ROOT ANN Interface for RunII

P. Koehn, C. Ciobanu,
R. Hughes, B. Winer

thinkshop2

Nov. 11, 2000

- Motivation
- ANN review
- ROOT interface (example)
- Summary

Motivation

- The use of ANN's is popular.
  - D0 has used them extensively in studying ttbar decays.
  - Top and Higgs analyses in Run II
  - Today at the Weak Interactions Discussion Group:
    - Presentations of Neural Net studies by L. Dudko and C. Ciobanu
- Many use **JETNET**
  - Package (from CERN and University of Lund) that performs training (minimization) and testing of feed-forward ANN.
  - Specifically aimed at HEP.
  - Anonymous ftp site: **thep.lu.se**

Goal

- To make a ROOT interface to JETNET that is simple, fast, and flexible.

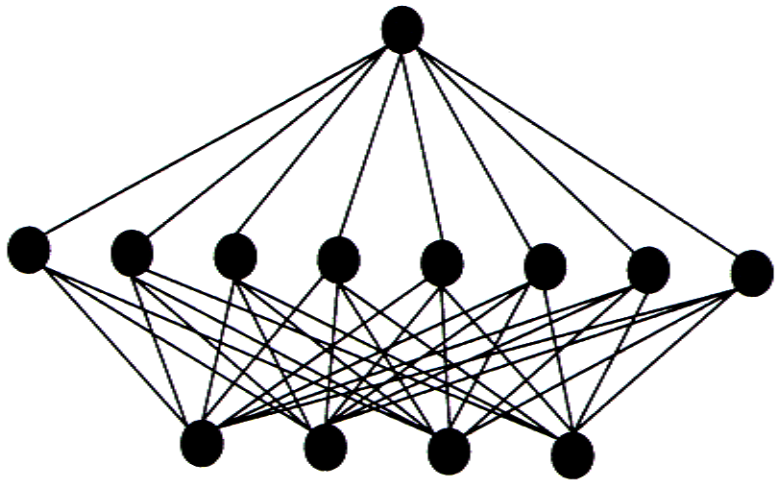
- Useful JETNET and ANN links:
  - www.thep.lu.se/public_html/jetnet_30_manual/jetnet_30_manual.html
  - html://neuralnets/web.cern.ch.NeuralNets/nnwInHep.html
  - www-dapnia.cea.fr/Spp/Experiences/OPAL/opalcern/nnrecipe.html

ANN as a Multivariate Classifier

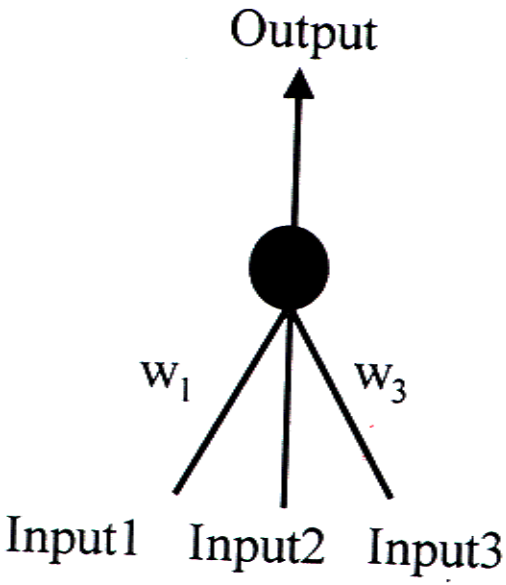
- We want to use the ANN to classify signal and background events.
- The events are assigned (eg. in the single output case):
  - 1 for signal
  - 0 for background
- Discrimination is achieved by looking at the output determined by the ANN.
- Technique exploits the correlations among variables, and provides discrimination between signal and background.

Artificial Neural Networks

- ANN is a function of N variables
  - Useful graphical representation:



  - All nodes above the input layer perform a simple calculation:



$$\text{Output} = g\left(\sum_i w_i \cdot \text{Input}_i - \text{Threshold}\right)$$

A common activation function is: $g(x) = \tanh(x)$

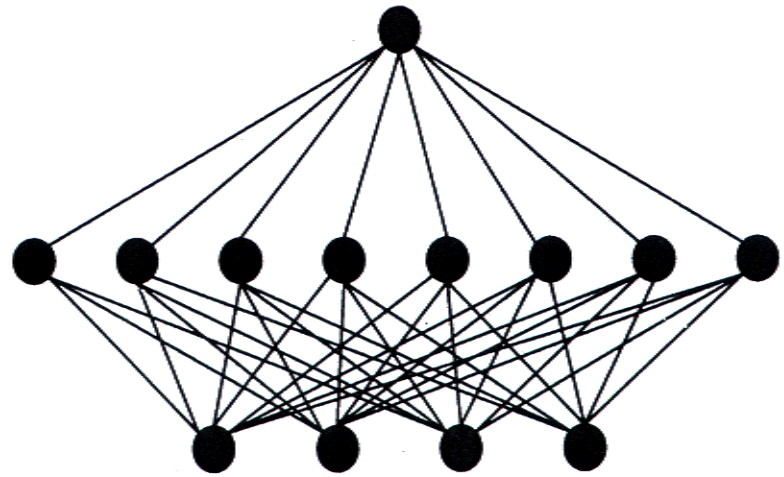
ANN
Training and Testing

- Behaviour of ANN determined by weights.
- **Training** process adjusts the weights:
  - Present ANN with input **patterns** (a group of variables from a single event).
  - Compare ANN Output with what is expected (Target).
  - Adjust the weights by minimizing a mean square error function.

$$E = \frac{1}{N_P} \sum_{p=1}^{N_P} \sum_{i=1}^{Out} (O_i^{(p)} - T_i^{(p)})^2$$

  - Repeat.

- The performance of the ANN is **Tested** by presenting independent events to ANN, and finding how often it gets the right answer.



ROOT Interface

- The ROOT Interface consists of two parts:
  - **ROOT script (root_to_jetnet.C) uses command line to:**
    - **Set ANN parameters (file).**
    - **Run the ANN**
    - **Plot input variables, NN performance, error, and output**
    - **Users will modify this**
  - **JETNET .exe (FORTRAN)**
    - **Expects ANN parameter file**
    - **Performs the Training & Testing using JETNET subroutines**
    - **Creates performance and weight files.**
    - **User would not have to touch this.**

Using the ROOT Interface

Example: Classify ttbar and W + jet background events.

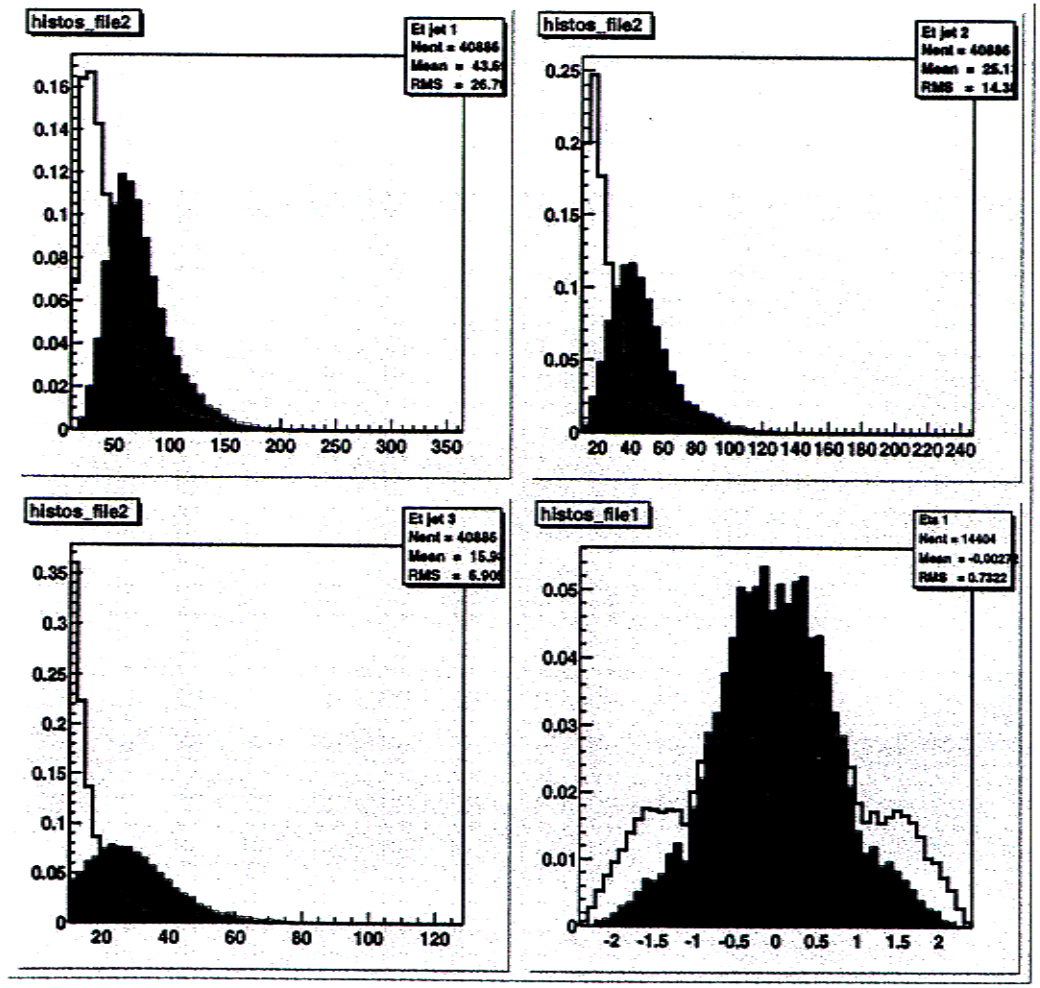
- Input Files
  - Global set of all possible inputs one may choose from
  - Generated by user.
  - 1 file per sample (signal, bkg1, bkg2, ...)
  - Rows (Events) and Columns (Pattern Variables)
- Set the ANN Parameters.
  - eg. choose input variables, how many events to train on, number of training cycles, number of hidden nodes, etc...
- Run JETNET
- Plot the results:
  - Performance, error, ANN output
- Use the net (apply ANN to data).

Et jet 1
Et jet 2
Et jet 3
Eta 1
Eta 2
Eta 3
Met
Ht
Njet

54.65887451	40.88462067	23.49944496	-1.06890965	-1.96102273	-0.79334229	49.09893036...
36.25290298	30.39608765	28.32822800	-1.17113495	-1.06486785	-1.40596402	32.69572830...
140.77398682	28.88099861	21.95326996	-0.18924452	-0.79315042	-0.07041404	124.64159393...
96.64276123	68.13753510	60.78130341	0.52958304	0.03265952	-1.33528435	43.20137024...
43.22183609	35.94001389	25.91762161	0.55198044	-0.49529329	-0.27864811	55.30059433...
151.61328125	69.37616730	55.84277344	-0.56662613	-0.58165526	-0.06336756	41.51983643...
81.05252075	26.22723579	17.36202240	-0.44931754	0.78753597	-0.34974739	42.99569321...
103.23712158	75.03862000	60.64249420	-0.51652193	1.77900207	-1.20707572	62.73186111...
65.72234344	18.69539452	15.88183308	-0.41062140	1.20873225	0.31330562	36.68163300...
44.71592331	28.77424622	23.04918861	-1.04198885	1.08706748	-1.92912865	59.88166809...
51.25721359	43.08668137	27.27217293	0.85083866	1.12633574	-0.70891237	38.99965668...
40.56472778	33.02461624	28.44169617	-1.56939483	-0.41746897	0.07960255	29.67476654...
76.32497406	64.35749817	50.71455765	1.03300512	0.31696343	0.18754163	54.43646240...

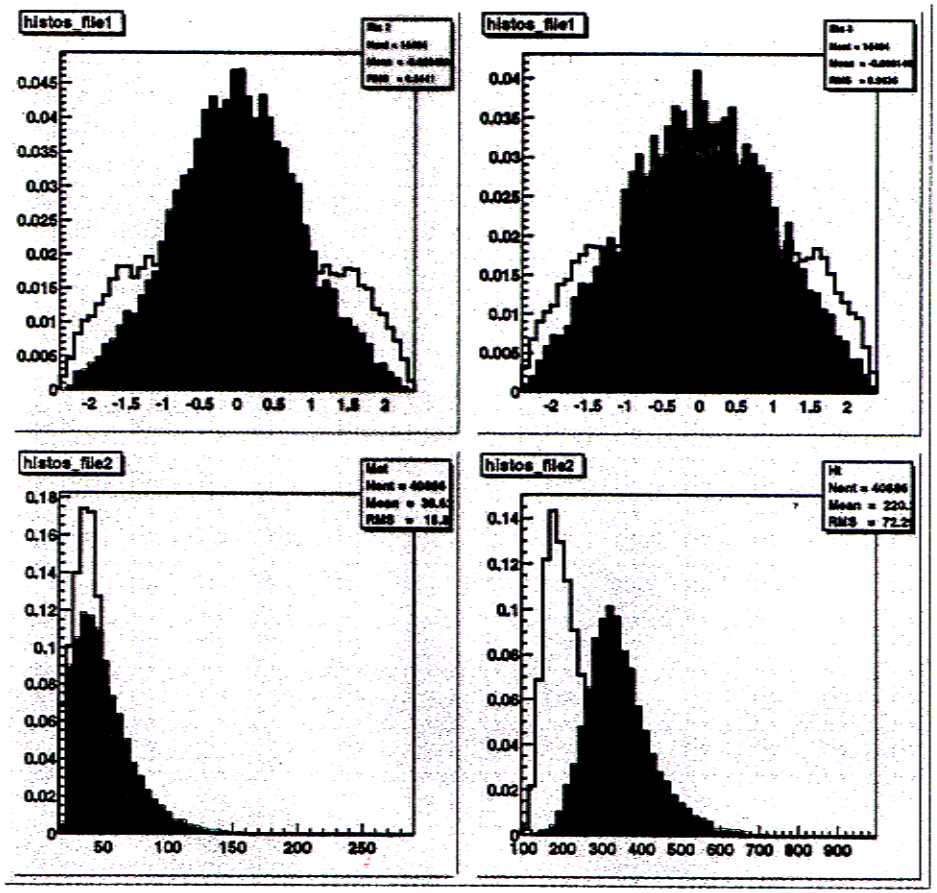
ROOT Interface
Input Variables to ANN

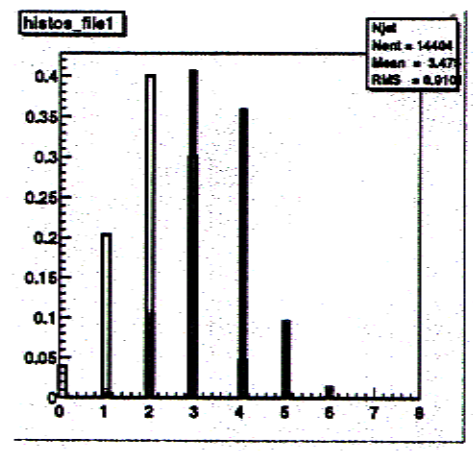
- **Input Samples**
  - Signal - Herwig ttbar 175
  - Background - Vecbos W+njet
- **9 variables:**
  - E_T jet 1,2,3
  - η jet 1,2,3
  - Missing E_T
  - H_T
  - Number of jets with $E_T > 10$ GeV, and $|\eta| < 2.75$
- **User may plot the variables for input samples.**



*ROOT Interface
Input Variables to ANN

P. Koehn
thinkshop2
11-Nov-00





ROOT Interface ANN Parameters

- What are they?
  - Number of (and which)input, hidden, and output nodes.
  - Number of Epochs(training cycles).
  - Minimization technique (normal backprop, Manhattan, others...).
  - Input Samples
    - Number of Training and Testing events
    - Expected ouput pattern(target).
  - Generation of “performance” and/or “weights” files.
- How are they set?
  - Command line “set()” methods to change individual parameters.
  - Or just edit the initial values in the script.
  - getnet() to check the setup.
- Generate a parameter file for the jetnet.exe.

Sample root session...

```
pkoehn /home/pkoehnrun1/NN/myana>root
  ******************************************
  *                                        *
  *        W E L C O M E  to  R O O T      *
  *                                        *
  *   Version   2.24/04      11 July 2000   *
  *                                        *
  *  You are welcome to visit our Web site *
  *          http://root.cern.ch           *
  *                                        *
  ******************************************

CINT/ROOT C/C++ Interpreter version 5.14.37, Apr 29 2000
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.

root [0] .L root_to_jetnet.C
root [1] setEpoch(1000)
root [2] setMini(0)
root [3] getnet()
 ***  Current jetnet parameters ***
 Maximum number of inputs = 9
 Desired input pattern = 111111111
 hidden nodes = 18
 output nodes = 1
 input file 0 /home/pkoehn/run1/NN/myana/topcutnn_top_mc.dat
 Target pattern 1.00
 events = 10000
 training events = 5000
 input file 1 /home/pkoehn/run1/NN/myana/topcutnn_vecbos_bkg.dat
 Target pattern 0.00
 events = 10000
 training events = 5000
 performance flag = 0
 number of epochs = 300
 minimization = 1
 print spew flag = 0
 output directory = /home/pkoehn/run1/NN/myana/
 jetnet .exe directory = /home/pkoehn/Root_Jetnet/src/
 jetnet parameter file = topcutnn_netspecs.dat
 jetnet performance file = /home/pkoehn/run1/NN/myana/p_111111111_9_18_1.dat
 jetnet weights file = /home/pkoehn/run1/NN/myana/w_111111111_9_18_1.dat

root [4] jetnet("111111111")
root [5] plotPerform()
root [6] plotNN(1000,"top_mc.dat","w_111111111_9_18_1.dat")
root [7] plotNN(1000,"vecbos_bkg.dat","w_111111111_9_18_1.dat")
```

Set net parameters.

Run!

make plots of results.

*ROOT Interface
Run the ANN

- Different ways one may want to run:
  - Run in a single shot mode, using current parameters.
    - root[ ] jetnet()
  - Choose different input variables and run again.
    - root[ ] jetnet("110100111")
  - Loop over numbers of hidden nodes.
    - root[ ] jetnet("110100111",9,18)
    - Generates 10 performance and 10 weights files
  - Loop over combinations of both
- Output is a file containing ANN weights
  - Generates a .C function that produces the NN output.
    - to be used in other scripts and/or AC++ code.
  - Plots to check NN output, it's performance as a function of the epoch number (and number of hidden nodes).

ROOT Interface
Plot and Check results

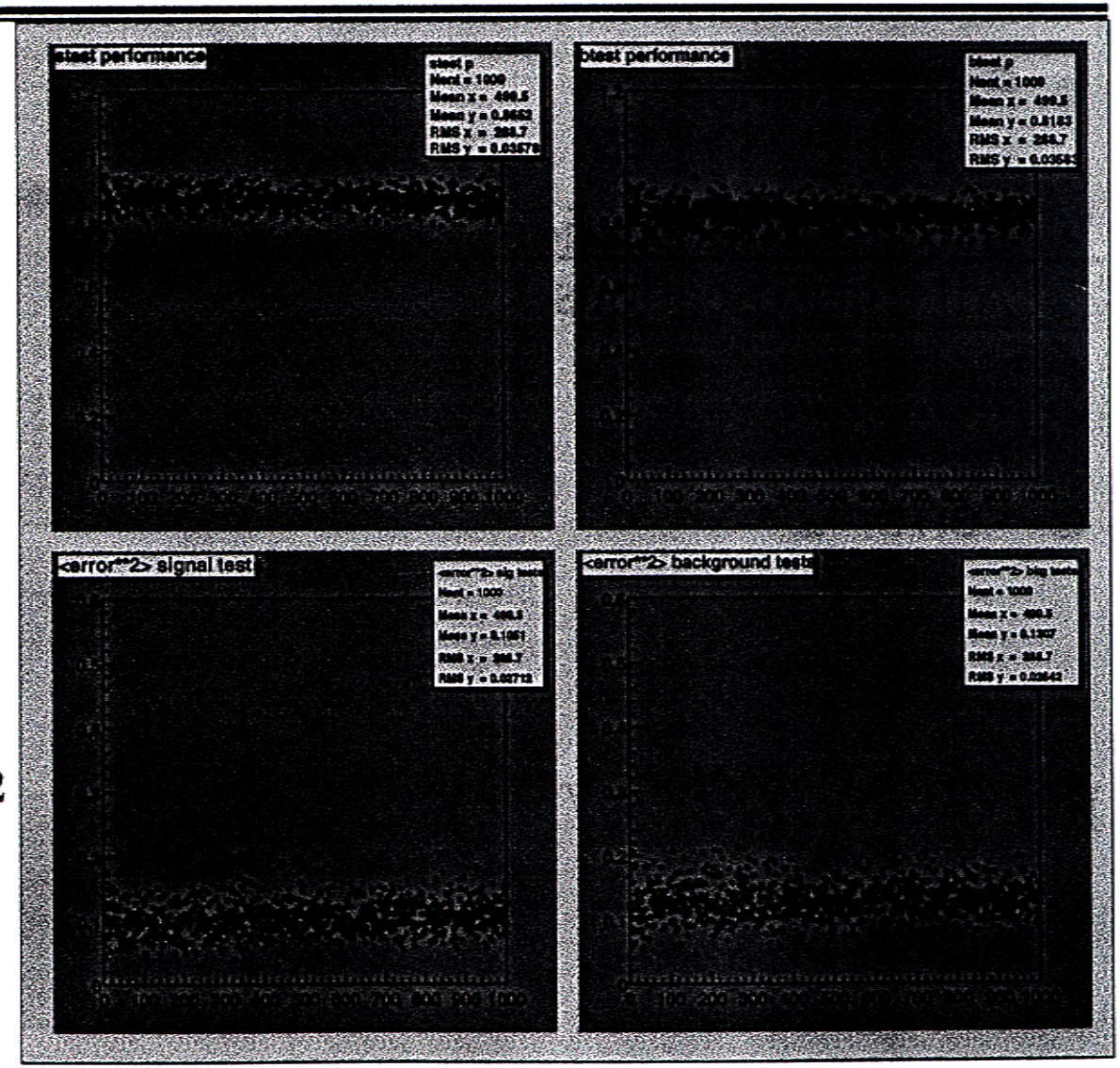
P. Koehn
thinkshop2
11-Nov-00
Page 13

- Performance

"Correct" Classifications
All Classifications

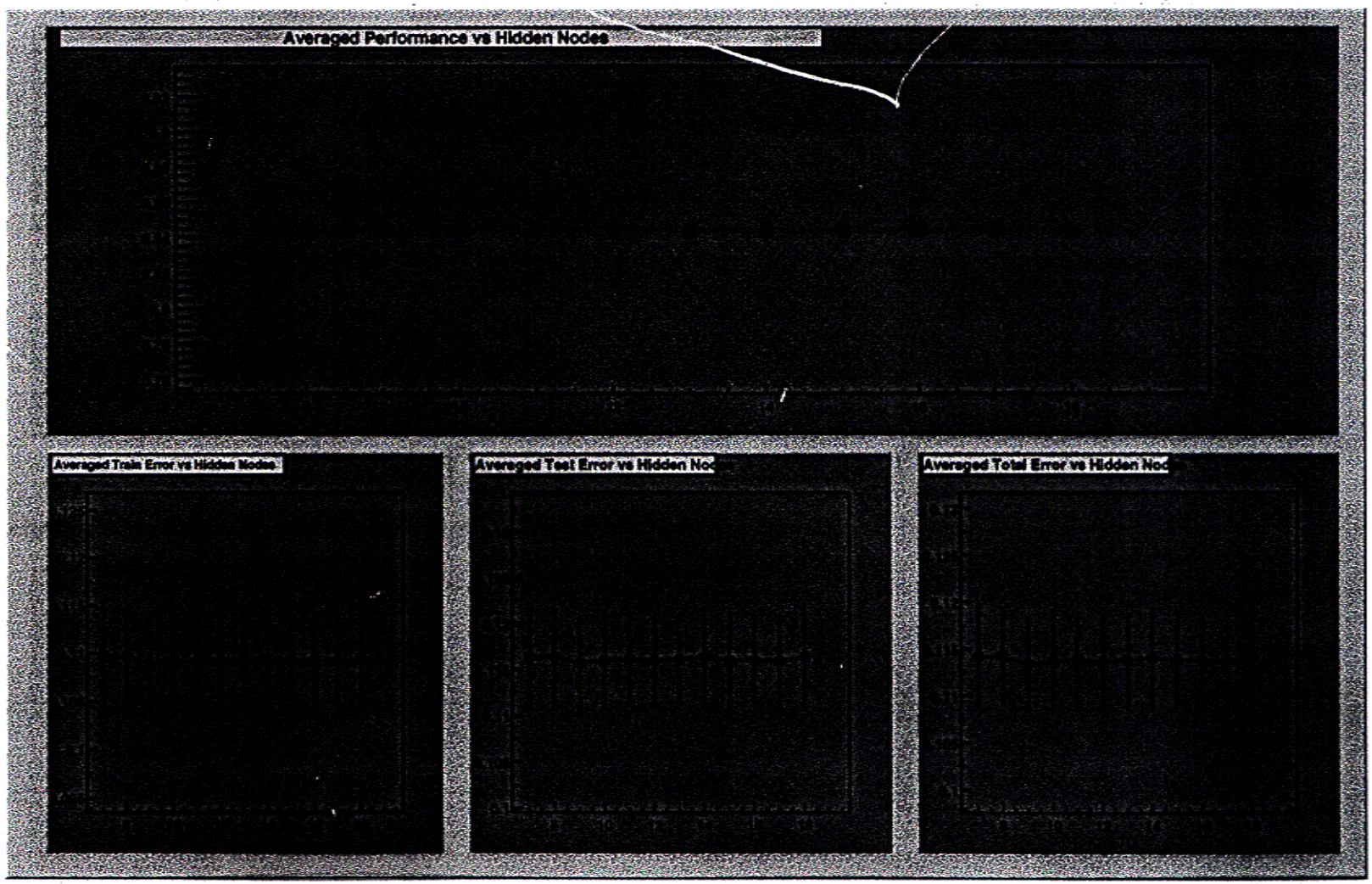
- <error^2>

$$E = \frac{1}{N_P}\sum_{p=1}^{N_P}(O^{(p)} - T^{(p)})^2$$



*ROOT Interface
Plot and Check results

P. Koehn
thinkshop2
11-Nov-00



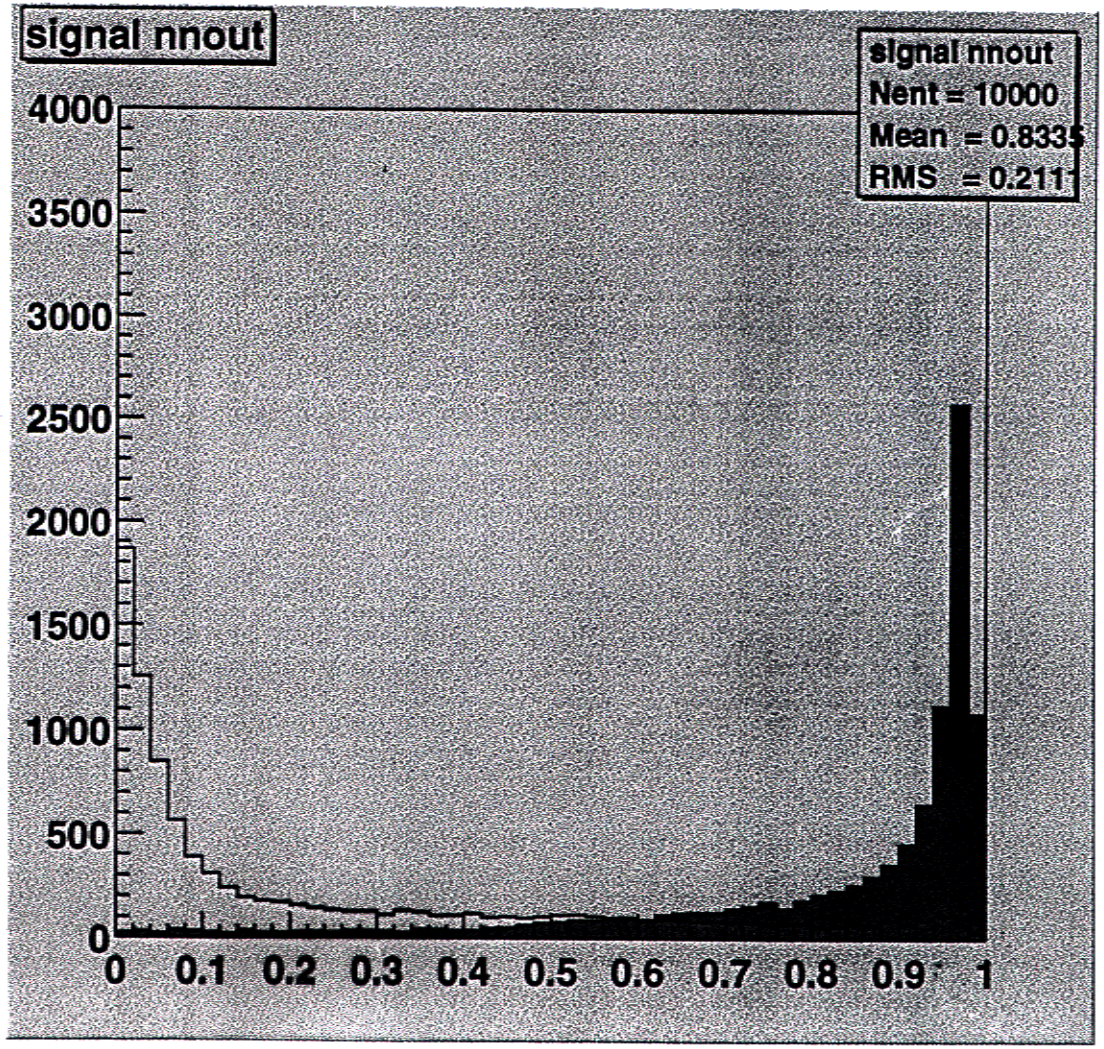
ROOT Interface
Plot and Check results

P. Koehn
thinkshop2
11-Nov-00

- **ANN output**

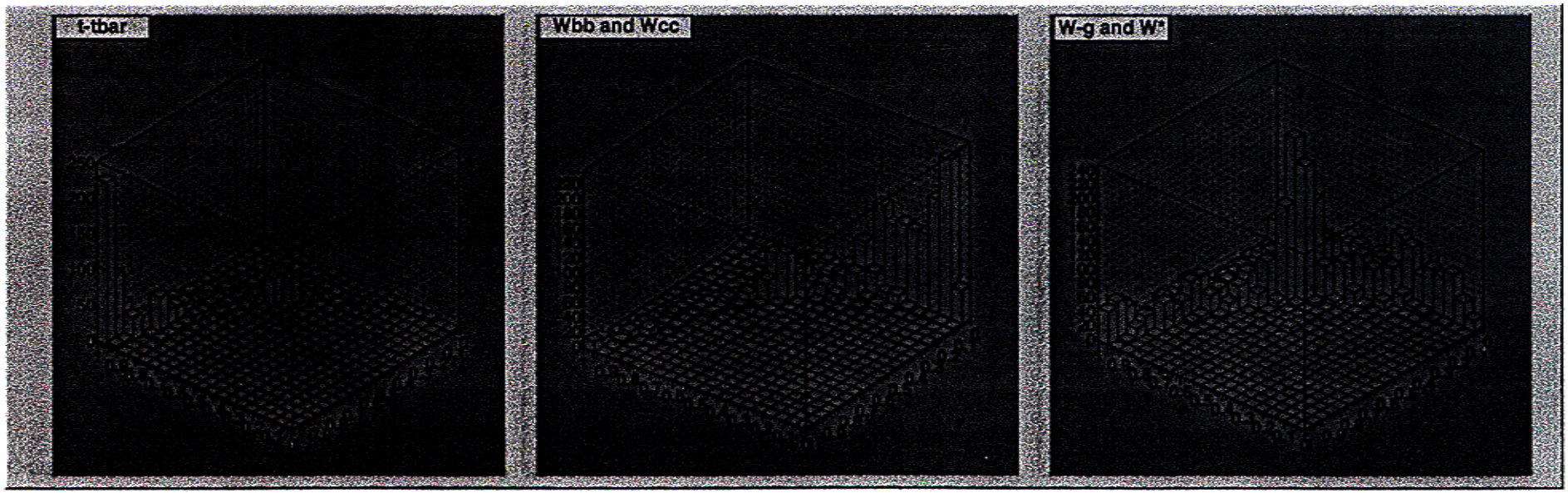
→ Weights file

→ NNout.C



Summary

- We have a ROOT interface that provides a simple command line interface to a JETNET executable.
  - This script allows user to:
    - Vary number and type of input samples.
    - Change NN parameters easily.
    - Vary run modes(single, loop over inputs, hidden nodes, ...).
    - Plot – input variables, NN performance, error, and output.
    - Create weights file to run on the data.
    - .C function based on weights file for general use.
- Future plans
  - Add features:
    - Input from ROOT or Ntuple files.
    - Vary parameters in JETNET minimization techniques.
    - Additional hidden layers.
  - Release for general consumption(in CVS repository at CDF).

*Example using 2 output nodes: Single Top

- 3-layer network
  - Input layer:: 9 input variables => 9 input nodes
  - Output layer:: 2 nodes (minimum to accommodate three classes)
- Require:
  - Out(1) = 0, Out(2) = 0 for Wbb, Wcc
  - Out(1) = 1, Out(2) = 1 for ttbar
  - Out(1) = 1, Out(2) = 0 for Wg, W*