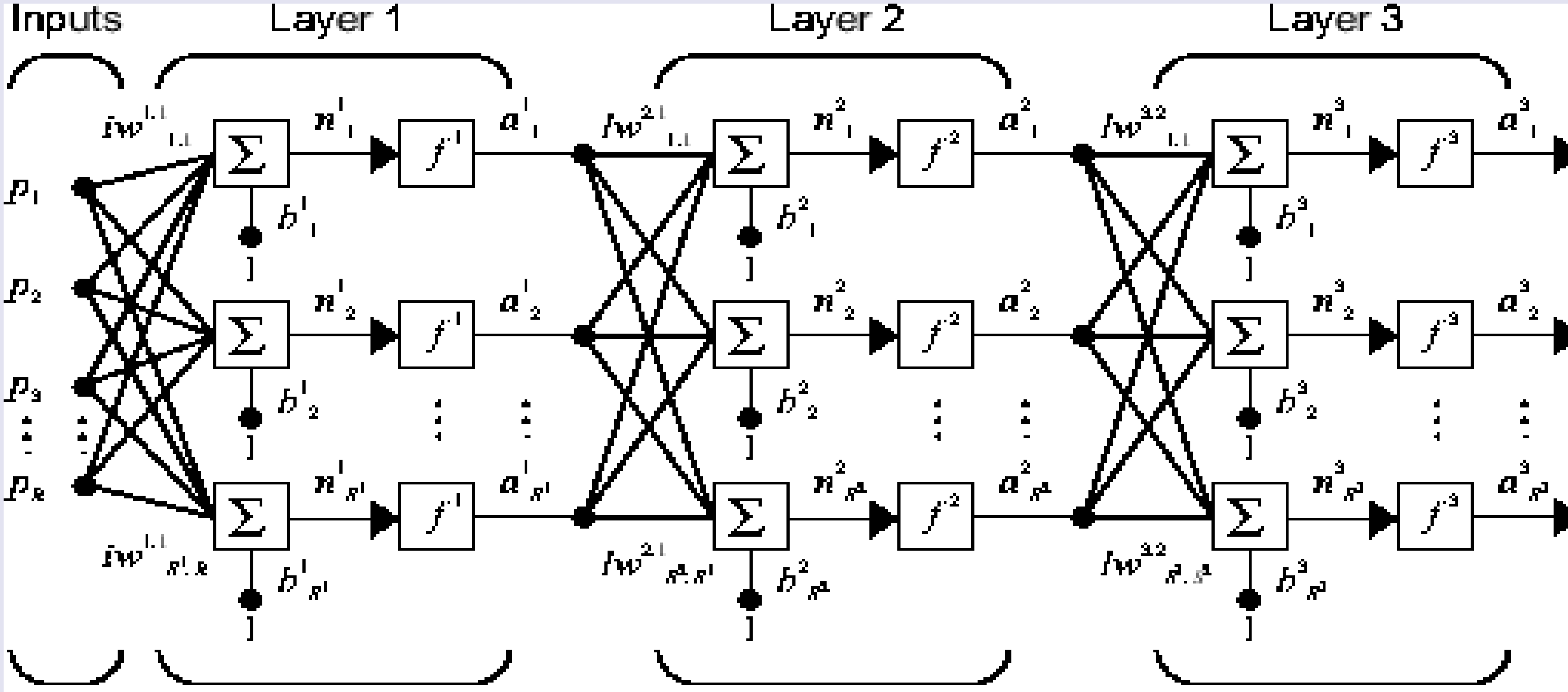
Artificial Neural Network (ANN)

ANN versus the Biological Neural Network: **Single Artificial Neuron**



Artificial Neural Network (ANN)

ANN versus the Biological Neural Network: **3- Layer Feed-forward Multilayer Perceptron (MLP) Artificial Neural Network**



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Source http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf

Artificial Neural Network (ANN)

Feed-forward MLP ANN: A Universal Approximator

1. Statistical techniques such as non-linear regression and discriminate analysis are parametric
2. Non-linear regression requires assumptions regarding form of functional relationship between predictor/predictand
3. Discriminate analysis assumes that probability density function of the predictors is Gaussian
4. ANNs are non-parametric; they don't require assumptions regarding functional relationship between predictor/predictand
5. The optimal functional relationship can be obtained after adjusting the number and size of hidden layers and associated neurons
6. An ANN with 1-hidden layer, with sigmoidal units in the hidden layer, can approximate any continuous function with a finite number of discontinuities

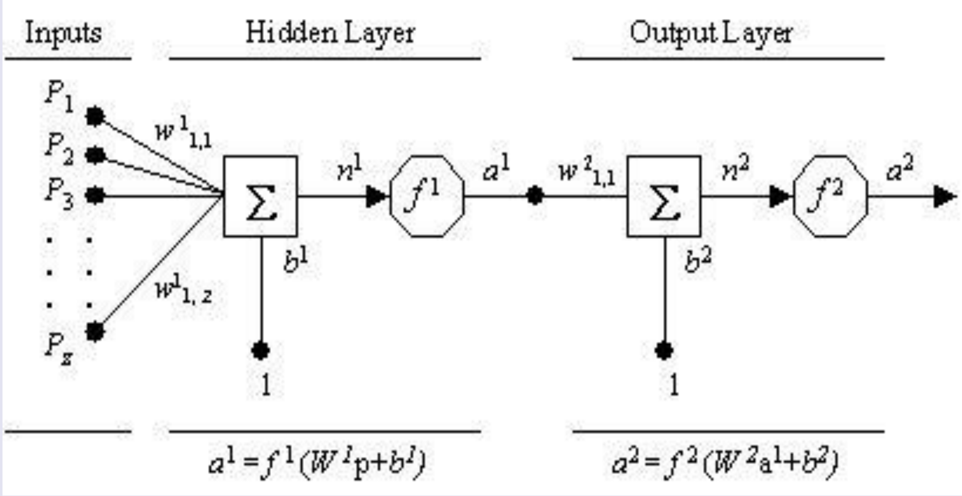
Hornik, K., M. Stinchcombe, and H. White, 1989: Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2, pp. 359-366.

Hornik, K. M., Stinchcombe, and H. White, 1990: Universal approximation of an unknown mapping and its derivatvies using multilayer feedforward networks. *Neural Networks*, 3, pp 551-560

Artificial Neural Network (ANN)

Characterization (de Silva, 2000)

- **Topology**: Manner in which interconnections between neurons are arranged and how data flows through the system (e.g. Feed-forward Multilayer Perceptron)
- **Transfer Function**: Algebraic function to transfer information across network
- **Learning Rule**: Algorithm to update weights and biases connecting neurons (train the ANN to perform a task) (e.g. Back-propagation)

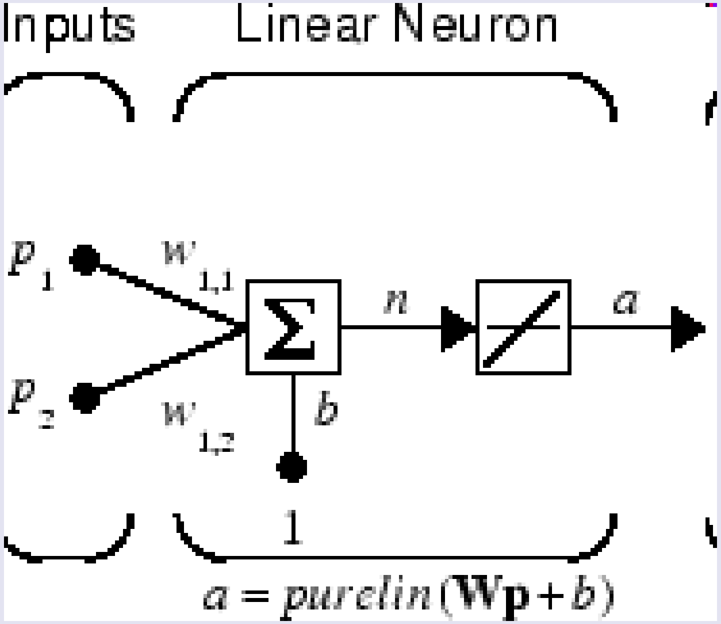


de Silva, C. W., 2000: *Intelligent Machines: Myths and Realities*. CRC Press LLC

Image adapted from: Hagan, M. T., H. B. Demuth, and M. Beale, 1996: *Neural Network Design*, International Thomson Publishing Inc.

Artificial Neural Network (ANN)

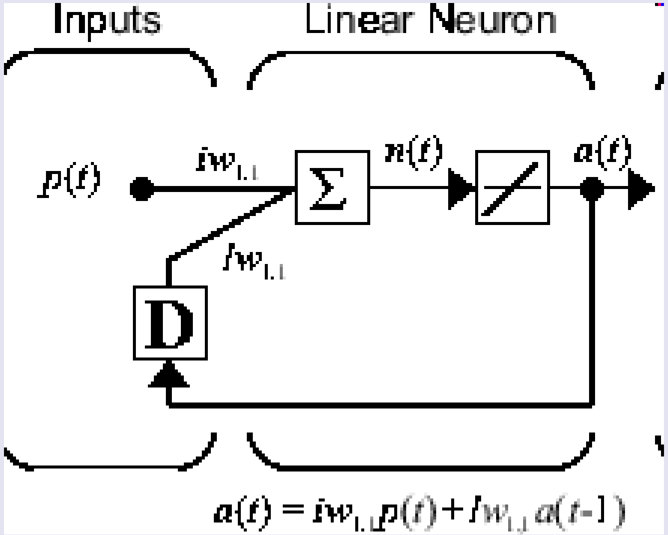
Feedforward Topology: Neurons connected in forward/unidirectional path starting from input to each layer and finally to the output layer.



URL http://www.mathworks.com/help/toolbox/nnet/

Artificial Neural Network (ANN)

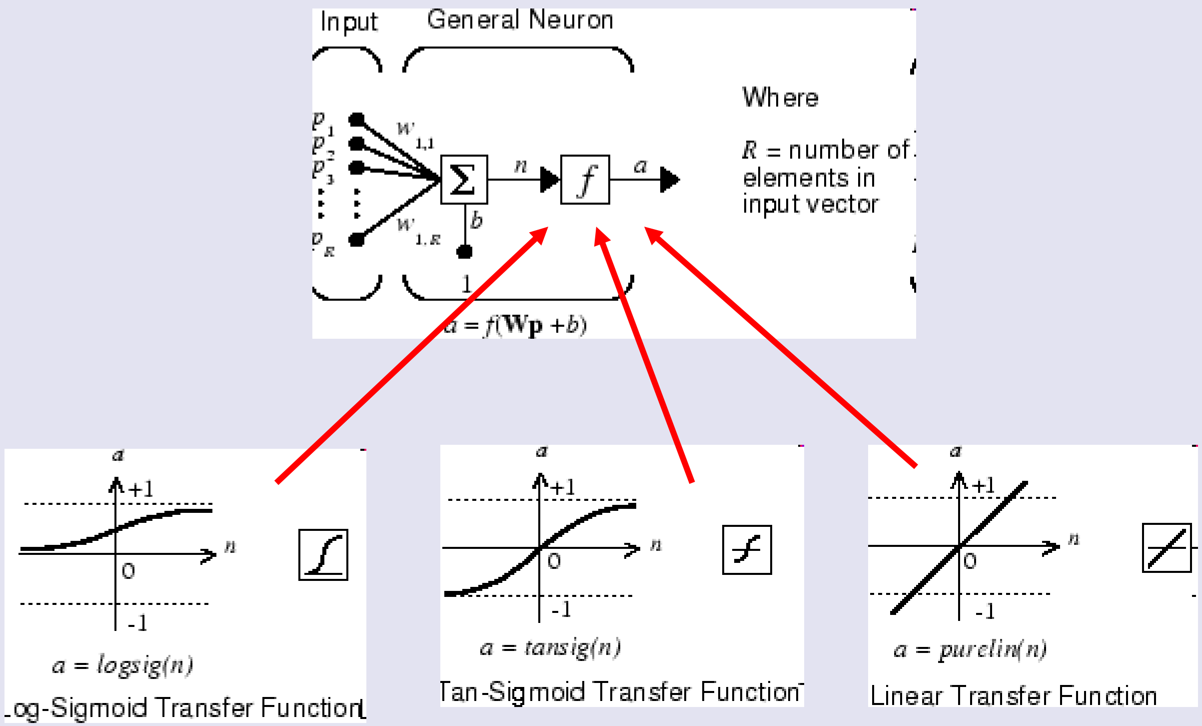
Recurrent Topology: Allow for feedback connections amongst various neurons.



URL http://www.mathworks.com/help/toolbox/nnet/

Artificial Neural Network (ANN)

Transfer Functions: Input weighted sum of inputs from other neurons and apply a nonlinear mapping



URL http://www.mathworks.com/help/toolbox/nnet/

Artificial Neural Network (ANN)

Learning Rule Categories

- Supervised Learning

1. Provide input (x)/correct output (t) to the network $\{x_1,t_1\}$,$\{x_2,t_2\}$..$\{x_N,t_N\}$
2. Adjust connection weights to minimize error between target and ANN output

- Unsupervised (Self-Organized) Learning

1. Input data presented to system
2. Connection weights adjusted based on network inputs
3. Similar to clustering

- Reinforcement Learning

1. Similar to supervised learning, yet instead of target given, a grade is assigned based on ANN performance for a given set of inputs

de Silva, C. W., 2000: *Intelligent Machines: Myths and Realities*. CRC Press LLC

Image adapted from: Hagan, M. T., H. B. Demuth, and M. Beale, 1996: *Neural Network Design*, International Thomson Publishing Inc.

Artificial Neural Network (ANN)

Basic Learning Rules

- **Error-Correction Learning**

Widrow-Hoff (Delta) Rule → minimize cost function $\xi = \frac{1}{2} {e_k}^2$, where $e_k = t_k - y_k$

- **Memory-based Learning**

Explicit memorization of training data

- **Hebbian Learning**

Hebbian synapse → "uses a time-dependent… mechanism to increase synaptic efficiency as a function of the correlation between presynaptic and postsynaptic activities "

- **Competitive Learning**

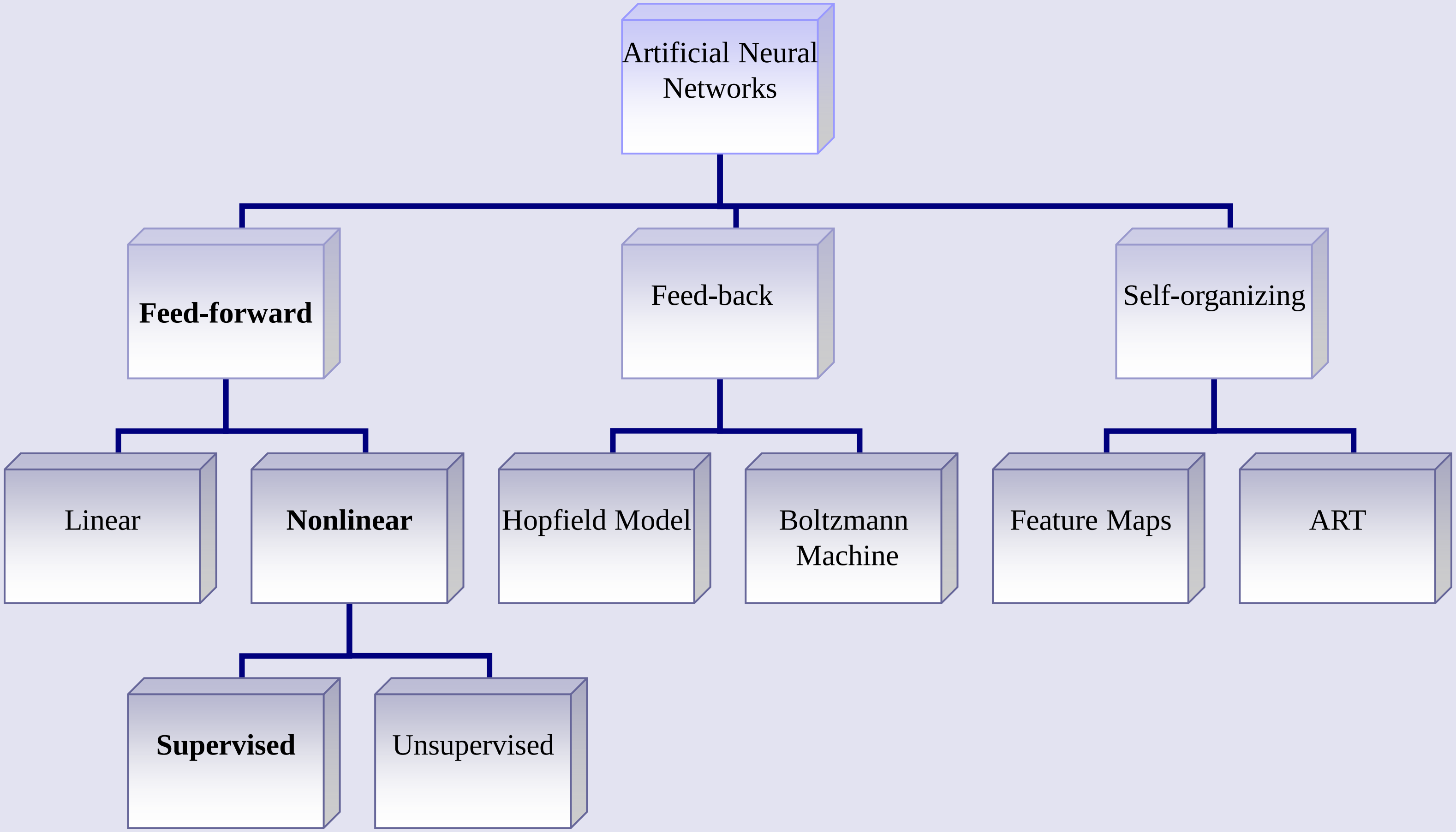
Competition to determine which neuron becomes active

- **Boltzmann Learning**

Stochastic learning based on statistical mechanics

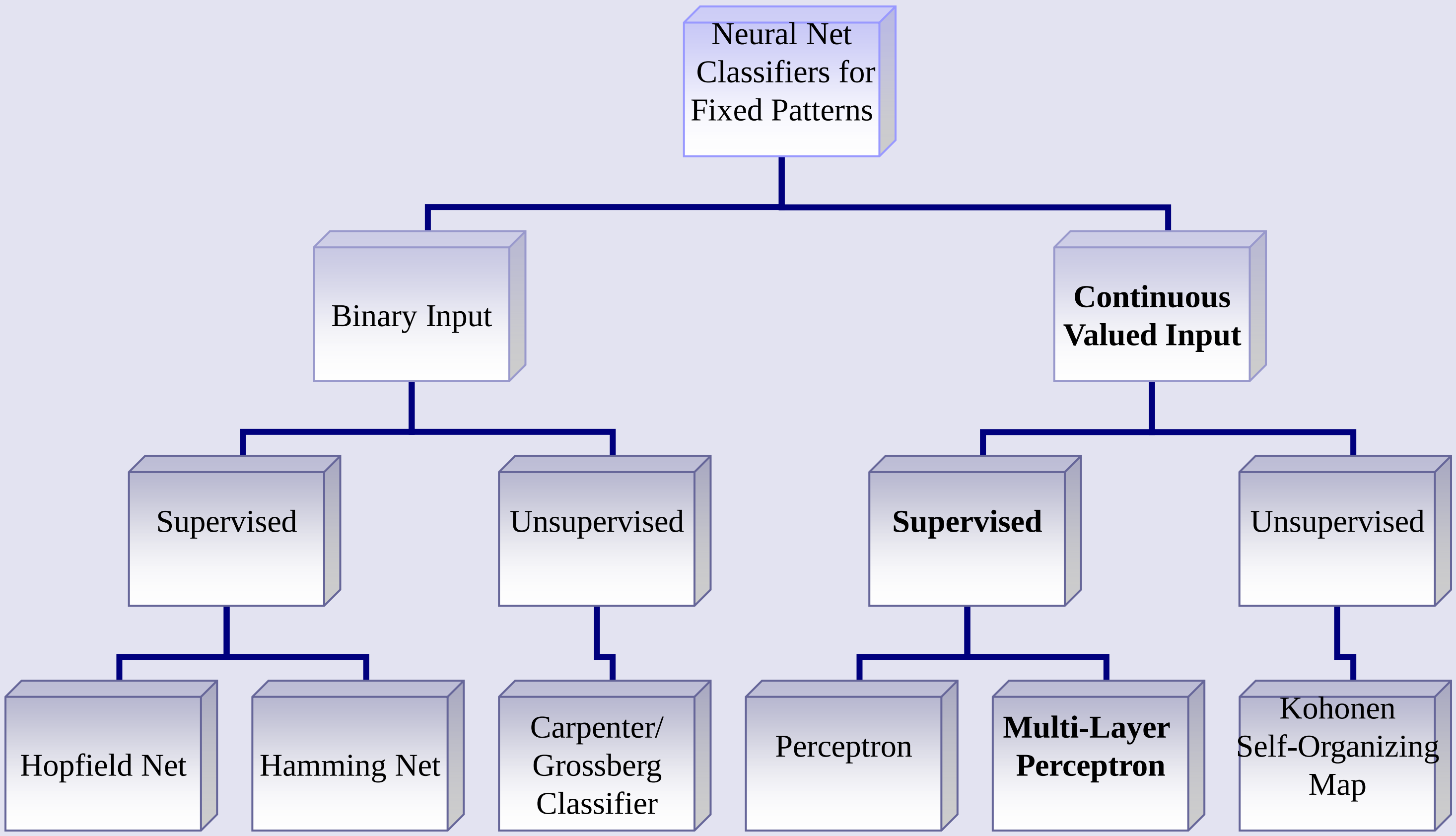
Haykin, S., 1999: *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall.

Artificial Neural Network (ANN)—Taxonomy with respect to ANN Topology



Karayiannis, N. B., 1993: *Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications*. Kluwer Academic Publishers, Boston

Artificial Neural Network (ANN)—Taxonomy with respect to Input Data Type and ANN Learning Rule Category



Lippmann, R. 1987: *An Introduction to Computing with Neural Nets,* IEEE ASSP Magazine. pp 4-22.

Artificial Neural Network (ANN)—Back-propagation Learning/ Training Algorithm

$$E_p = \frac{1}{2}\sum_j \left(t_{pj} - o_{pj}\right)^2 \quad (1)$$

$$net_{pj} = \sum_i w_{ij}\, o_{pi} \quad (2)$$

$$o_{pj} = f_j\left(net_{pj}\right) \quad (3)$$

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial net_{pj}}\left(\frac{\partial net_{pj}}{\partial w_{ij}}\right) \quad (4)$$

$$\frac{\partial net_{pj}}{\partial w_{ij}} = \frac{\partial\left(\sum_i w_{ij}\, o_{pi}\right)}{\partial w_{ij}} = o_{pi} \quad (5)$$

$$Define \quad -\frac{\partial E_p}{\partial net_{pj}} = \delta_{pj} \quad (6)$$

$$Thus, \quad \frac{\partial E_p}{\partial w_{ij}} = -\delta_{pj}\, o_{pi} \quad (7)$$

$Interpretation\ of\ (7): \downarrow E_p\ suggests\ making\ weight\ changes\ \propto to\ \delta_{pi}\, o_{pi}:$

$$\Delta_p w_{ij} = \eta \delta_{pj}\, o_{pi} \quad (8)$$

Using (7) and the chain rule:

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}}\frac{\partial o_{pj}}{\partial net_{pj}} \quad (9)$$

$$\frac{\partial o_{pj}}{\partial net_{pj}} = \dot{f}_j\left(net_{pj}\right) \quad (10)$$

$$\frac{\partial E_p}{\partial o_{pj}} = \frac{\partial\left(\frac{1}{2}\sum_j\left(t_{pj} - o_{pj}\right)^2\right)}{\partial o_{pj}} = \left(t_{pj} - o_{pj}\right)(-1) \quad (11)$$

$$\boldsymbol{\delta_{pj} = -\dot{f}_j\left(net_{pj}\right)\left(t_{pj} - o_{pj}\right) \quad (12)}$$

*Will use (12) to compute δ_{pj}, the error signal for pattern p on neuron j (**output layer**)*

in backpropagation algorithm

Beal R. and T. Jackson, 1990: Neural Computing: An Introduction, Institute of Physics Publishing.

Artificial Neural Network (ANN)—Back-propagation Learning/ Training Algorithm

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} \quad (13)$$

$$\frac{\partial E_p}{\partial net_{pk}} = \delta_{pk} \quad (14)$$

$$\frac{\partial net_{pk}}{\partial o_{pj}} = \frac{\partial\left(\sum_i w_{ik}\, o_{pi}\right)}{\partial o_{pj}} = w_{jk} \quad (15)$$

$$\frac{\partial E_p}{\partial o_{pj}} = -\sum_k \delta_{pk}\, w_{jk} \quad (16)$$

$$\frac{\partial o_{pj}}{\partial net_{pj}} = \frac{\partial\left(f_j\left(net_{pj}\right)\right)}{\partial net_{pj}} = \dot{f}_j\left(net_{pj}\right) \ (17)$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \ (18), also\ (9)$$

$$\boldsymbol{\delta_{pj} = \dot{f}_j\left(net_{pj}\right) \sum_k \delta_{pk} w_{jk} \ (19)}$$

*Will use (19) to compute δ_{pj} , the error signal for pattern p on neuron j (**hidden layer**)*

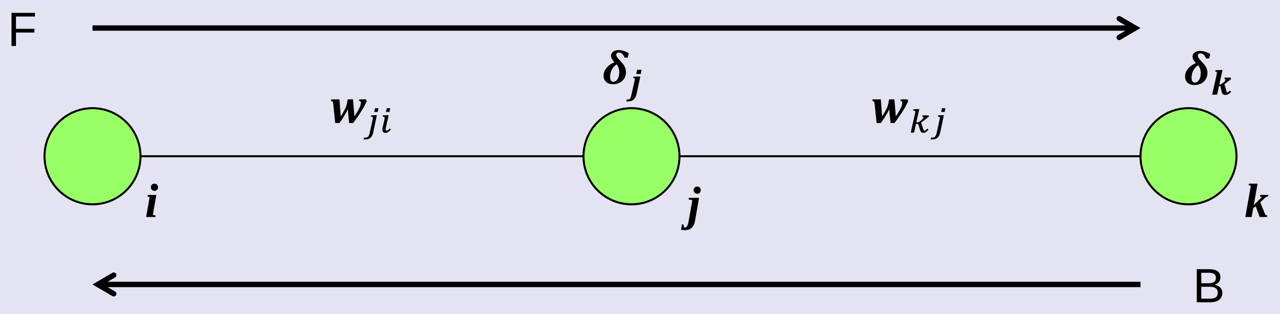
in backpropagation algorithm

Beal R. and T. Jackson, 1990: Neural Computing: An Introduction, Institute of Physics Publishing.

Artificial Neural Network (ANN)—Back-propagation Learning/ Training Algorithm

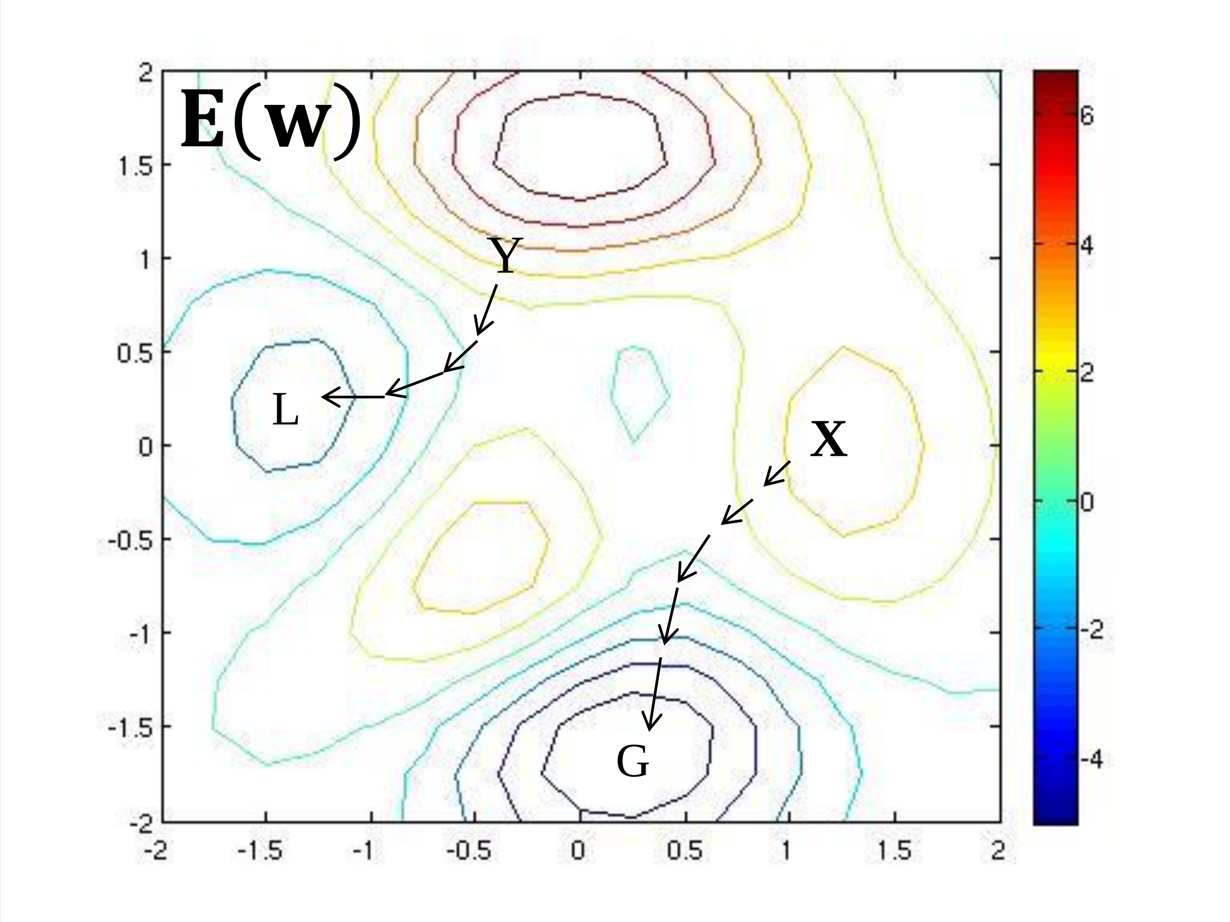
Back-propagation training algorithm/learning rule with respect to a multilayer perceptron (MLP)

1. Apply input vector $\boldsymbol{X}^p$ to the network and perform forward propagation through the network via $\boldsymbol{net_{pj}} = \sum_i \boldsymbol{w_{ij} o_{pi}}$ $(\mathbf{2})$ and $\boldsymbol{o_{pj}} = \boldsymbol{f_j}(\boldsymbol{net_{pj}})$ $(\mathbf{3})$ which obtains activations of all hidden and output neurons
2. Compute $\boldsymbol{\delta_k}$ for all **output neurons** using $\boldsymbol{\delta_{pk}} = -\dot{\boldsymbol{f}}_{\boldsymbol{j}}(\boldsymbol{net_{pk}})(\boldsymbol{t_{pk}} - \boldsymbol{o_{pk}})$ $(\mathbf{12})$
3. Back-propagate all $\boldsymbol{\delta_k}$ via $\boldsymbol{\delta_{pj}} = \dot{\boldsymbol{f}}_{\boldsymbol{j}}(\boldsymbol{net_{pj}}) \sum_k \boldsymbol{\delta_{pk} w_{jk}}$ $(\mathbf{19})$ in order to obtain $\boldsymbol{\delta_j}$ for each **hidden neuron** within the network
4. Update weights $\boldsymbol{w_{ij}}$ on input layer via $\boldsymbol{\Delta_p w_{ij}} = \boldsymbol{\eta \delta_{pj} o_{pi}}$ $(\mathbf{8})$
5. Continue until convergence



Bishop, C. M, 1995: Neural Networks for Pattern Recognition, Oxford University Press.

Artificial Neural Network (ANN)—Back-propagation Learning/ Training Algorithm: Problem with Back-propagation



$\mathbf{w}_2$

$\mathbf{w}_1$

Artificial Neural Network (ANN) Variations on Back-propagation: A *Solution to Back-propagation Problem*

Heuristic Back-propagation Modifications

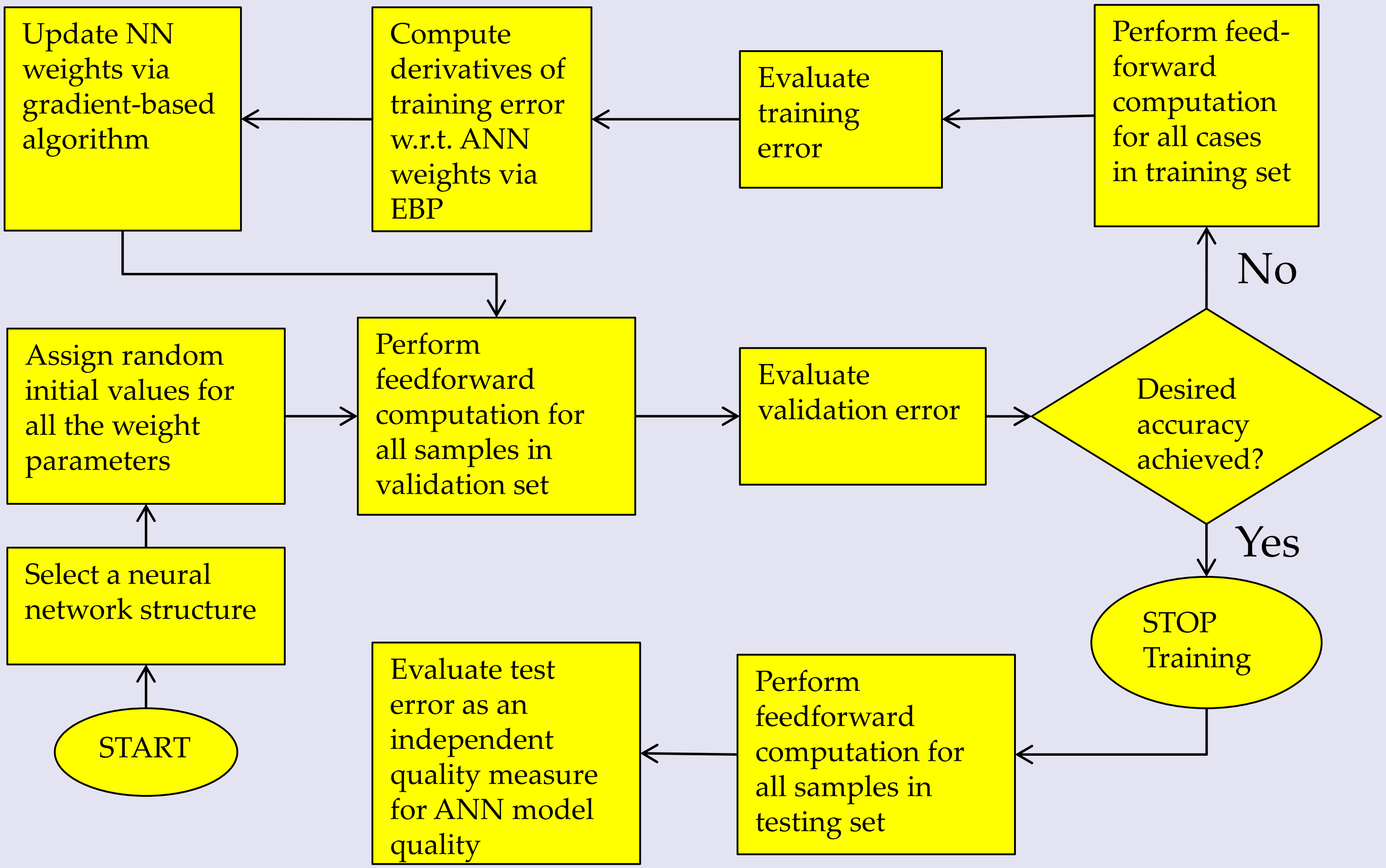
1. Momentum
2. Variable Learning Rates

Numerical Optimization Techniques

1. Conjugate Gradient
2. Levenberg-Marquardt Algorithm

Hagan, M. T., H. B. Demuth, and M. Beale, 1996: *Neural Network Design*, International Thomson Publishing Inc.

Artificial Neural Network (ANN): Model Development Process



Zhang, Q.J. K.C. Gupta, and V.K. Devabhaktuni, 2003: Artificial neural networks for RF and microwave design: from theory to practice, *IEEE Trans. Microwave Theory Tech.*, 51, pp. 1339-1350

Artificial Neural Network (ANN): Model Development Process

1. Formulate the problem and identify input/output parameters
2. Choose an architecture (topology, transfer function, training algorithm)
3. Data processing (organize data to train/validate/test; preprocess data)
4. Train/validate the model on the training/validation data sets
5. Assess model performance/skill on novel data (testing set)
6. If model performs well on both training and testing sets, the model is said to *generalize* well
7. If model performs well on training set yet subpar performance on testing set, *overfitting* is likely occurring.
8. Minimize overfitting via validation during training, use of selected learning algorithms, etc.

Zhang, Q.J. K.C. Gupta, and V.K. Devabhaktuni, 2003: Artificial neural networks for RF and microwave design: from theory to practice, *IEEE Trans. Microwave Theory Tech.*, 51, pp. 1339-1350

Artificial Neural Network (ANN): Applications/Learning Tasks

Pattern Association[1] (Associative Memory)

Autoassociation → store patterns and recall correct pattern given incomplete version of pattern (unsupervised)

Heteroassociation → paired input/output patterns (supervised)

Pattern Recognition[1] (Input pattern/signal → Class/Category)

Input →Feature Extraction (unsupervised) → Classification (supervised)

Input → Feature Extraction (supervised MLP) → Classification

Function Approximation[1] (Approximate nonlinear input-output mapping)

Control (Maintain a system/process in a controlled condition)

Feedback control system

Filtering[1] (Extract information about quantity of interest from noisy data)

Beamforming[1] (Spatial filtering; distinguish between signal and noise)

Haykin, S., 1999: *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall.

Artificial Neural Network (ANN): Selected Application Examples

- **Aerospace** (aircraft component fault detectors)
- **Automotive** (warranty activity analyzers)
- **Banking** (credit application evaluation)
- **Defense** (facial recognition)
- **Electronics** (process control)
- **Entertainment** (market forecasting)
- **Financial** (currency price prediction)
- **Insurance** (policy application evaluation)
- **Manufacturing** (paper quality prediction)
- **Medical** (EEG and ECG analysis)
- **Oil and Gas** (Exploration)
- **Robotics** (vision systems)
- **Speech** (speech recognition)
- **Securities** (automatic bond rating)
- **Telecommunications** (real-time spoken language translation)
- **Transportation** (vehicle scheduling)

Hagan, M. T., H. B. Demuth, and M. Beale, **1996:** Neural Network Design, International Thomson Publishing Inc.

Artificial Neural Network (ANN): Assortment of Recent (1999-2012) Applications

- **Efficiency of atmospheric model physics parameterizations**

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov, 2005: New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Longwave Radiation in a Climate Model, *Monthly Weather Review*, **133**, 1370-

- **Efficiency of Internet Search Engines**

S. Bo, and S. Kak, 1999: A neural network-based intelligent metasearch engine, *Information Sciences*, **120**, 1-11.

- **Efficient Clustering of World Wide Web Documents in accordance with inquirer's s needs**

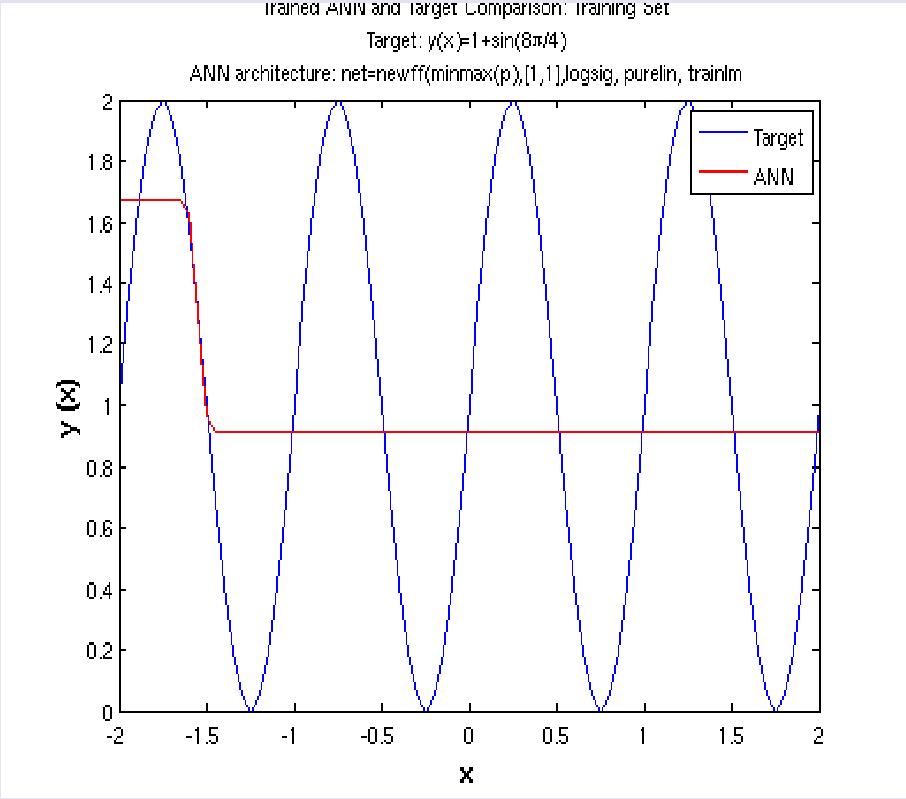
M. S. Khan, and S. W. Khor, 2004: Web document clustering using hybrid neural network, Applied Soft Computing, 4, 423-432.

- **Recognize Copper Cable Theft**

http://www.newscientist.com/article/dn21989-ai-system-helps-spot-signs-of-copper-cable-theft.html

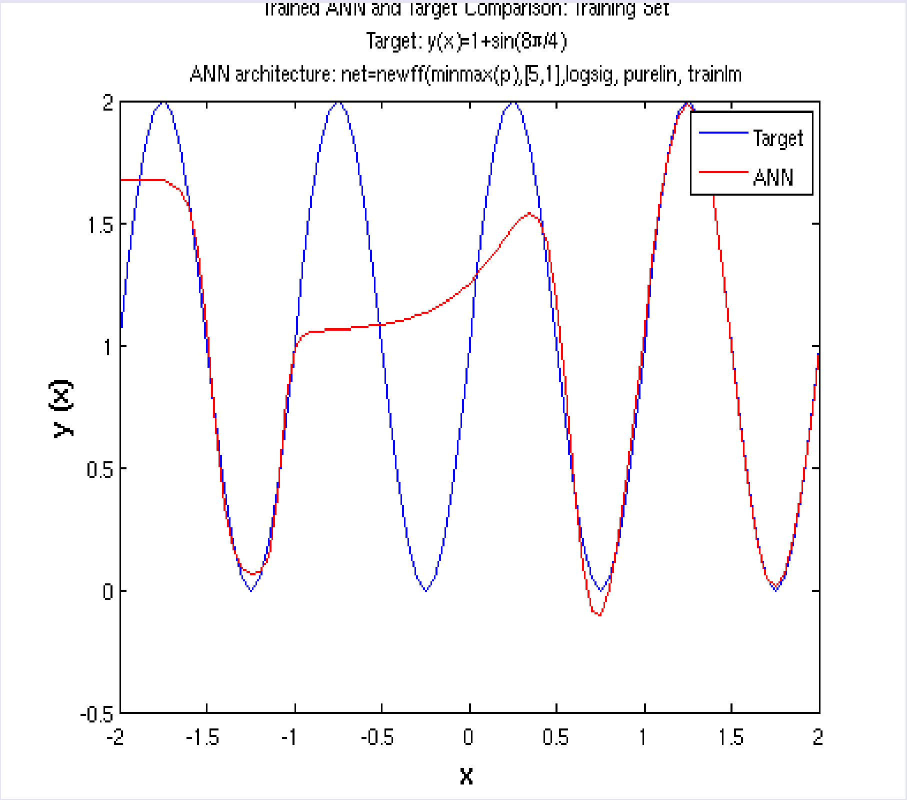
Artificial Neural Network (ANN)—Function Approximation Example 1: $y(x) = 1 + \sin 2\pi x$

Training Set



Topology: Feed-forward MLP [1,1,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

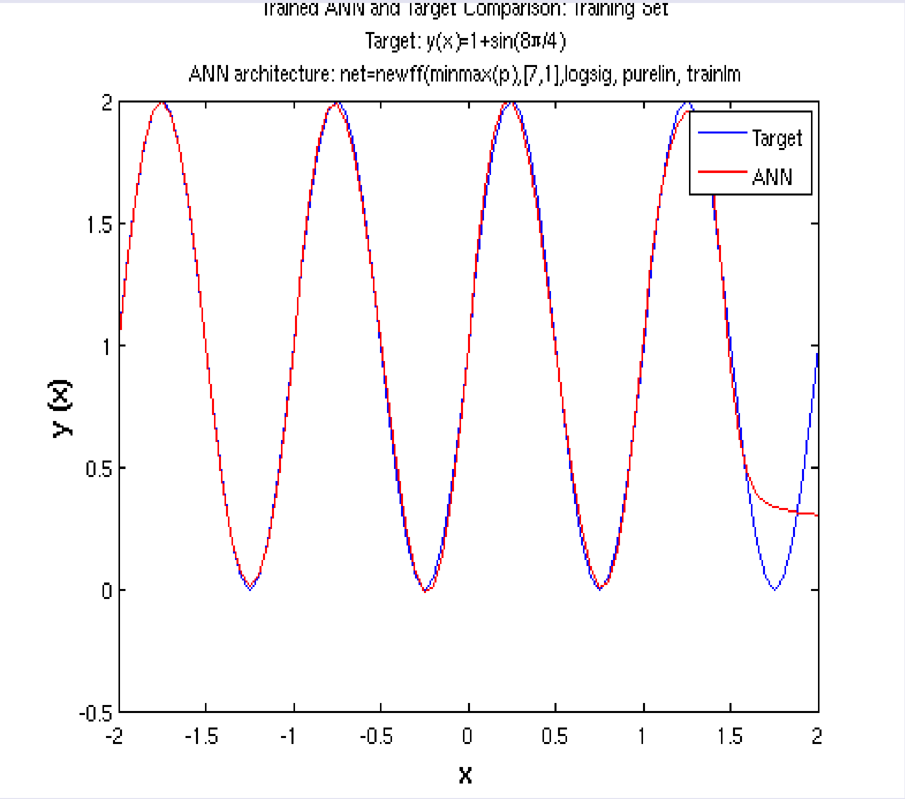
Training Set



Topology: Feed-forward MLP [1,5,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

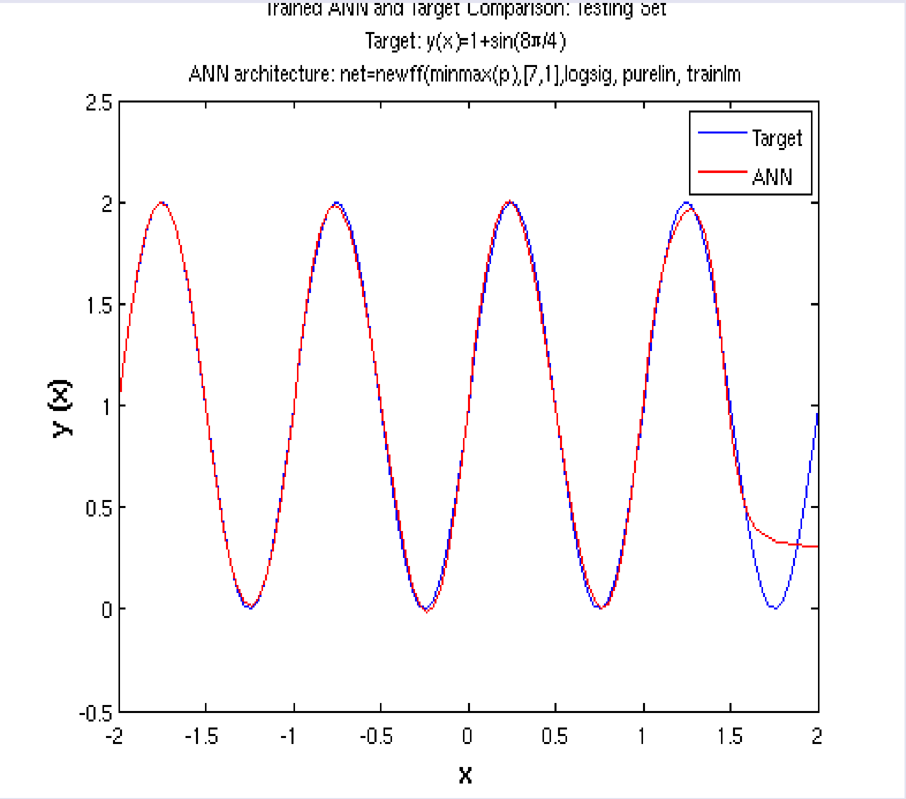
Artificial Neural Network (ANN)—Function Approximation Example 1: $y(x) = 1 + \sin 2\pi x$

Training Set



Topology: Feed-forward MLP [1,7,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

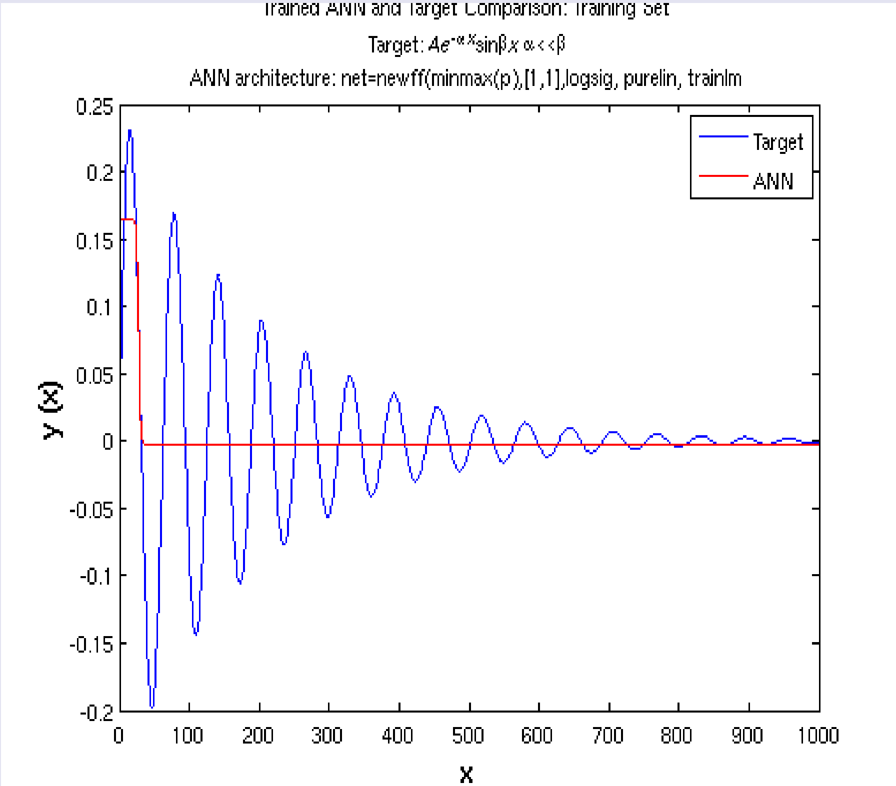
Training Set



Topology: Feed-forward MLP [1,7,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

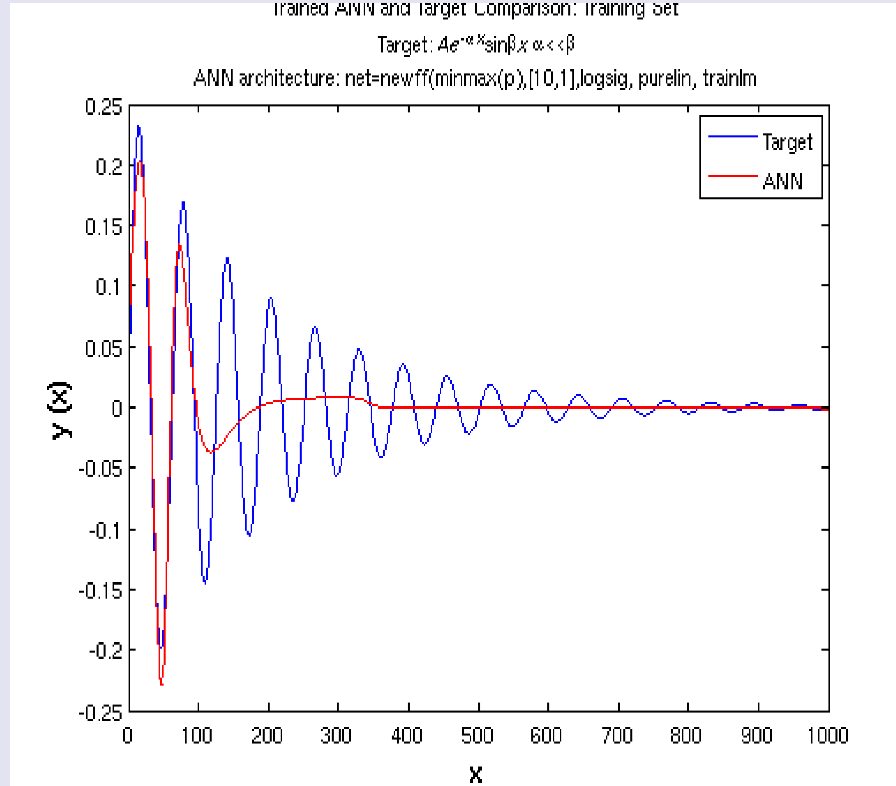
Artificial Neural Network (ANN)—Function Approximation Example 2: $y(x) = A(e^{-\alpha x})\sin\beta x, \alpha \ll \beta$

Training Set



Topology: Feed-forward MLP [1,1,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

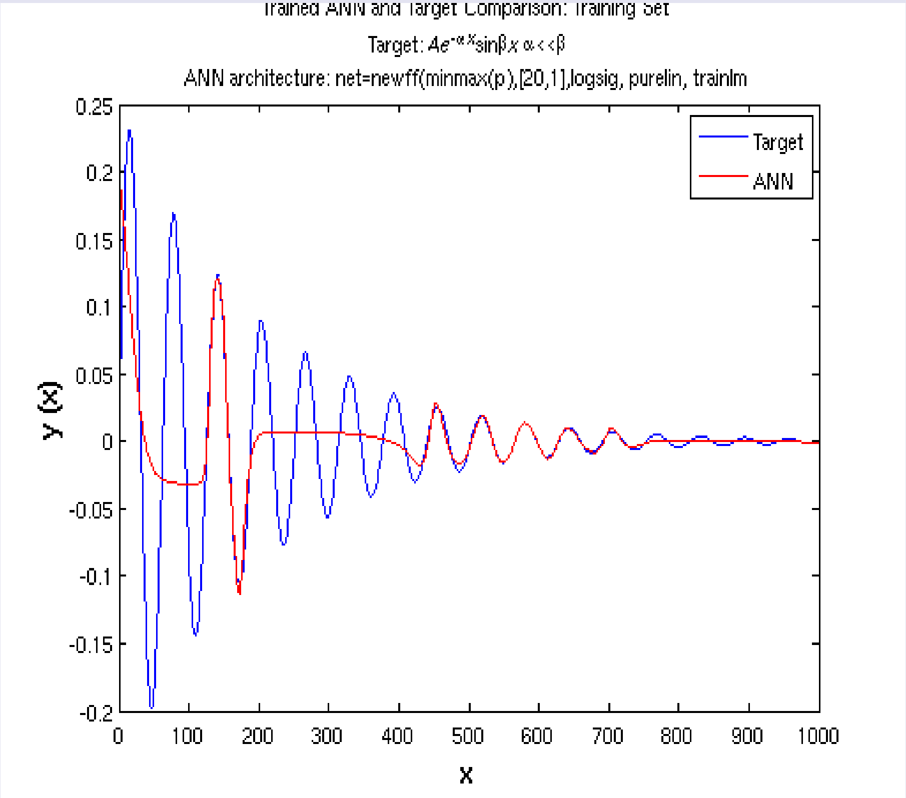
Training Set



Topology: Feed-forward MLP [1,10,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

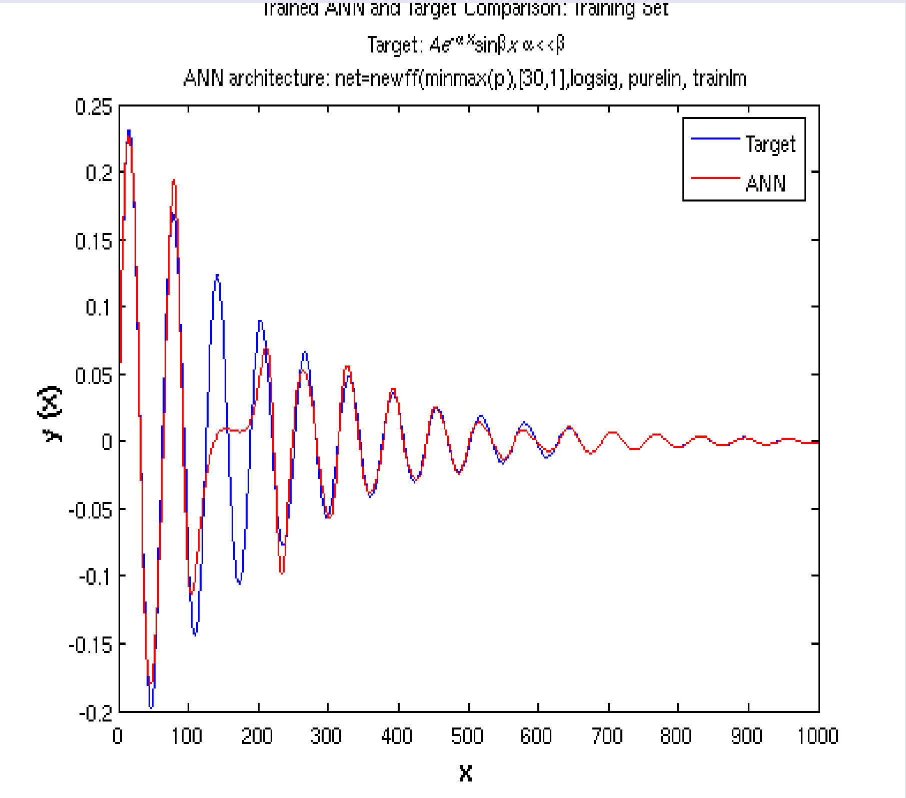
Artificial Neural Network (ANN)—Function Approximation Example 2: $y(x) = A(e^{-\alpha x}) \sin \beta x, \alpha \ll \beta$

Training Set



Topology: Feed-forward MLP [1,20,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

Training Set

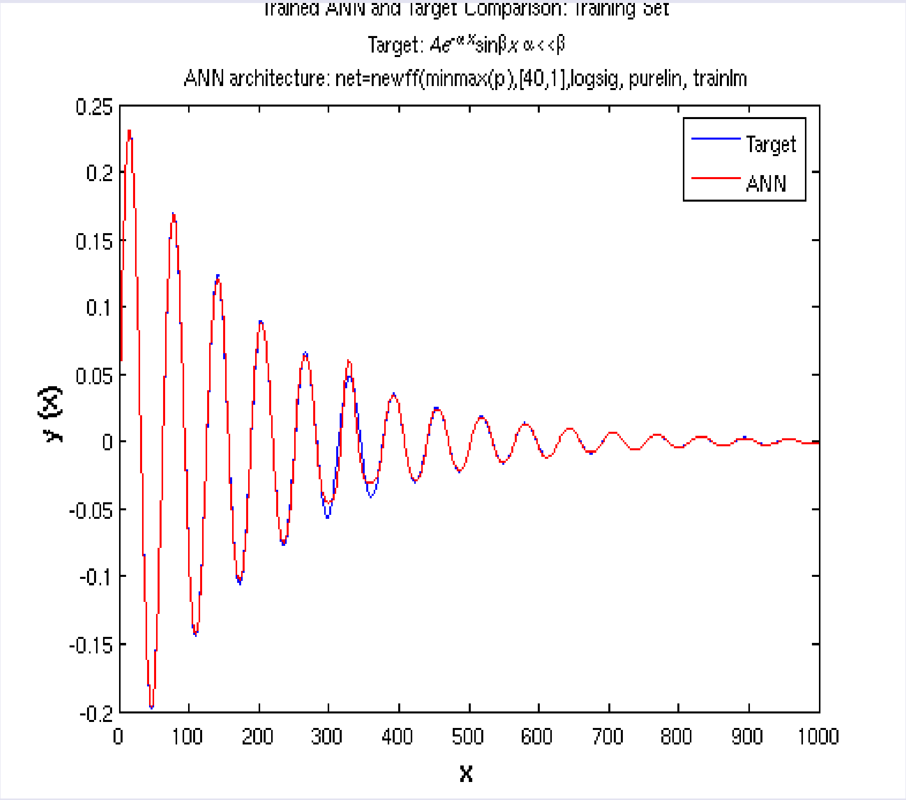


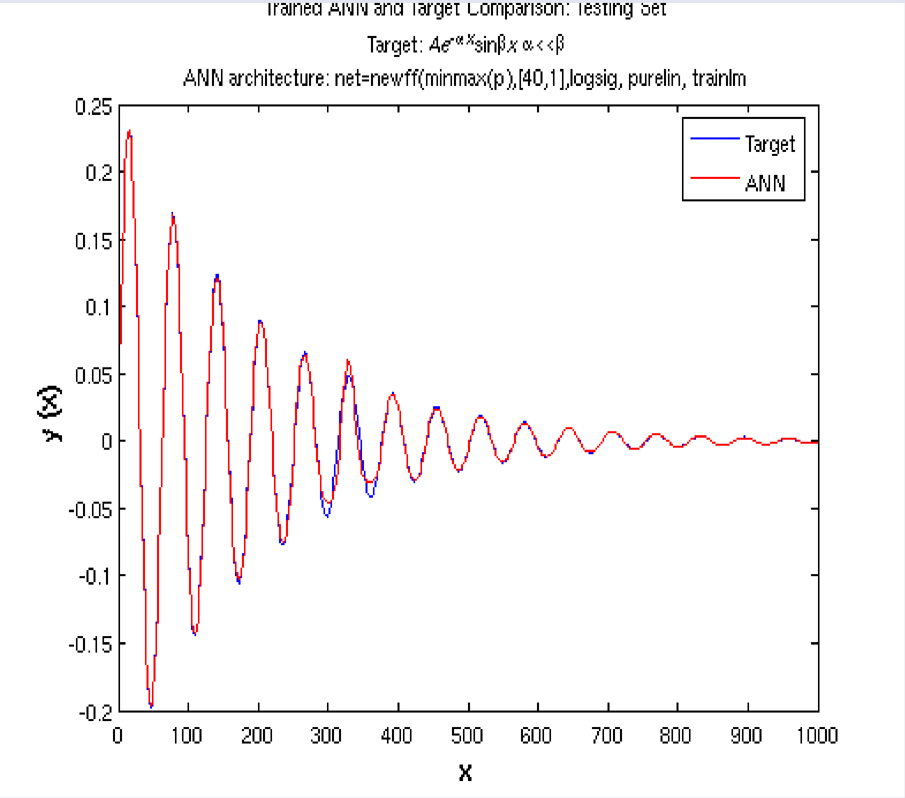
Topology: Feed-forward MLP [1,30,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

Artificial Neural Network (ANN)—Function Approximation Example 2:

$$y(x) = A(e^{-\alpha x})\sin\beta x, \alpha \ll \beta$$

Training Set **Testing Set**





Topology: Feed-forward MLP [1,40,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

Topology: Feed-forward MLP [1,40,1]
Transfer Functions: [log-sigmoid, linear]
Training Algorithm: TRAINLM

Artificial Neural Network — Classification (Yang et al 2000)

Application of artificial neural networks in image recognition of crop and weeds

Fig. 1. Examples of 100x100 cropped images.

Fig. 2. The ANN structure for Type 1 output

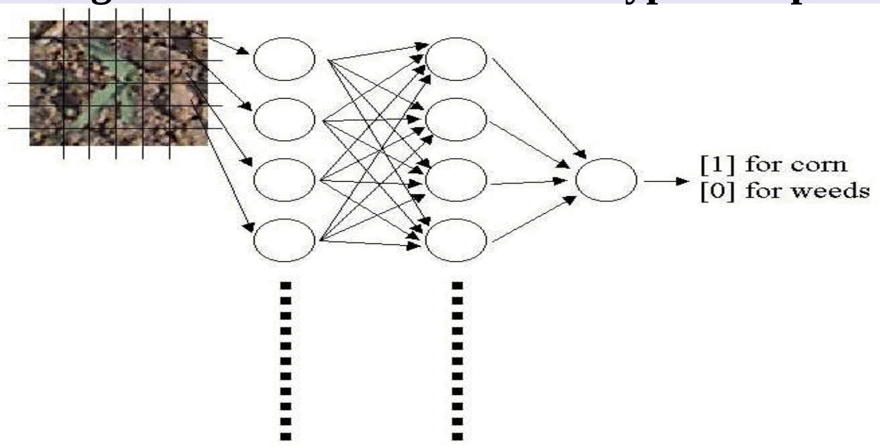
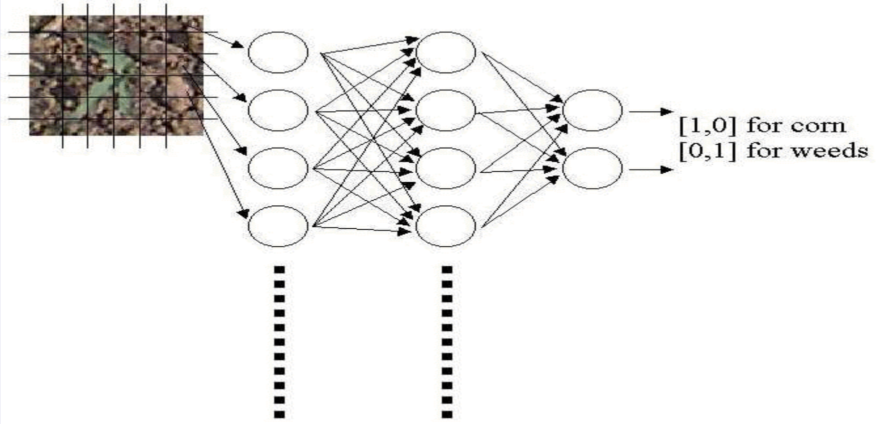


Fig. 3. The ANN structure for Type 2 output



Yang, C.C., S.O. Prasher, J.A. Landry, H.S. Ramaswamy, and A. Ditommaso, 2000: Application of artificial neural networks in image recognition of crop and weeds, Canadian Agricultural Engineering Vol. 42, 147-152

Artificial Neural Network — Classification (Yang et al 2000)

Application of artificial neural networks in image recognition of crop and weeds

Table I. The output types of ANNs and the threshold of classification

Type	Corn	Weeds
1-A	Output > 0.5	When the output did not match the condition in the left column
1-B	Output > Average value of all outputs	

- Image: KDC (756x504) -> BMP (100x100)-> RGB (100x100)
- RGB (0-255) -> RGB (0-1)
- ANN: 10000-1-1; log sigmoid in hidden layer
- 70 to 300 PE (neurons) in hidden layer
- Training: 40 weed and 40 corn images
- Back-propagation training algorithm
- Training stops at 2000 epochs unless 1X10^{-5} SSE achieved
- Testing: 10 other weed and corn images

Yang, C.C., S.O. Prasher, J.A. Landry, H.S. Ramaswamy, and A. Ditommaso, 2000: Application of artificial neural networks in image recognition of crop and weeds, Canadian Agricultural Engineering Vol. 42, 147-152

Artificial Neural Network — Classification (Yang et al 2000)

Application of artificial neural networks in image recognition of crop and weeds

Table II. Success classification rate for Type 1 ANNs

PEs in the hidden layer	Brier Score	Type 1-A		Type 1-B	
		Corn (%)	Weeds (%)	Corn (%)	Weeds (%)
70	0.23	80	60	70	70
80	0.17	90	60	80	70
90	0.23	80	40	60	80
100	0.17	80	80	70	80
110	0.15	100	60	80	70
120	0.19	80	50	70	80
130	0.21	70	70	60	70
140	0.25	60	50	60	70
150	0.20	90	60	70	60
160	0.27	80	50	80	50
180	0.19	90	60	90	70
200	0.24	90	40	90	50
220	0.21	90	60	90	70
230	0.17	90	60	90	70
240	0.24	90	50	90	60
260	0.22	90	50	80	60
280	0.21	90	60	80	60
300	0.19	90	70	90	70

Yang, C.C., S.O. Prasher, J.A. Landry, H.S. Ramaswamy, and A. Ditommaso, 2000: Application of artificial neural networks in image recognition of crop and weeds, Canadian Agricultural Engineering Vol. 42, 147-152

Thunderstorm Artificial Neural Network (TANN)

1. Motivation
2. Design/Framework
3. Model Development/Optimization Strategy
4. Most recent Results/Conclusions
5. Operational TANN

Thunderstorm Artificial Neural Network (TANN)

Motivation

Thunderstorm Prediction Models

Statistical

■**Logistic regression** (Sanchez et. al. 2001)

Relate the probability of one element of a binary outcome to a linear combination of predictor variables, using a non-linear logistic function; fit regression parameters using method of maximum likelihood (e.g. Wilks, 2006)

$$\mathbf{f}(\mathbf{z}) = \frac{1}{1+e^{-z}}\ ;\ z = \propto + \sum_{j=1}^{K} \beta_j x_j\ ;\ \text{ if } \mathbf{f}(\mathbf{z}) \left\{ \frac{\geq \theta, storm}{< \theta, no\ storm} \right.$$

■**Model Output Statistics** (MOS; Reap and Foster 1979; Schmeits 2005)

"Statistical relationships between model-forecast variables and observed weather variables, used for either correction of model-forecasts variables or prediction of variables not explicitly forecast by the model" (Glickman, 2000)

$y_t = f_{MOS}(\mathbf{x}_t)$ for both development and implementation (Wilks, 2006)

Wilks, D. S., 2006: Statistical Methods in the Atmospheric Sciences. 2nd Edition, Elsevier, Oxford.

Thunderstorm Artificial Neural Network (TANN)

Motivation

Thunderstorm Prediction Models

Statistical

■ **Multiple Discriminant Analysis** (e.g. McNulty 1981)

Discrimination: Estimate functions of the training data set (x_i) that best describe the separation of the known group (G) membership of each x_i

Discriminate Analysis (G=2): Determine a linear function [Fisher's Linear Discriminant] of the K elements of the observation vector x that best allows for future K-dimensional observation vector to be classified as belonging to either G=1 or G=2

Multiple Discriminant Analysis: Application of Fisher's linear discriminant for G >2.

Wilks, D. S., 2006: Statistical Methods in the Atmospheric Sciences. 2nd Edition, Elsevier, Oxford.

Thunderstorm Artificial Neural Network (TANN)

Motivation

Thunderstorm Prediction Models

Artificial Intelligence (AI)

■ **Artificial Neural Networks** (McCann 1992; Chaudhuri 2010)

Discussed in previous section

■ **Expert Systems** (Colquhoun 1987; Lee and Passner 1992)

Knowledge-based system, containing a *knowledge base* [human knowledge/understanding], *data base*, *inference engine* [receive observation, traverse the knowledge base, identify possible outcomes or conclusion(s)], and *machine/human interface*; used by non-expert to improve problem-solving abilities; used by experts to provide support/corroboration during the decision making process (de Silva, 2000)

Thunderstorm Artificial Neural Network (TANN)

Motivation

Thunderstorm Prediction Models

Numerical Weather Prediction (NWP)

"The integration of the governing equations of hydrodynamics by numerical methods subject to specified initial conditions" (Glickman, T. S., 2000)

- Mesoscale convection systems (Weisman et. al 1997; Weisman et. al. 2008)

Thunderstorm Artificial Neural Network (TANN)

Motivation

Limitations of previous studies

Predictors acquired solely from Rawinsonde output at particular time/location

■Chaudhuri 2010, Lee and Passner 1992, McNulty 1981

Predictors acquired solely from NWP Analysis

■Perler and Marchand 2009

(Use of NWP model predictions essential when predicting beyond 3 hours (Wilson et. al, 1998))

Predictors acquired from NWP predictions yet too coarse to account for processes that trigger individual storms

■McCann 1992, Schmeits 2005

High resolution (4-km) NWP models successful predicting thunderstorm occurrence/mode, yet not location/timing.

■Fowle and Roebber 2003, Weisman et. al. 2008

Higher resolution (≤1-km) NWP models may not provide additional skill owing to extreme sensitivity of model output to initial condition, parameterization trigger thresholds, predictability limits of NWP models

■Elmore et. al. (2002)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

- Incorporate selected variables/parameters from the 12-km Eta and WRF-NMM NWP model output (hereafter, NAM) related to Convective Initiation
- Incorporate sub-grid scale (4-km) surface heterogeneity (soil moisture patterns, gradients, and variability) related to CI via negative feedback (synoptically benign environments)
- Incorporate sub-grid soil moisture magnitudes related to CI via positive feedback
- The foregoing serves as input into a supervised feed-forward multi-layer perceptron ANN
- NAM output provides information to assess whether larger mesoscale environment is conducive to CI, while sub-grid scale data may identify locations where CI is most likely to occur.
- By using AI, the TANN will learn from the less than optimal NAM output.

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

- Domain: Grid of 13 x 22 equidistant points; 20-km grid spacing. Grid points serve as boundaries for 286 20-km x 20-km square regions ("boxes"). Area slightly larger than WFO CRP Area of Responsibility.

(400-km^2 box regions are sufficiently large enough to support soil moisture heterogeneities of a scale large enough to generate mesoscale circulations. Taylor et. al, 2007; Taylor et. al, 2011)

- For each box, train an ANN using selected NAM and soil moisture data
- Interpolated 12-km NAM output at the center of each box.
- 4-km soil moisture for each grid cell (HRAP grid) within each box computed using the Antecedent Precipitation Index (API) with the WGRFC MPE (4-km) as source data.
- National Lightning Detection Network (NLDN) CTG lightning data serve as thunderstorm proxy during supervised training, validation, and testing.

Thunderstorm Artificial Neural Network (TANN)

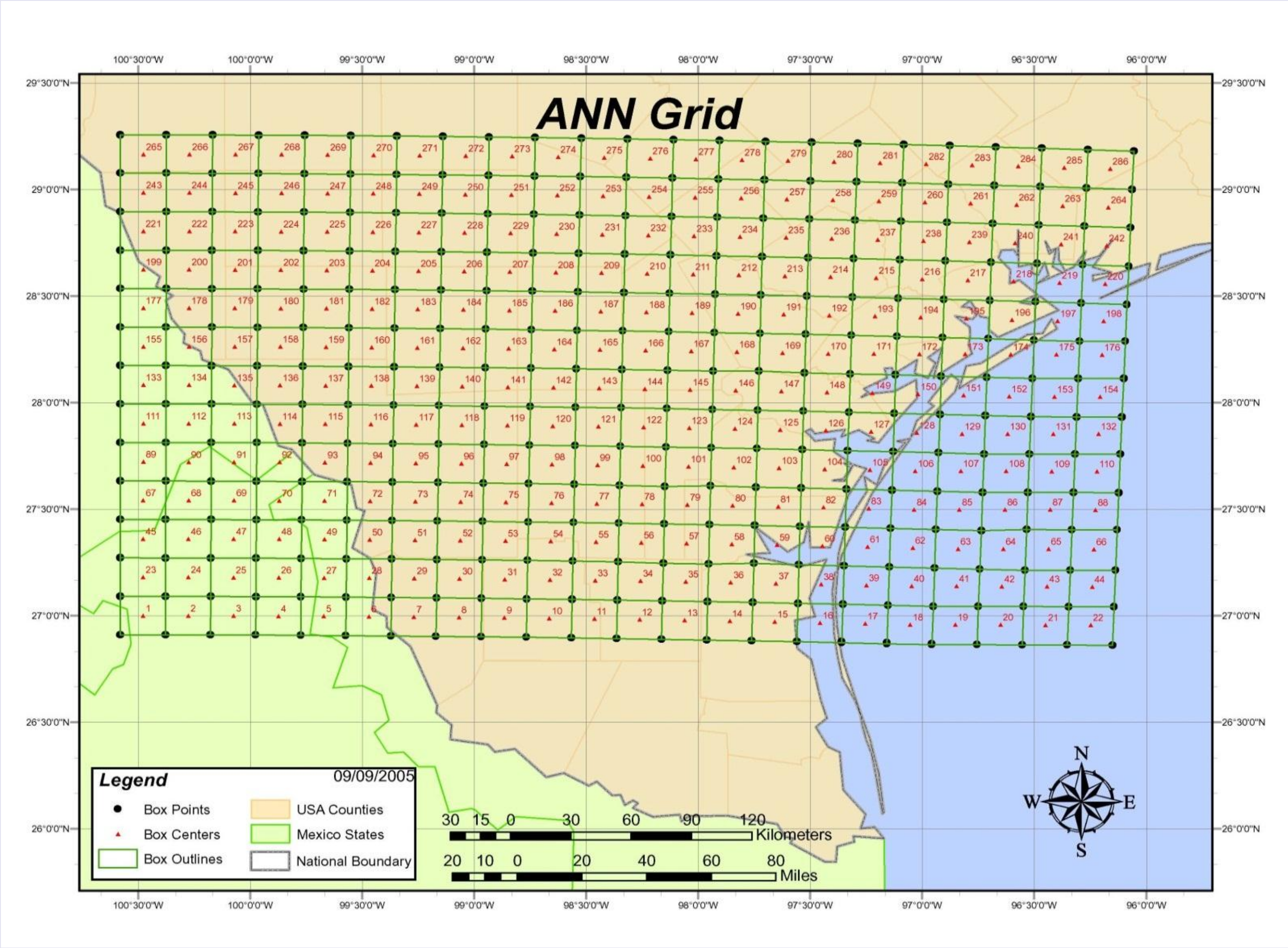
Model Design/Framework

Predict thunderstorms at a higher spatial resolution relative to previous studies

Study	Verification Domain	Prediction Period (h)	Model Type
McNulty (1981)	1^{o} radius	12	Discriminate Analysis
McCann (1992)	1^{o} grid	3-7	ANN
Lee and Passner (1993)	100-km radius	12	Expert System
Sanchez et. al. (2001)	6825 km^2	6-12	Logistic Regression
Schmeits et. al. (2005)	7200 km^2	48	MOS
Manzato (2005)	5000 km^2	6	ANN
Perler and Marchand (2009)	729 km^2	24	Adaptive Boosting
Current Study	**400km^2**	12	ANN

Thunderstorm Artificial Neural Network (TANN)

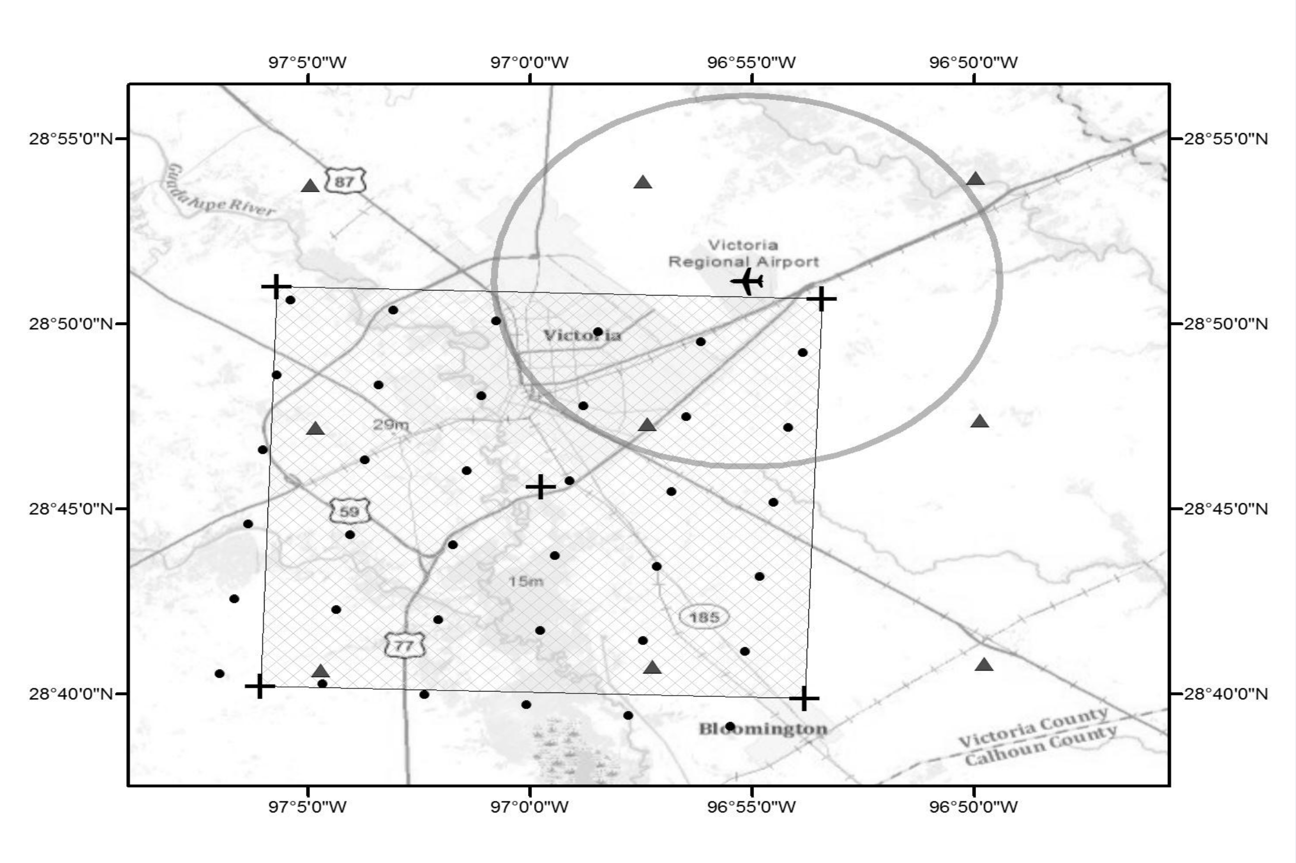
Model Design/Framework



Input latitude-longitude values: *Inverse* and *Forward* computer software (United States National Geodetic Society) **Output**: GIS software run by Rick Smith (TAMUCC)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework



Box 238 (smaller box region); (+) center and four corners of the box. Locations of NAM grid points (12-km) (▲); values of NAM parameters/ variables based on bi-linear interpolation at box center. (●) MPE grid points (4-km). Airplane symbol depicts location of VCT METAR; circle depicts 9.26 km (5.00 nm) radius relative to VCT. Figure credits: Sergei Reid, Julien Clifford (TAMUCC)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

TANN Input Variables/Parameters: NAM Predictions

Abbreviation	Description (Units)
PWAT	Precipitable water (mm)
MR850	Mixing ratio at 850-hPa (gkg^{-1})
RH850	Relative humidity at 850-hPa (%)
CAPE	Convective Available Potential Energy (surface-based parcel) (Jkg^{-1})
CIN	Convective Inhibition (Jkg^{-1})
LI	Lifted Index (K)
Uxxx,Vxxx	U,V wind components at the surface and at 850-hPa (ms^{-1})
VVxxx	Vertical velocity at 925, 700, and 500-hPa (Pas^{-1})
Dropoff Proxy	Proxy for Potential Temperature Dropoff (K) (Crook, 1996)
LCL	Lifted Condensation Level (m)
LCL_T	Lifted Condensation Level Temperature (K)
CP	Convective Precipitation (kgm^{-2})
VShearS8	Vertical wind shear in the surface to 800-hPa layer ($x10^{-3}$ s^{-1})
VShear86	Vertical wind shear in the 800 to 600-hPa layer ($x10^{-3}$ s^{-1})

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

TANN Input Variables/Parameters: NAM Initialization

Abbreviation	Description (Units)
Uxxx,Vxxx	U,V wind components at the surface and at the 900,800,700, 600, and 500-hPa levels (ms^{-1})
HI_{low}	Humidity Index (^oC) (Findell and Eltahir, 2003b)
CTP Proxy	Proxy for Convective Triggering Potential (dimensionless) (Findell and Eltahir, 2003a)
VShearS7	Vertical wind shear in the surface to 700-hPa layer ($x10^{-3}$ s^{-1})

TANN Input Variable/Parameter: Miscellaneous

Abbreviation	Description (Units)
JD	Julian Day (0-366)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

Sub-grid Scale (4-km) Variables/Parameters used in TANN

Abbreviation	Description (Units)
SMC_MAXGRAD	Maximum *Soil Moisture* Content (SMC) Gradient ($kgm^{-2}km^{-1}$)
SMC_SD	*Soil Moisture* Content (SMC) Standard Deviation (kgm^{-2})
SMC_MEAN	Mean *Soil Moisture* Content (SMC) (kgm^{-2})
SMC_MAX	Maximum *Soil Moisture* Content (SMC) (kgm^{-2})
GMORANI	Global Moran's I of *Soil Moisture* Content (SMC) (unit less)
NDRY	Number of dry days [MPE=0] since the previous rain day (days)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework -- Computation of SMC

$$SMC(i,j,t)_{init} = SMC(i,j,t)_{nldas}$$

$$API(i,j,t+1) = MPE(i,j,t+1) + \theta API(i,j,t)$$

$$\theta = 1 - 0.04\left\{sin\left[2\pi\left(\frac{(jd-a-b)}{c}\right)\right] + 1\right\}$$

$$FSM(i,j,t) = \frac{API(i,j,t)}{API_{max}}$$

$$API_{max} = (1-\mu)^{-1}$$

$$if\ API(i,j,t) > API_{max}\ then\ API(i,j,t) = API_{max}$$

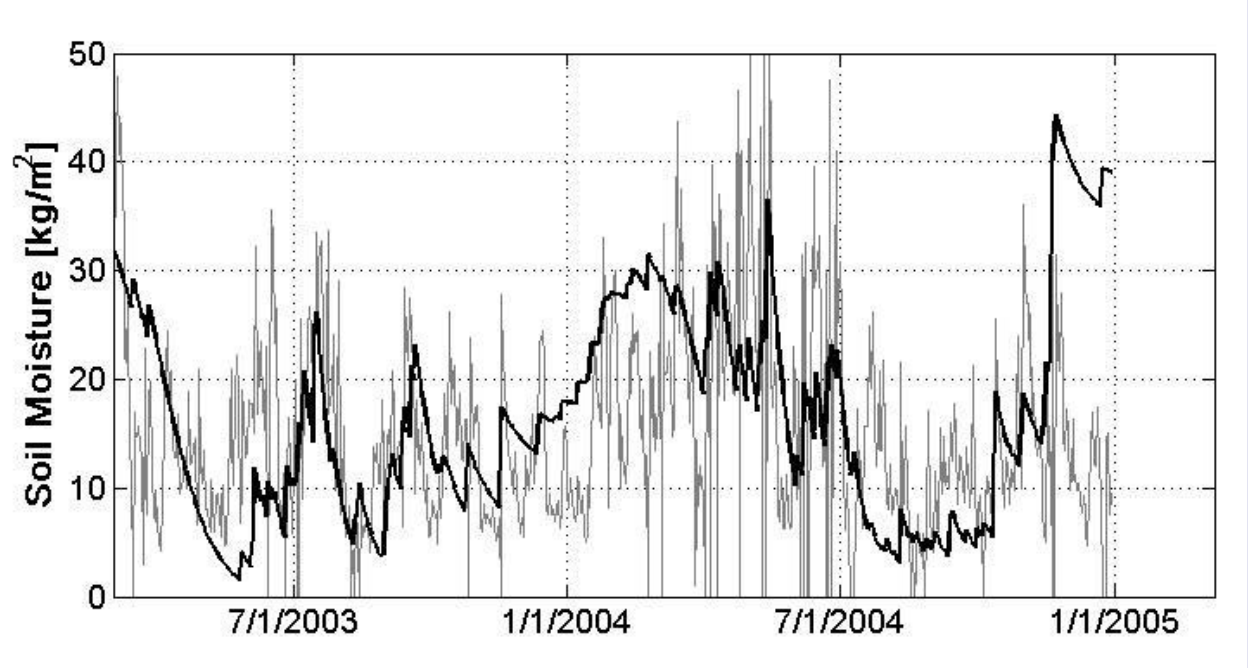
$$VSM(i,j,t) = SVMC(i,j,t) \times FSM(i,j,t)$$

$$SMC(i,j,t)[kgm^{-2}] = VSM(i,j,t)[m^3m^{-3}] \times 10^3\ [kgm^{-3}] \times d\ [m]$$

Cheng, W. Y. Y., and W. R. Cotton, 2004: Sensitivity of a Cloud-Resolving Simulation of the Genesis of a Mesoscale Convective System to Horizontal Heterogeneities in Soil Moisture Initialization. *Journal of Hydrometeorology*, 5, pages 934-958.

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework -- Computation of SMC



Comparison of mean soil moisture content (SMC) from the API model (solid black line) to that from the **Land Surface Microwave Emission Model** (LSMEM; gray) based on **TRMM** (Tropical Rainfall Measuring Mission) Microwave Imager 10.65 GHz (X-band) radiometer (Gao et. al, 2006), for Box 238.

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – Computation of Soil Moisture Heterogeneity

Soil Moisture Heterogeneity

1. Variability

SMC_MAXGRAD: Maximum *Soil Moisture* Content (SMC) Gradient (kgm-2km-1) [MATLAB *Gradient* Function]

SMC_SD: *Soil Moisture* Content (SMC) Standard Deviation (kgm^{-2})

2. Pattern

GMORANI: Global Moran's I of *Soil Moisture* Content (SMC) (unit less)

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – Global Moran's I (GMORANI)

GMORANI Derivation:

Spatial Autocorrelation (Γ_{ij}) (Fischer and Getis, 2010)

$$\Gamma_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} Y_{ij}$$

Γ_{ij}= spatial autocorrelation for n georeferenced observations
i= specific location under consideration; *j*= surrounding locations within a particular range
W_{ij} = Weight matrix describing spatial relationship between *i* an *j*
Y_{ij} = Matrix describing (non-spatial) association between realizations of variable at *i* and *j*

Fischer, M. M., and A. Getis (eds.), 2010: *Handbook of Applied Spatial Analysis: Software Tools, Methods and Application*s, Springer-Verlag Berlin Heidelberg, DOI 10.1007/978-3-642-03647-7_14,

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – Global Moran's I (GMORANI)

GMORANI Derivation continued:

Setting Y_{ij} as a covariance matrix and multiply by $\frac{n}{W}[\sum_{i=1}^{n} y_i - \bar{y}^2]$, where $W = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}$, Γ_{ij} becomes Moran's I:

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(y_i - \bar{y}) - (y_j - \bar{y})}{\sum_{i=1}^{n} y_i - \bar{y}^2} \qquad i \neq j$$

Global Moran's I (GMORANI): Mean of all local Moran's I values computed at each grid point within the 20-km x 20-km box region

Fischer, M. M., and A. Getis (eds.), 2010: *Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications*, Springer-Verlag Berlin Heidelberg, DOI 10.1007/978-3-642-03647-7_14,

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – GMORANI: Idealized Examples

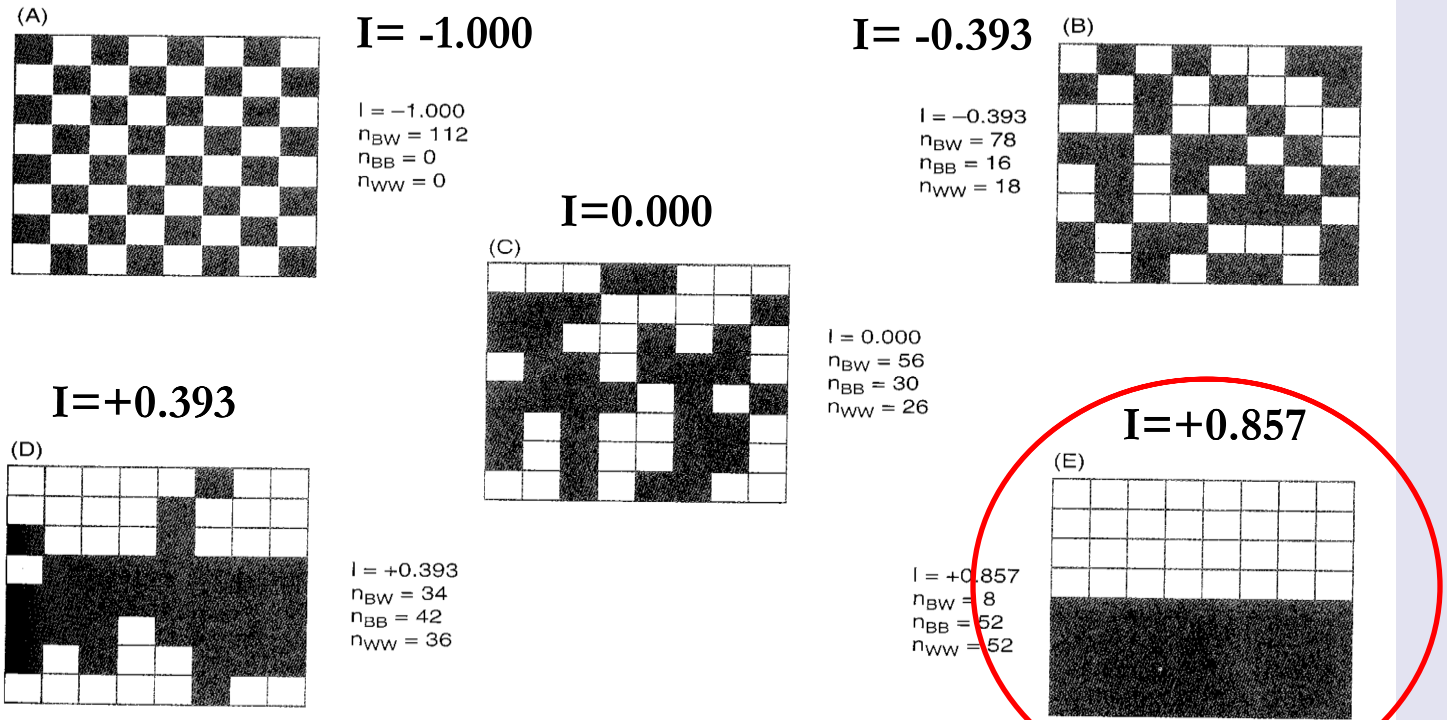


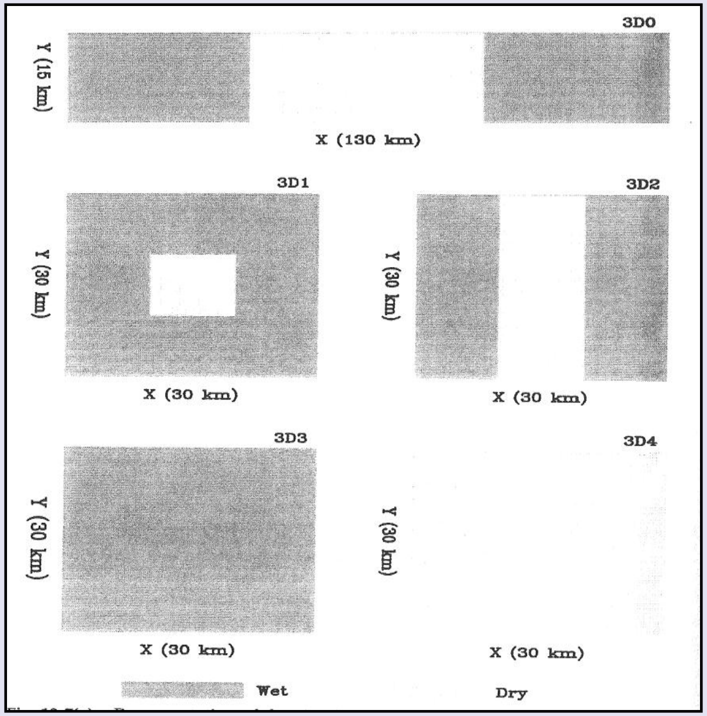
Figure 4.1 Field arrangements of blue and white cells exhibiting: (A) extreme negative spatial autocorrelation; (B) a dispersed arrangement; (C) spatial independence; (D) spatial clustering; and (E) extreme positive spatial autocorrelation. The values of the *I* statistic are calculated using the equation in Section 4.6 (*Source*: Goodchild 1986 CATMOG, GeoBooks, Norwich)

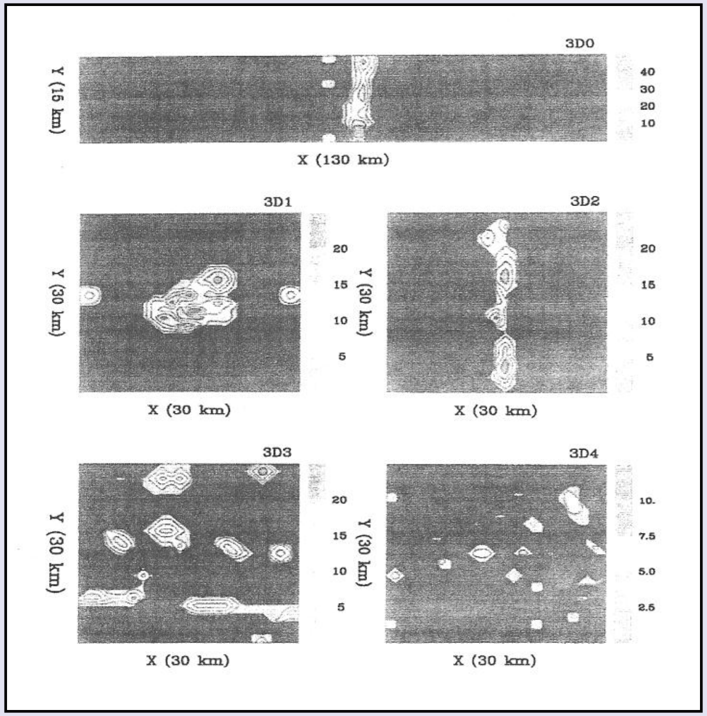
Longley, P. A., M. F. Goodchild, D.J. Maguire, and D. W. Rhind, 2005: Geographic Information Systems and Science, 2nd Edition, John Wiley & Sons, Ltd., 517pp.

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework

Convective Initiation via negative feedback involving surface heterogeneity

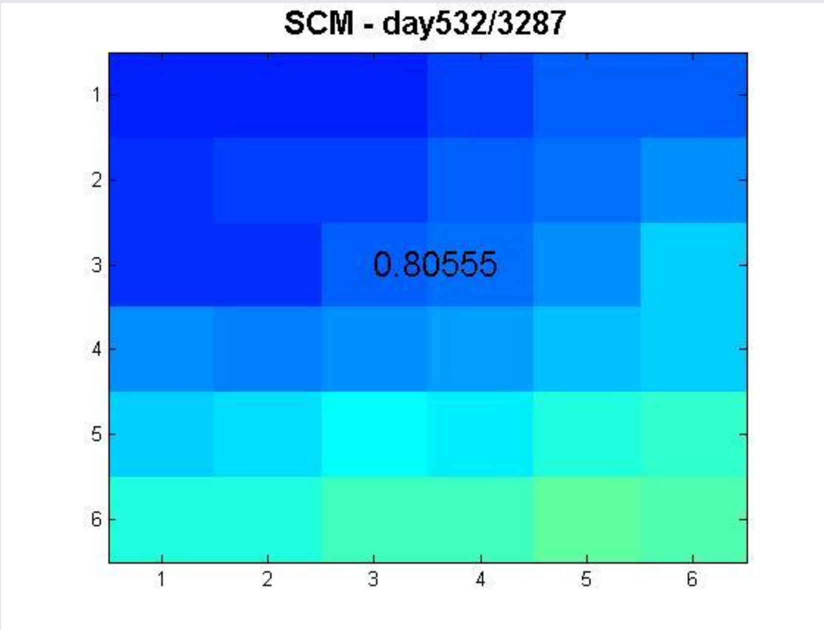


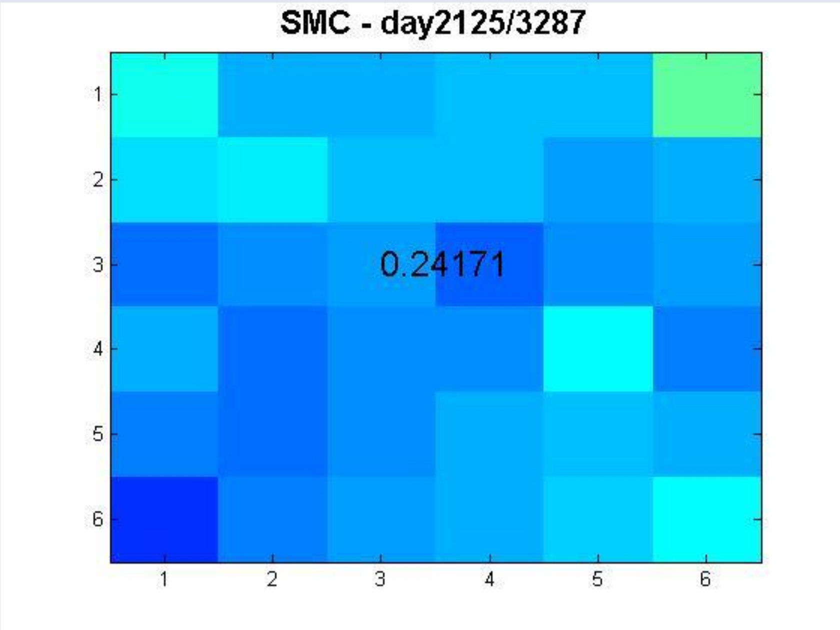


Source: Avissar, R., and Y. Liu. 1996. Three-dimensional numerical study of shallow convective clouds and precipitation induced by land surface forcing. *J. Geophys. Res.* 101, 7499-7518.

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – SMC/Heterogeneity TANN Input Parameters: Example (TANN Domain Box 238)

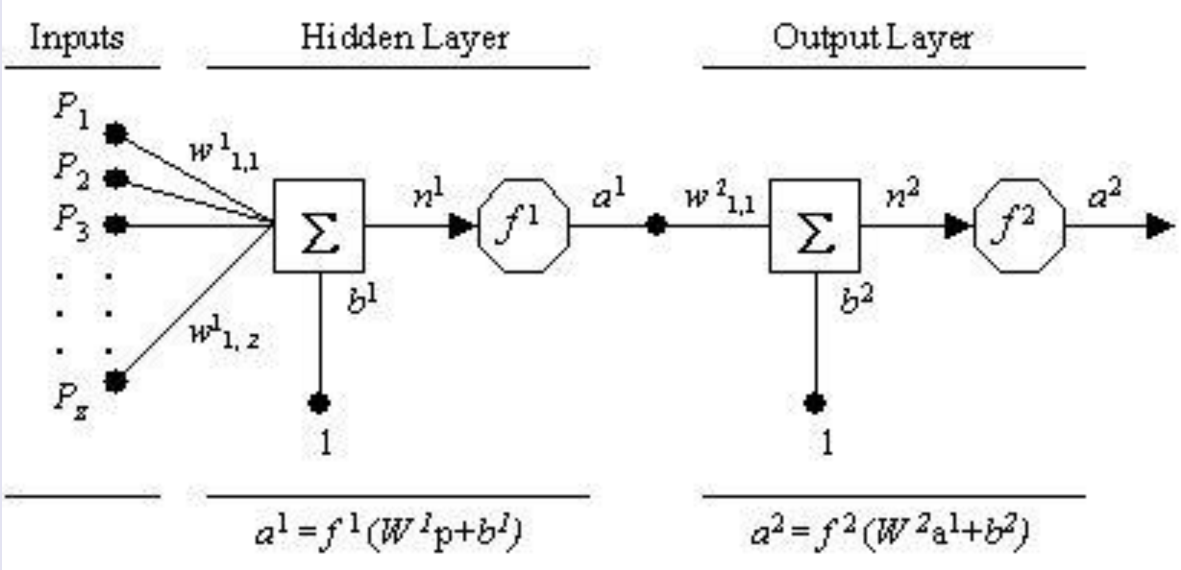




SMC Day 532/3287	Parameter (Units)	SMC Day 2125/3287
0.81	GMORANI	0.24
2.61	SMC_MAXGRAD $(kgm^{-2}km^{-1})$	1.66
1.50	SMC_MEANGRAD $(kgm^{-2}km^{-1})$	0.35
5.96	SMC_SD (kgm^{-2})	1.07
17.98	SMC_MEAN (kgm^{-2})	6.04
32.59	SMC_MAX (kgm^{-2})	9.39
1.00	NDRY	1.00

Thunderstorm Artificial Neural Network (TANN)

Model Design/Framework – TANN Topology, Transfer Function & Learning Rules



- **Topology:** Feed-forward, multi-layer perceptron, 1-hidden layer
- **Transfer Function:** log-sigmoid (hidden layer), linear (output layer)
- **Learning Rule/Training Algorithm:** Scaled conjugate gradient
- **Training Category:** Supervised
- **Inputs:** NAM (Eta,WRF-NMM), JDay, Sub-grid scale SMC parameters
- **Data preprocessing:** Transform inputs to range (-1,1)
- **Target:** NLDN

Thunderstorm Artificial Neural Network (TANN)

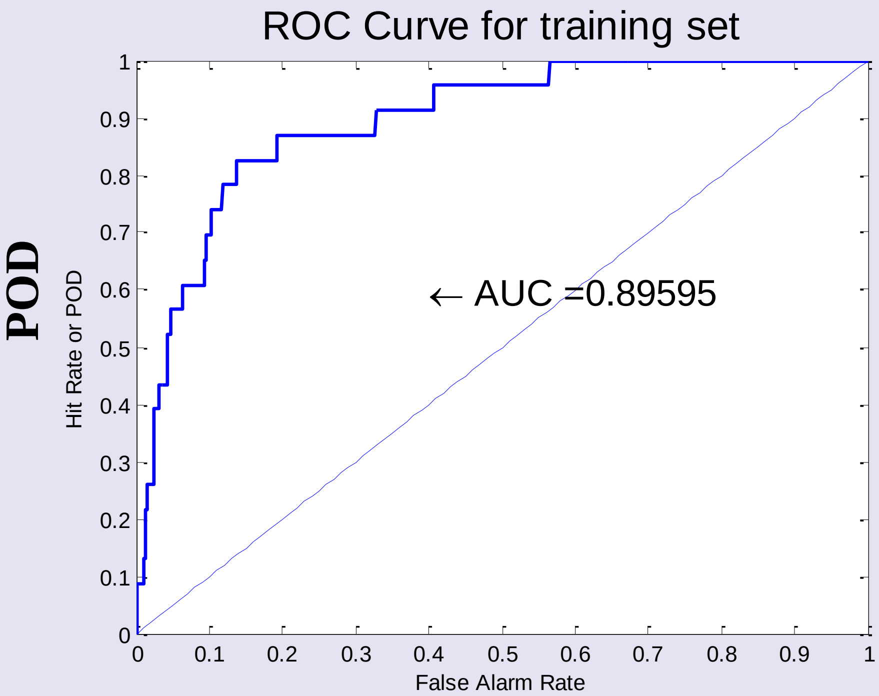
Model Development/Optimization Strategy

- (A) **Model "Optimization**":
  - Data: 3/1/2004 – 12/31/2006 & 1/1/2009 – 12/31/2011
  - Randomly divide data in training /validation/testing sets [0.4,0.2,0.4]
  - Train/validate model
  - Develop Relative (or Receiver) Operator Characteristic (ROC) curve based on testing set
  - Use ROC curve to (1) assess model skill and (2) select decision threshold that optimizes model performance
- (B) **"Optimized" Model Evaluation**:
  - Data: 1/1/2007 – 12/31/2008
  - Comparison with NDFD, TANN-MOS, and TAF

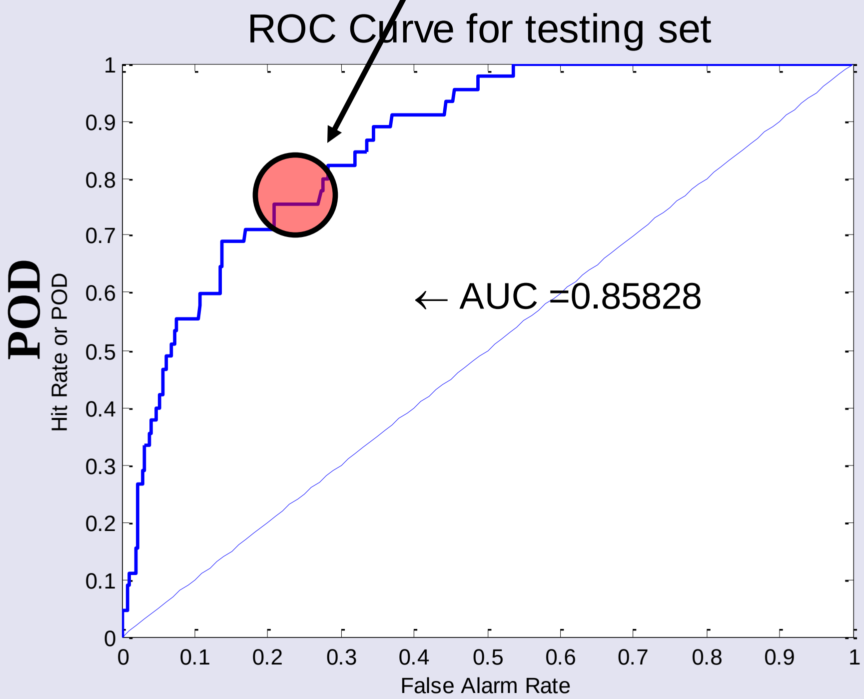
Thunderstorm Artificial Neural Network (TANN)

Model Development/Optimization Strategy

Threshold selection, preferably: POD >75% & F<25%



False Alarm Rate



False Alarm Rate

Thunderstorm Artificial Neural Network (TANN)

Most Recent Model Performance Results: Metrics

Scalar Performance Metrics

Performance Metric [Value Range]	Symbol	Equation
Probability of Detection [0,1]	POD	a/(a+c)
False Alarm Rate [0,1]	F	b/(b+d)
False Alarm Ratio [0,1]	FAR	b/(a+b)
Critical Success Index [0,1]	CSI	a/(a+b+c)
Peirce Skill Score [-1,1]	PSS	(ad-bc)/[(b+d)(a+b)]
Heike Skill Score [-1,1]	HSS	2(ad-bc)/[(a+c)(c+d)+(a+b)(b+d)]
Yule's Q (Odds Ratio Skill Score) [-1,1]	ORSS	(ad-dc)/(ad+bc)
Clayton Skill Score [-1,1]	CSS	(ad-bc)/[(a+b)(c+d)]
Gilbert Skill Score [-1/3,1]	GSS	$(a-a_r)/(a+b+c-a_r)$; $a_r=[(a+b)(a+c)]/n$

Thunderstorm Artificial Neural Network (TANN)

Performance: TANN (43-1-1), TANNMOS (36-1-1), TAF, NDFD

Box 238: 9-12 hour Prediction Performance Results: Scalar Metrics (2007-2008)

6-9 hour Forecasts									
	POD	**FAR**	**F**	**PSS**	**CSI**	**HSS**	**YuleQ**	**CSS**	**GSS**
TAF	0.54	0.79	0.08	0.46	0.18	0.26	0.86	0.19	0.15
9 hour Predictions/Forecasts									
TANN	0.92	0.79	0.36	0.57	0.21	0.22	0.90	0.20	0.13
TANN-MOS	0.94	0.80	0.37	0.54	0.20	0.20	0.90	0.19	0.11
NDFD	0.96	0.78	0.34	0.62	0.22	0.25	0.96	0.21	0.14
9-12 hour Forecasts									
TAF	0.38	0.83	0.06	0.32	0.14	0.20	0.80	0.15	0.11
12 hour Predictions/Forecasts									
TANN	0.96	0.88	0.40	0.54	0.11	0.12	0.92	0.11	0.07
TANN-MOS	0.88	0.89	0.38	0.48	0.11	0.12	0.83	0.10	0.06
NDFD	0.73	0.88	0.27	0.46	0.12	0.14	0.76	0.10	0.07

Thunderstorm Artificial Neural Network (TANN)

Conclusions (Box 238)

• TANN, TANN-MOS, and NDFD demonstrate comparable skill for forecast/prediction hours 3, 6, 9, and 12

• TANN, TANN-MOS demonstrate greater skill than NDFD for forecast/prediction hour 12, yet only with regard to *Yule Q* and *Pierce Skill Score*.

• Sub-grid scale data did not provide significant improvement within TANN

• In the context of *Random Forest* variable importance output, the sub-grid data is important (discussed later)

Thunderstorm Artificial Neural Network (TANN)

Current Operational TANN (Older version) –18 June 2012 Case

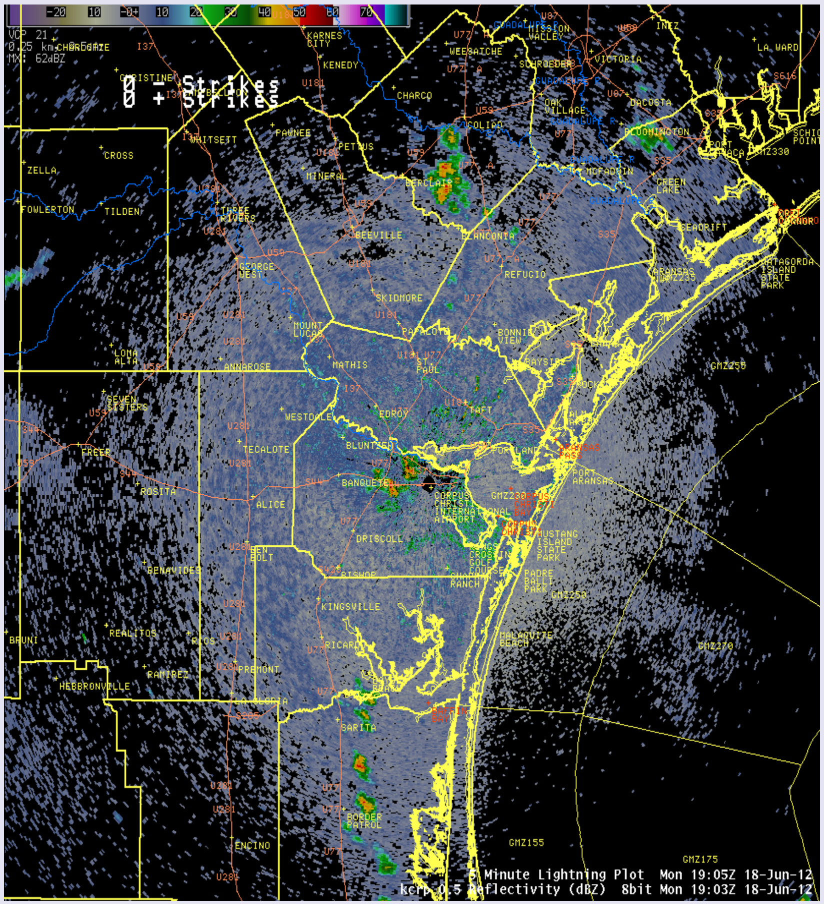
AWIPS Text Product: CRPWRKANN (Since 4 March 2009)

```
ANN Predictions for onset of Thunderstorms for Box 103
*****************************************************
Predictions for 12Z cycle of Jun 18, 2012 : 3 6 9 12 hour Predictions

Model trained based on data from June 1, 2004 to October 31, 2007
The data was split in 3 sets - learning/validation/testing.  The
thresholds were set
to optimize performance based on the ROC curve (POD vs. FAR) on the
testing set.

Result of ANN thunderstorm for   3 hour prediction = 0.015
The threshold for this model is 0.030
The occurence of thunderstorms for box 103 is therefore not predicted
in   3 hours

Result of ANN thunderstorm for   6 hour prediction = 0.056
The threshold for this model is 0.115
The occurence of thunderstorms for box 103 is therefore not predicted
in   6 hours

Result of ANN thunderstorm for   9 hour prediction = 0.053
The threshold for this model is 0.025
The occurence of thunderstorms for box 103 is therefore predicted in
9 hours

Result of ANN thunderstorm for  12 hour prediction = 0.047
The threshold for this model is 0.085
The occurence of thunderstorms for box 103 is therefore not predicted
in 12 hours
```

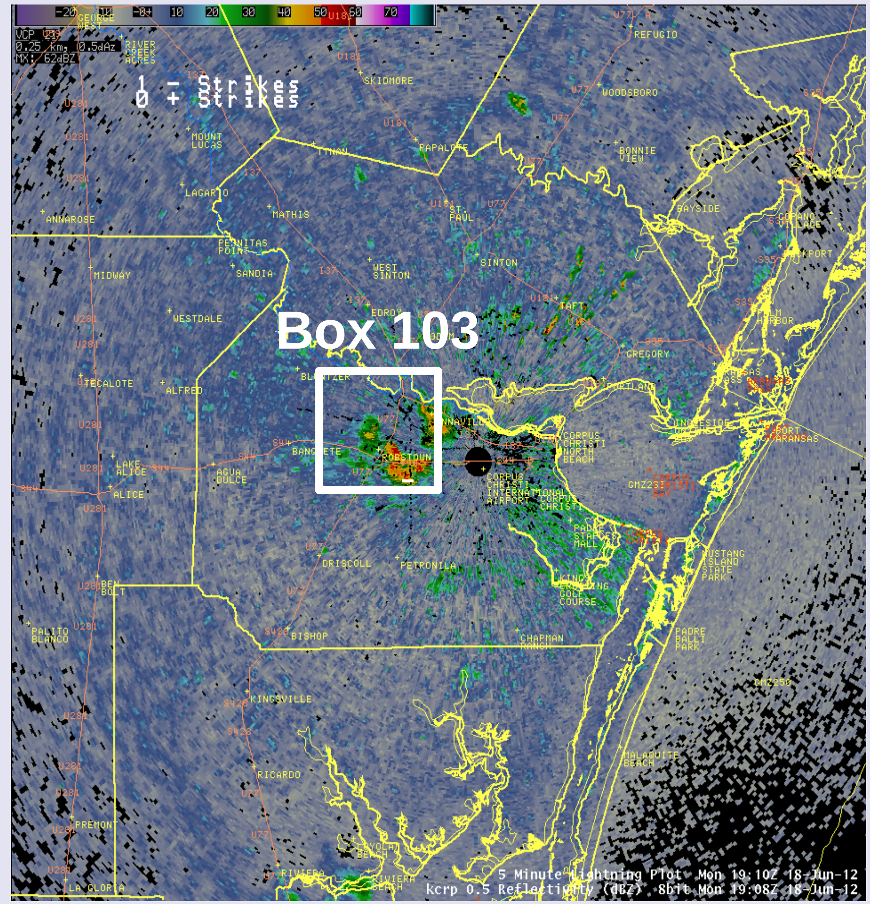
```
---------------LEGEND---------------
Box 101 [ALI ASOS Included]
Box 103 [RBO AWOS Included]
Box 190 [BEA AWOS Included]
Box 238 [VCT ASOS Included]
3 hour prediction valid 13-17 UTC
6 hour prediction valid 16-20 UTC
9 hour prediction valid 19-23 UTC
12 hour prediction valid 22-02 UTC
```

Thunderstorm Artificial Neural Network (TANN)

Current Operational TANN (Older version) **–18 June 2012 Case**



Z_e: 1905 UTC 18 June 2012



Z_e: 1908 UTC 18 June 2012

CTG: 1910 UTC 18 June 2012

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

NASA-funded project proposal submitted to the NASA ROSES 2009 Gulf of Mexico solicitation (NNH09ZDA001N,A40)

- NASA Research Opportunities in Space and Earth Sciences (ROSES) funded project
- **Principal Investigators:**
  John Mecikalski, University of Alabama in Huntsville (UAH)
  John Williams, National Center for Atmospheric Research (NCAR)
  Philippe Tissot, Texas A&M University-Corpus Christi (TAMUCC)
- **Others:** David Ahijevych (NCAR), Nathan Bledsoe (UAH), Waylon Collins (NOAA), Chris Jewett (UAH), Wayne MacKenzie (NOAA), Whitney Rutledge (TAMUCC), Mathew Saari (UAH), John Walker (UAH).

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

Models developed/improved to be evaluated by WFO CRP forecasters and others

Feasibility study. If successful, will lead to improvements in the *Convective Nowcast Oceanic* (CNO; Kessinger et al. 2009) and *Global Turbulence decision support systems* (Williams et al. 2009) (currently under development with NASA funding) for use in the *World Area Forecast System (WAFS)*. WAFS was initiated in 1982 by the World Meteorological Organization (WMO) and the International Civil Aviation Organization (ICAO) for provision of international aircraft operations-related weather guidance

Kessinger, C., H. Cai, N. Rehak, D. Megenhardt, M. Steiner, R. Bankert, J. Hawkins, M. Donovan, and E.R. Williams, 2009: The oceanic convection diagnosis and nowcasting system. *16th Conf. Satellite Meteor. Ocean.*, Amer. Met. Soc., Phoenix, AZ, 12-15 Jan. 2009.

Williams, J.K., R. Sharman, C. Kessinger, W. Feltz, A. Wimmers, and K. Bedka, 2009b: Global turbulence and convection nowcast and forecast system. *1st AIAA Atmospheres and Space Environments Conference*, AIAA and Amer. Meteor. Soc., San Antonio, TX, 22-25 June 2009.

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region" - SATCAST (SATellite Convection AnalySis and Tracking) Algorithm Methodology

1. Acquire satellite data (GOES visible, 3.9, 6.5, 10.7 and 13.3 μm)
2. Mesoscale Atmospheric Motion Vectors (MAMV) derivation (GOES)
3. Convective Cloud Mask (CCM) generation (pre-convective/immature clouds)
4. Cloud Object Tracking (OT) (via MAMVs and CCM)
5. Interest Field (IF) (Spectral & Temporal differencing tests) calculations
6. CI prediction (≤ 2h) determination (threshold: 5/6 IF critical thresholds met)

CI Interest Field	Critical Value
10.7 μm BT	0°C
10.7 μm BT time trend	$\leq \left(\frac{-4°C}{15\ min}\right)$
6.5-10.7 μm BT difference	$-35\ °C\ to -10\ °C$
13.3-10.7 μm BT difference	$-25\ °C\ to -5\ °C$
6.5-10.7 μm BT time trend	$> \left(\frac{3°C}{15\ min}\right)$
13.3-10.7 μm BT time trend	$> \left(\frac{3°C}{15\ min}\right)$

Walker, J. R., W. M. MacKenzie Jr., J. R. Mecikalski, 2011: An Enhanced Geostationary Satellite-based Convective Initiation Algorithm for 0-2 Hour Nowcasting with Object Tracking, Submitted: Journal of Applied Meteorology and Climatology (DRAFT)

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

Objective 1: SATCAST Convective Initiation (CI) Algorithm Improvement

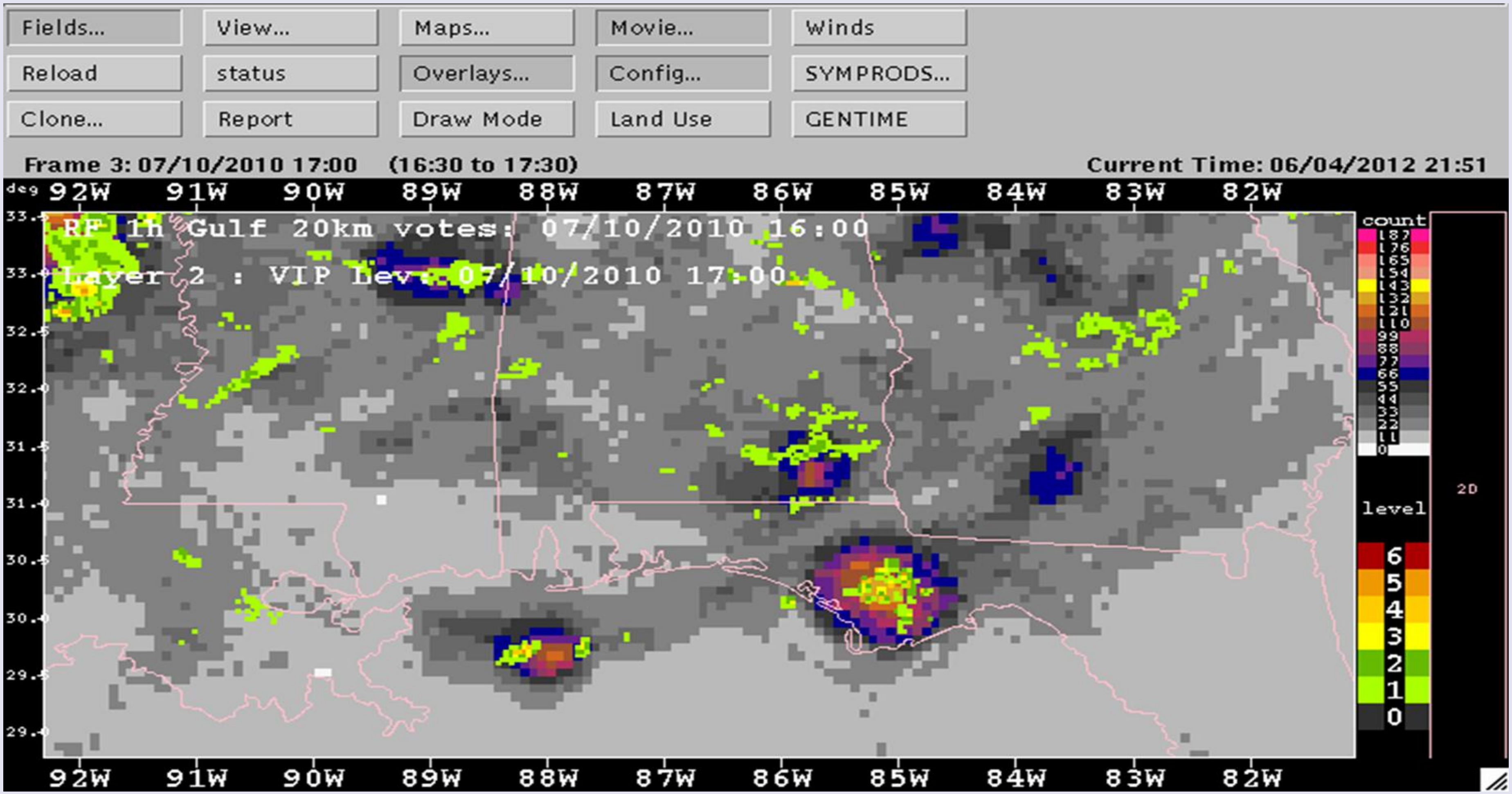
SATCAST output (including interest field parameters) into Random Forest (RF) framework to determine most important interest field parameters correlated to CI; use results to develop a more robust CI algorithm.

Objective 2: RF Analysis to Enhance over Ocean CI prediction

Input SATCAST output, NWP variables, other satellite fields, radar data; use RF to rank variables in terms of importance to CI; train/calibrate ensemble of decision trees to predict CI.

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

Objective 2: Example (Dr. John Williams & David Ahijevych, NCAR)



Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

Objective 3: TANN Improvement

Performance enhancement of TANN envisaged via the following:

- Utilize RF to determine relative importance of TANN input variables; easier comparison to NCAR RF models (objective 2)
- Modify TANN-MOS to include SST data as input (TANN-MOS-SST). Compare TANN-MOS and TANN-MOS-SST performance
- Modify TANN-MOS framework above to input SATCAST output (SATCAST validation dataset used to determine target)

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

What is a Random Forest? (Breiman, 2001)

• A Forest of Decision Trees
• Objective: Reduce instability (instability: small changes in training data→ large changes in classifier performance), and thus variance and prediction error, of decision trees
• Bootstrap Aggregation ("Bagging") is used to reduce the instability/variance of the final classifier/estimator

Breiman, L., 2001: Random Forests, *Machine Learning*, **45**, pp. 5-32.

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

What is a Random Forest? (Breiman, 2001)

Algorithm

- Consider data set [X,Y] X=Input (features) Y=Target
- Draw Z bootstrap samples from a random subset of original data set
- For each bootstrap sample q generate a decision tree
- For each node in the decision tree, choose random subset m of M features
- At each node, determine the optimal split of each variable based on subset m
- The foregoing results in a decision tree based on bootstrap sample q
- Repeat for every bootstrap sample, thus creating a forest of decision trees
- To create the classifier, apply input vector X down the Q trees in the forest
- For each input case, each tree votes for the class
- We classify each input instance into the class with the most votes over all trees Q

Breiman, L., 2001: Random Forests, *Machine Learning*, **45**, pp. 5-32.

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

What is a Random Forest? (Breiman, 2001)

Determining the importance of each feature/variable m

• Calculate performance on the instances not used in the q-th decision tree (out-of-bag data set) (A)

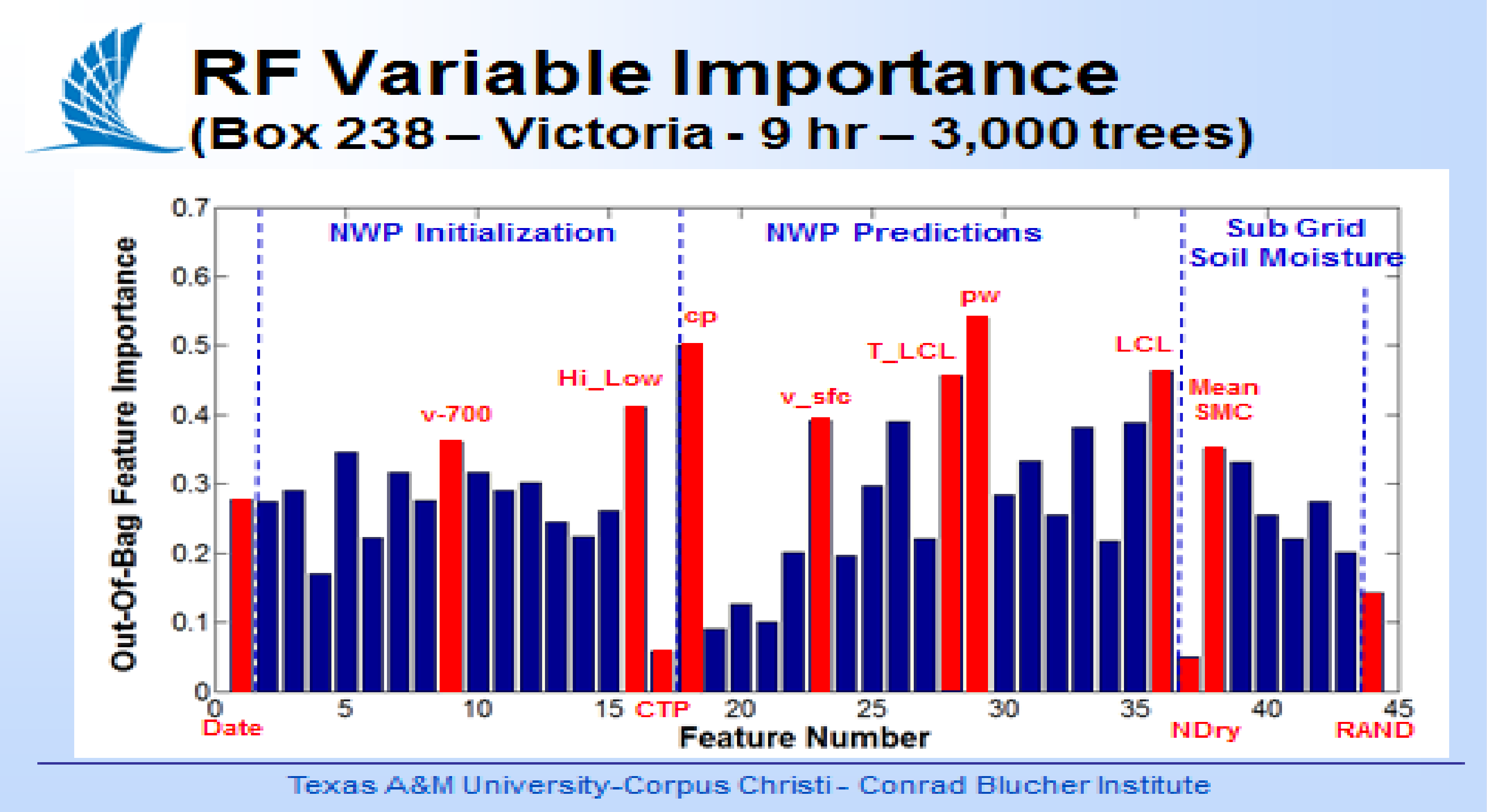
• Randomly permute the m-th feature in the out-of-bag dataset, then apply to the q-th decision tree again (B) to calculate performance

• Calculate A-B and then average over all decision trees that contain feature m (to obtain raw importance score) and standardize.

Breiman, L., 2001: Random Forests, *Machine Learning*, **45**, pp. 5-32.

Project #2: "Improved Convective Initiation Forecasting in the Gulf of Mexico Region"

Objective 3: Example



The End. For those still awake, Questions???