NEPPSR Analysis Project

Kevin Black (Harvard U.)
John Butler (Boston U.)
Stéphane Willocq (UMass Amherst)

New England Particle Physics Student Retreat
Craigville, 13-18 August 2006

Analysis Project

- Goals of the Project

  - Learn basic use of ROOT data analysis tool
    → Standard tool in particle physics

  - Apply statistical analysis to extract physical information
    (particle lifetime, mass, etc…)

  - Discriminate between signal and background events with
    a multivariate analysis technique

Two Analysis Projects

1. B lifetime measurement with likelihood method
2. Discrimination between signal and background with a neural network

Project I: application of maximum likelihood to measure B lifetime, see

http://people.umass.edu/willocq/neppsr/blifetime/AnalysisProject_blifetime.html

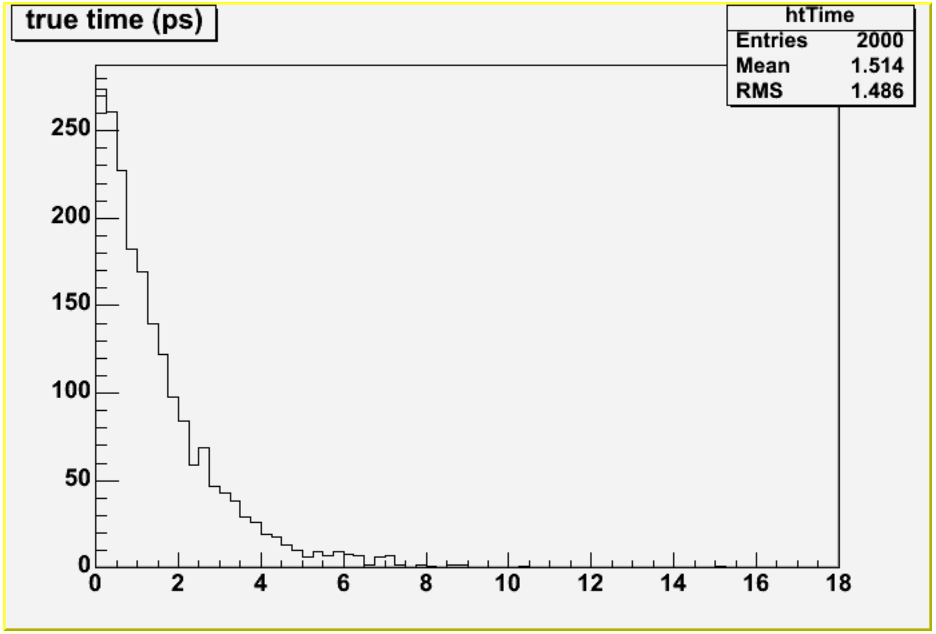
Project I: B Lifetime Analysis

- Sample

  - Simulation of the *exponential* proper time distribution of B mesons

    → Effect of limited resolution in the measured proper time simulated by smearing the true proper time with a constant Gaussian resolution

    → ROOT tuple contains: True time and measured time (in units of ps) for 2000 events



Lifetime used in the generation of the events was 1.532 ps

Project I: B Lifetime Analysis

- Project

  - *Determine the B meson lifetime and its statistical uncertainty* using each of the following methods:

    1) Least-squares fit to the true proper time histogram

    → Need to provide function to fit the distribution with

    f(t) = N exp(-t / tau)

    → Use ROOT built-in interface to do the chi-squared minimization

$$\chi^2 = \sum_{i=1}^{nbins} \left(\frac{\left(f(t_i) - N_i \right)^2}{\sigma_i^2} \right)$$

    2) 'Unbinned' Maximum Likelihood with true proper time

    → Compute and display -log(likelihood) as a function of lifetime

    3) 'Unbinned' Maximum Likelihood with reconstructed proper time

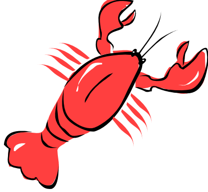
    → Need to determine time resolution

    → Compute and display -log(likelihood) as a function of lifetime

    (use ROOT's TMath::Erfc(x))

Code provided on web site

See Colin Gay's lecture

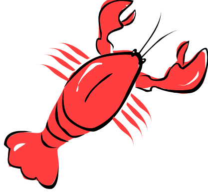


Analysis Project Introduction
Part Deux

John Butler

Boston University

Outline



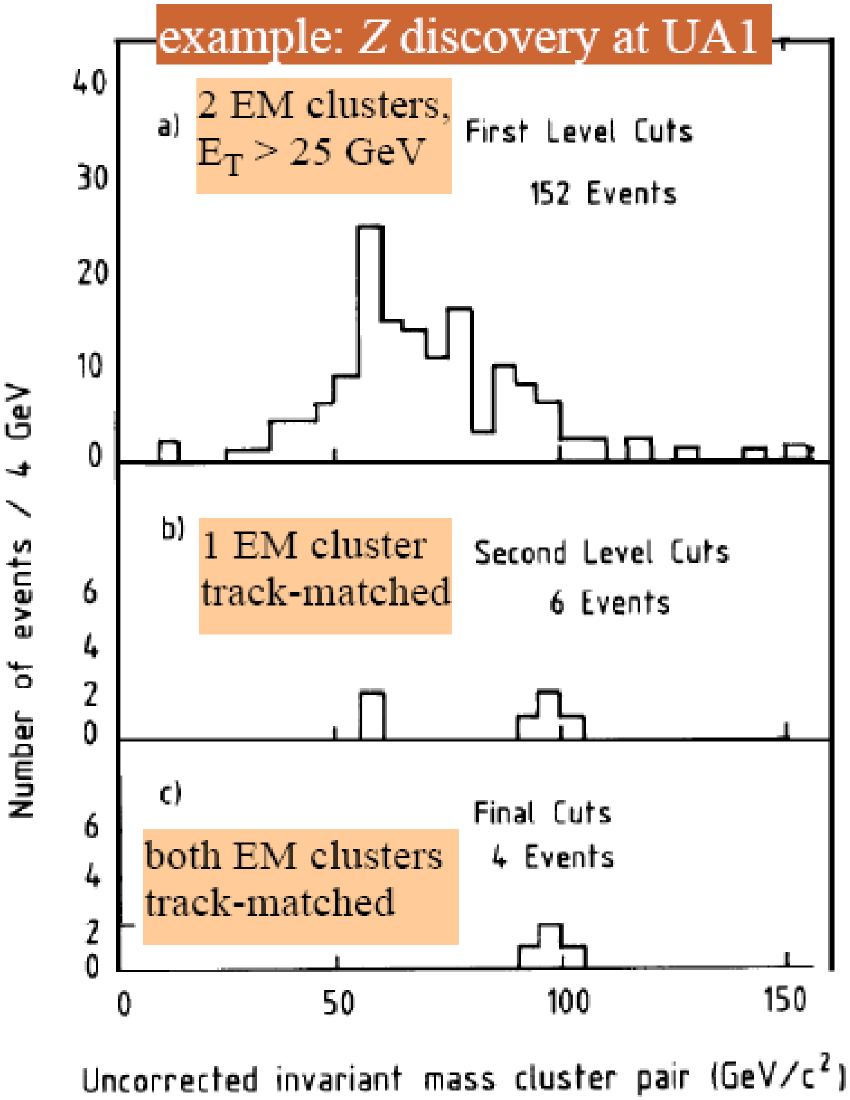
- Overview of old school and advanced analysis methods
- Intro to neural networks
  - Shamelessly steal from an excellent talk by Reinhard Schweinhorst, his full talk is linked from the NEPPSR site
- Analysis Project
  - Searching for new physics at the Tevatron in the Wbb final state
  - Neural net analysis in root using the TMultiLayerPerceptron class
  - Root macros
  - Your job

Warning, warning, warning!
I am not a NN expert
I do not even play one on TV

BOSTON UNIVERSITY

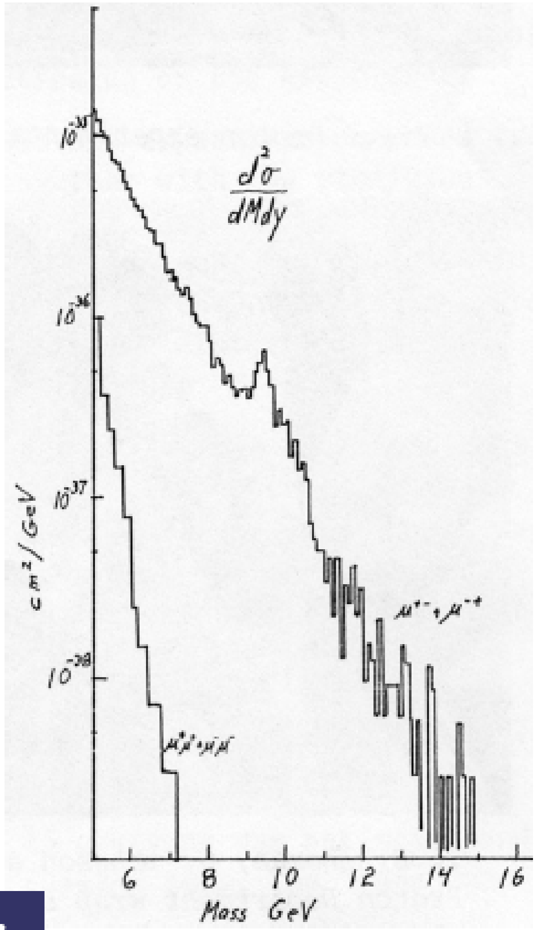
Event counting

- Apply cuts to variables describing the event
  - Object identification
  - Kinematic cuts on objects
  - Event kinematics
- Goal: cut until the signal is visible
  - No background left
  - Or large S/√B
- Sensitive to any signal with this final state
- Requires understanding of background



Peak in a characteristic distribution

- Find a variable that has a smooth distribution for background
  - Typically invariant mass
- Measure this distribution over a large range of possible values
- Look for possible resonance peaks
  - Example: b-quark discovery at Fermilab
- Sensitive to any resonance with this final state

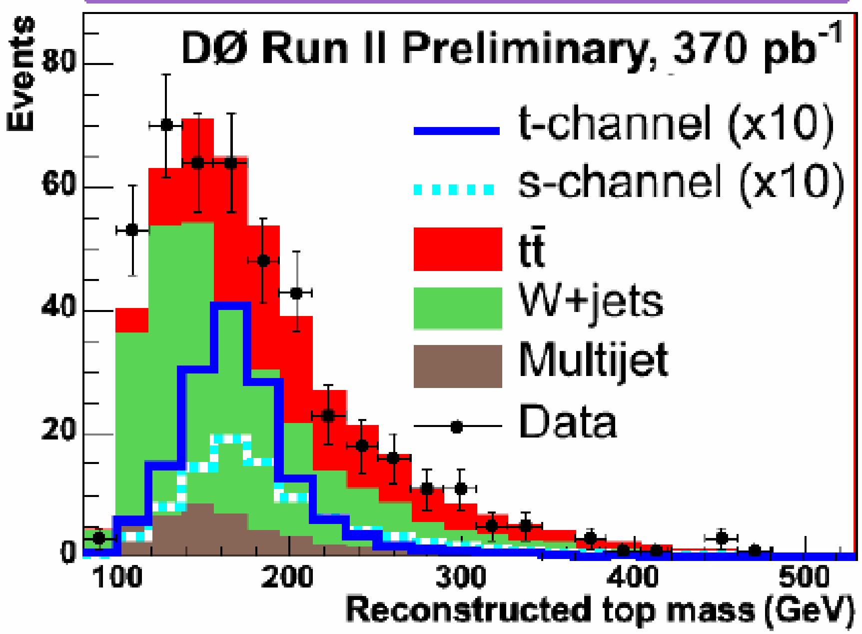


"Bump Hunting"

When Event Counting and Bump Hunting Fails

Example: DØ Single Top in 370 pb^{-1}

	s-channel	t-channel
Signal yield	9.5	15.0
Bkgnd yield	452	
Data	443	
Signal/bkgnd	1:50	1:30

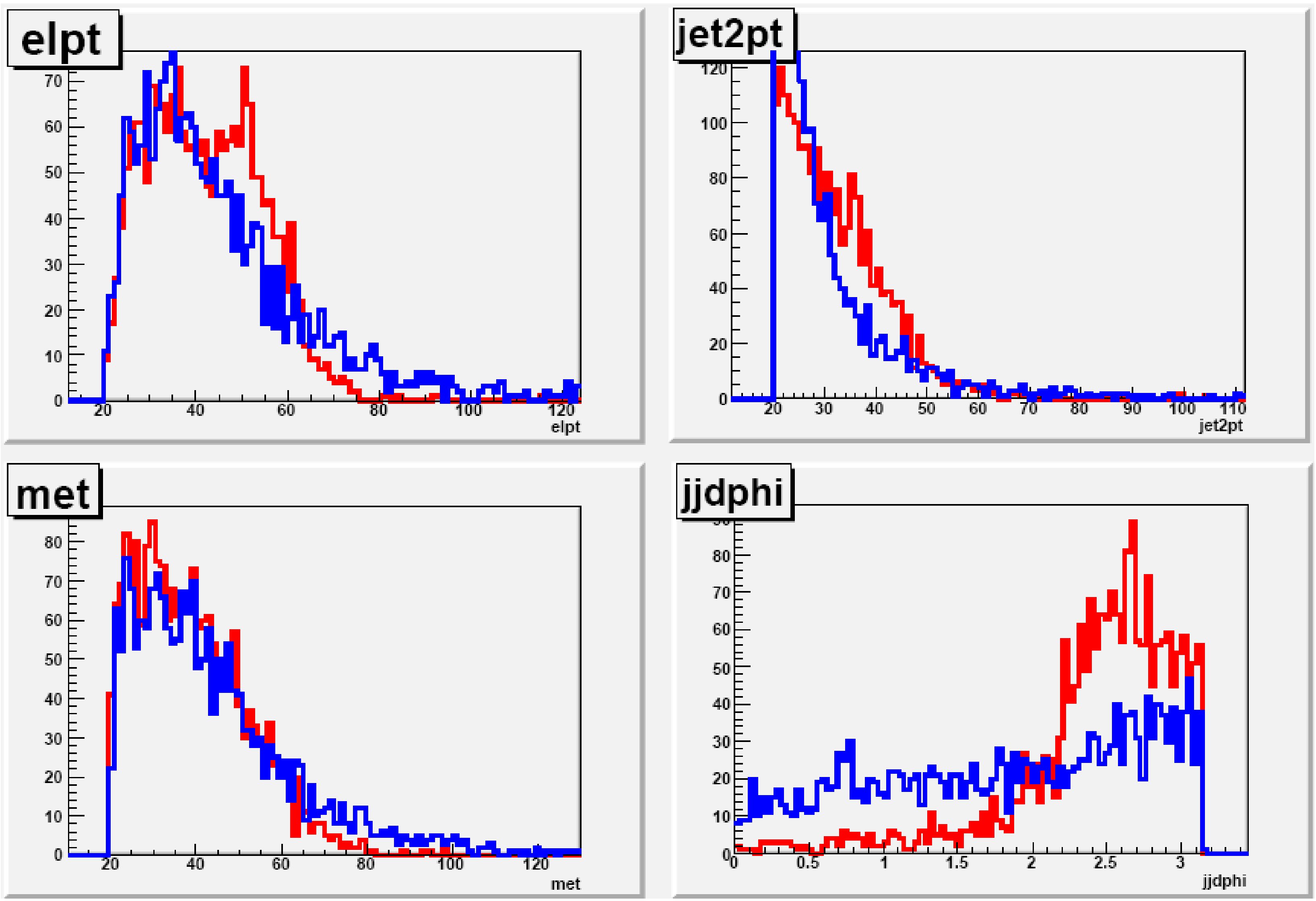


Signal/Background too small for event counting

Invariant mass too broad for bump hunting

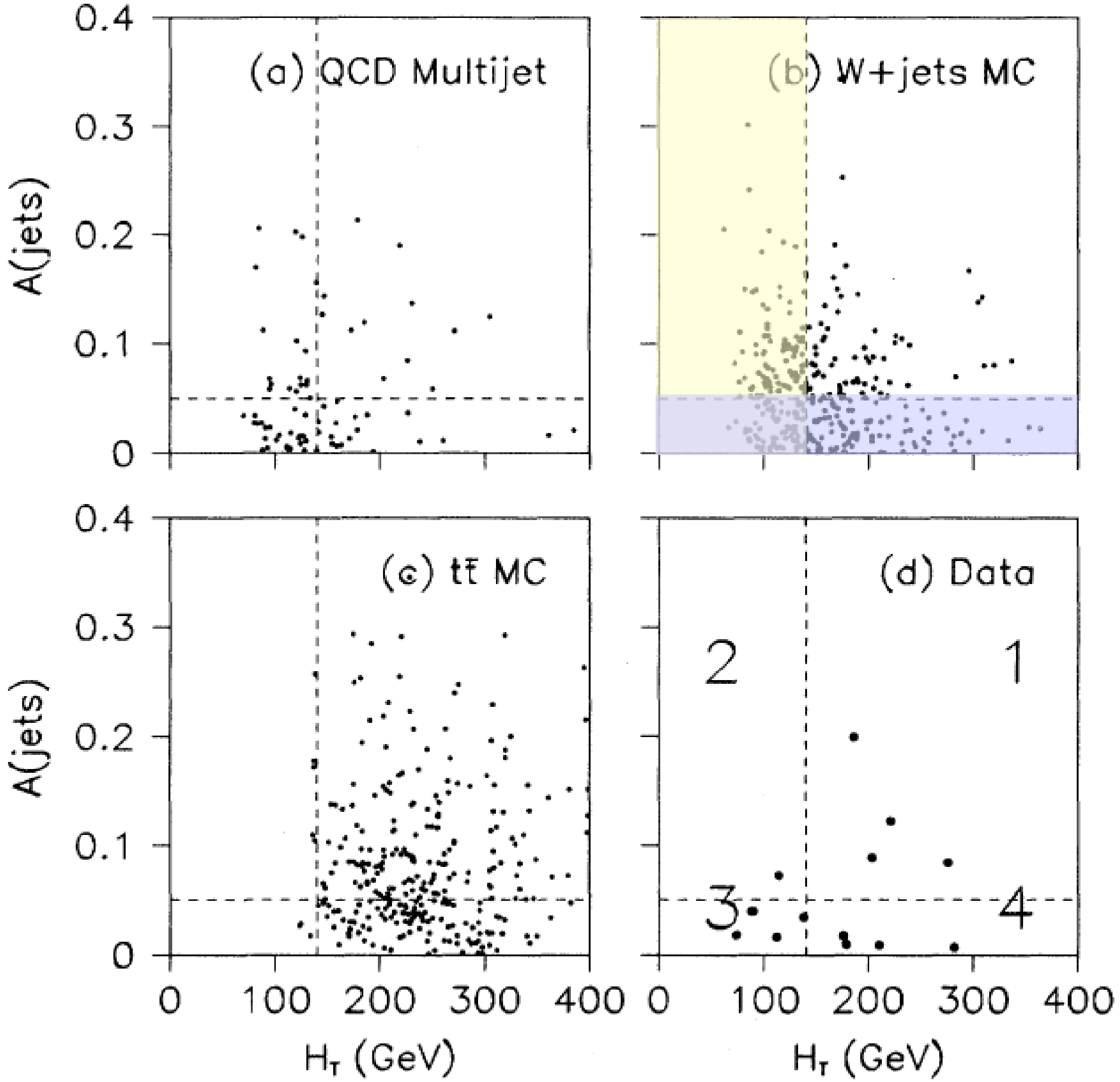
BOSTON UNIVERSITY

When Square Cuts Don't Cut It



BOSTON UNIVERSITY

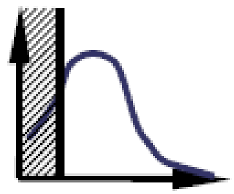
Square Cuts



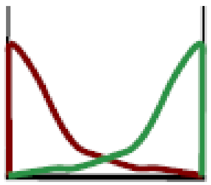
(a) QCD Multijet
(b) W+jets MC
(c) tt̄ MC
(d) Data
A(jets)
A(jets)
H$_T$ (GeV)
H$_T$ (GeV)
0.4
0.3
0.2
0.1
0
0
100
200
300
400
2
1
3
4

Event Analysis Techniques

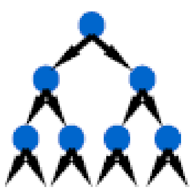
Cut-Based



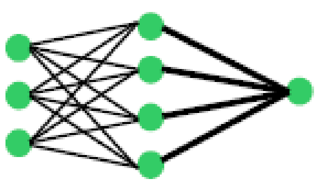
Likelihoods



Decision Trees



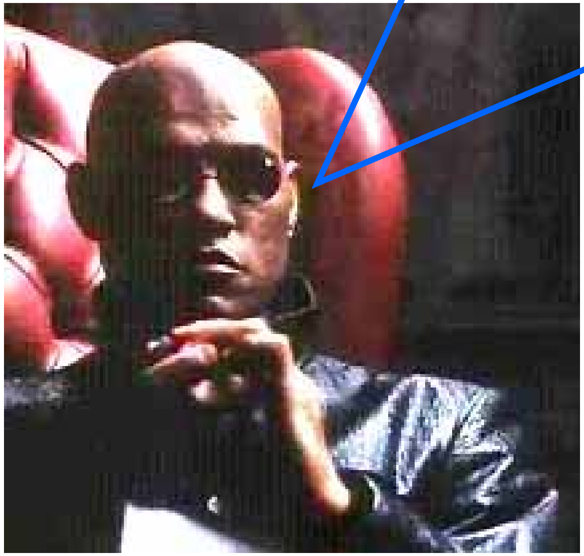
Neural Networks



Many others: Kernel methods, support vector machines, Matrix element, ...

What is a Neural Network?

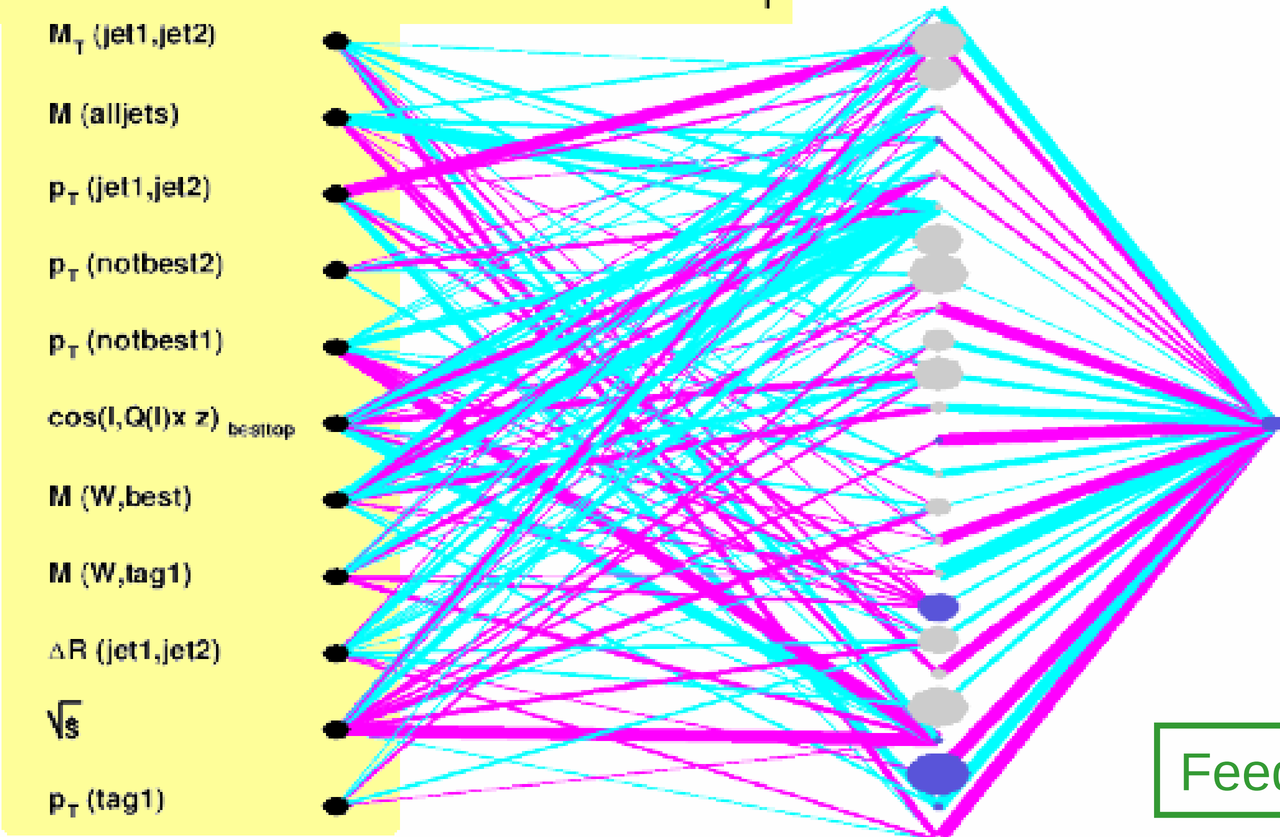
"The question 'What is a neural network?' is ill-posed."
- Pinkus (1999)



In HEP, neural nets are generally used for classification, e.g. S vs B

Neural Networks

Input Nodes: One for each variable x_i



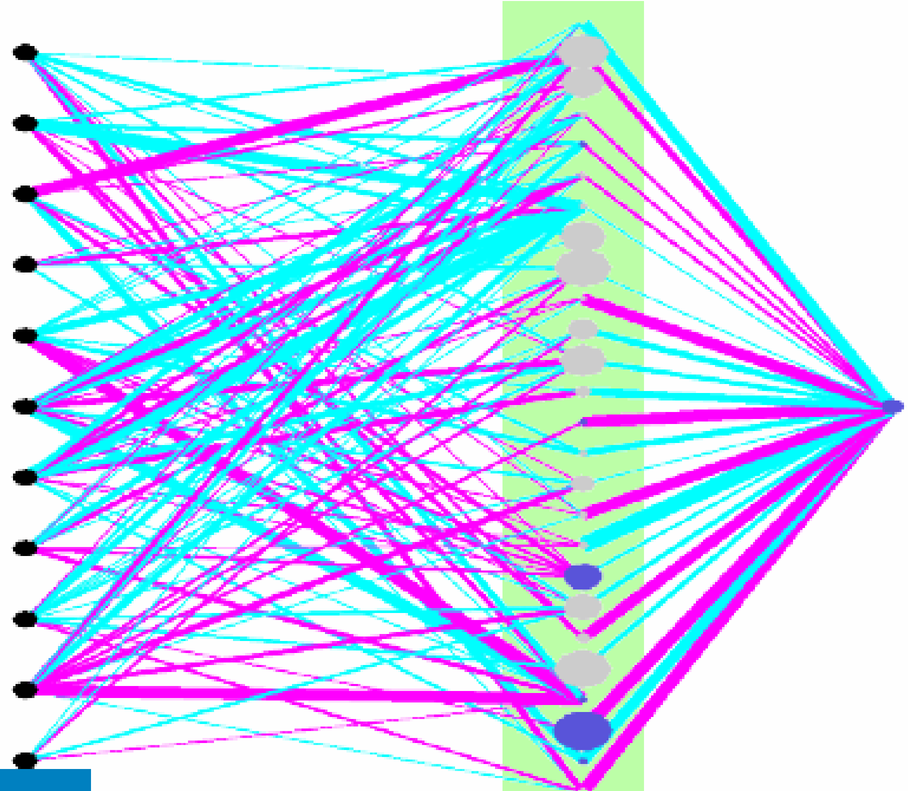
Feed Forward ➔

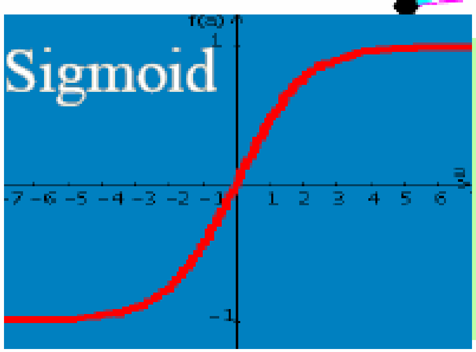
Goal: approximate signal probability

$$f(\vec{x}) \approx P(S|\vec{x})$$

In root, the networks are “multilayer perceptrons”

Neural Networks

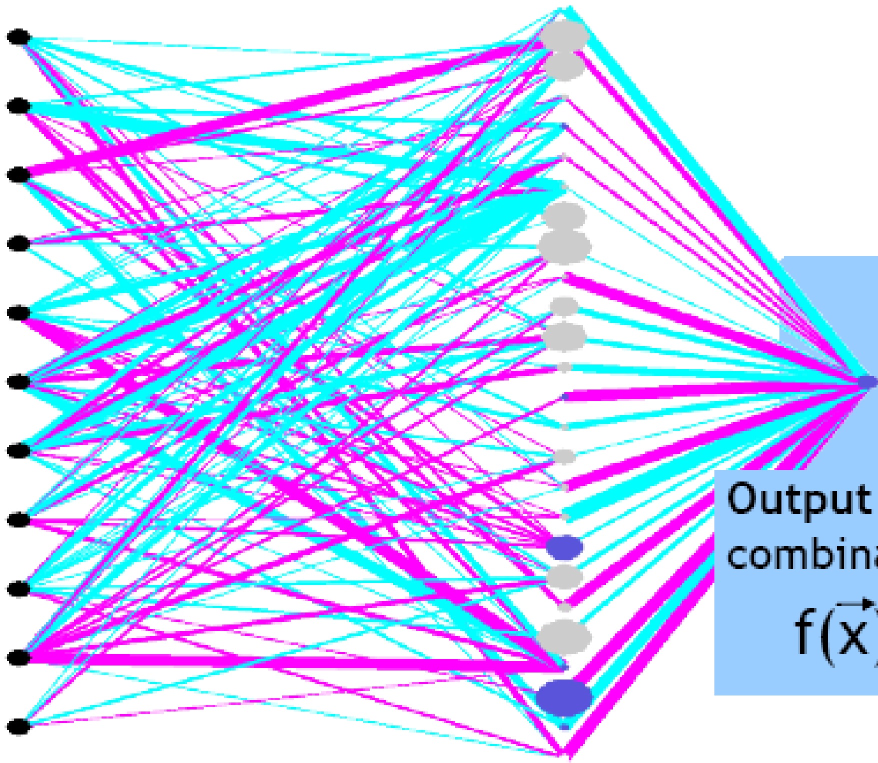




Hidden Nodes: Each is a sigmoid dependent on the input variables

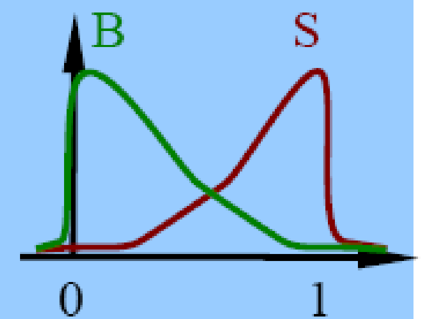
$$n_k(\vec{x},\vec{w}_k) = \frac{1}{1 + e^{-\Sigma w_{ik} x_i}}$$

Neural Networks



Output Node: linear combination of hidden nodes

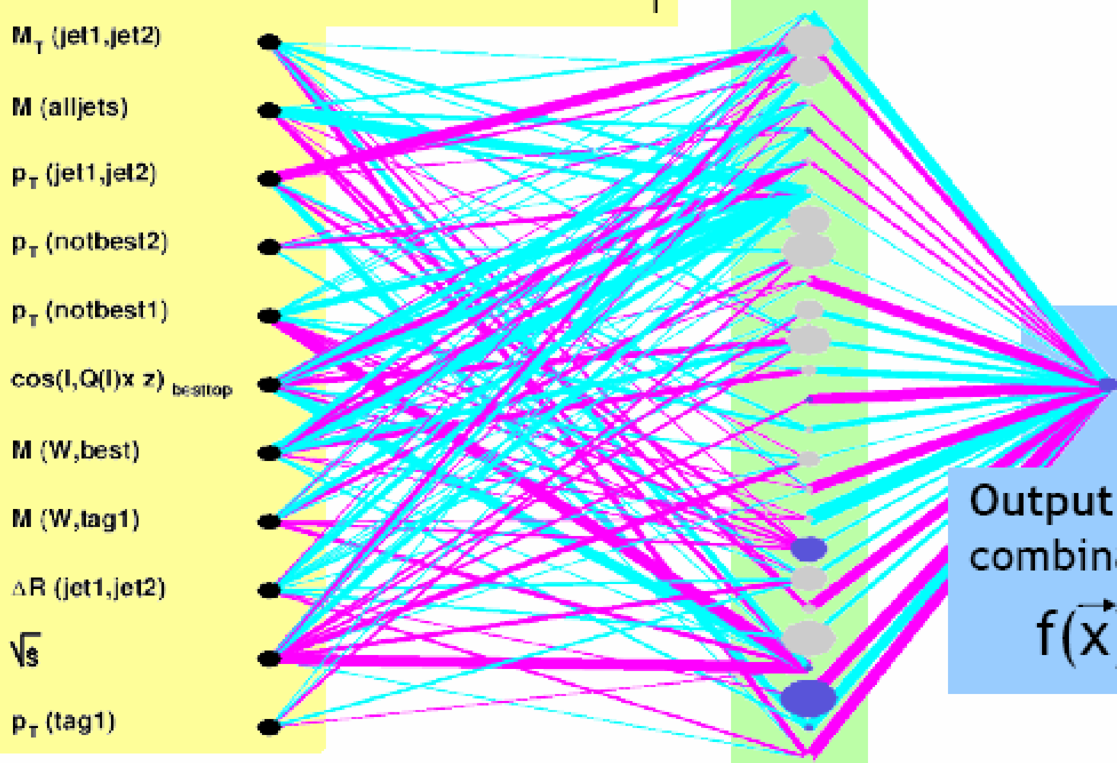
$$f(\vec{x}) = \Sigma\, w'_k\, n_k(\vec{x}, \vec{w}_k)$$



A linear combination of sigmoids
can approximate any continuous function

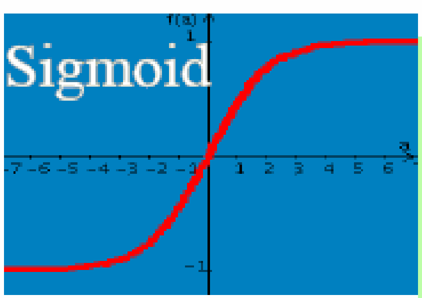
Neural Networks

Input Nodes: One for each variable x_i



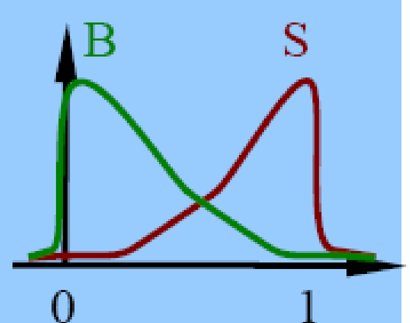
Output Node: linear combination of hidden nodes

$$f(\vec{x}) = \sum w'_k \, n_k(\vec{x},\vec{w}_k)$$



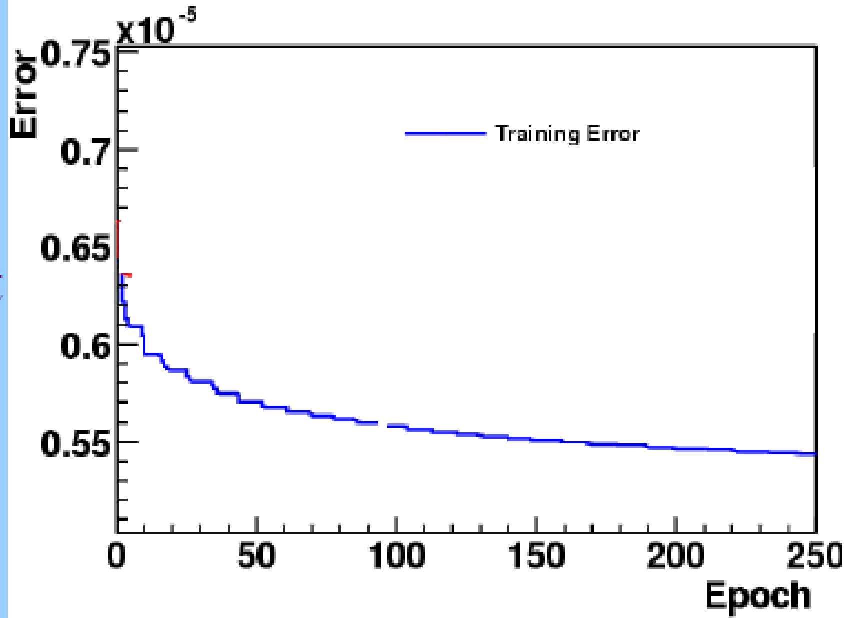
Hidden Nodes: Each is a sigmoid dependent on the input variables

$$n_k(\vec{x},\vec{w}_k) = \frac{1}{1 + e^{-\sum w_{ik} x_i}}$$



Neural Network Training

- Initialize NN weights
- Read in signal and background model events
  - Training sample
- Compute NN error
  - $\Sigma\,(f_{\text{observed}} - f_{\text{expected}})$
- Adjust all NN weights as result
- Compute NN error again
- Repeat until ...

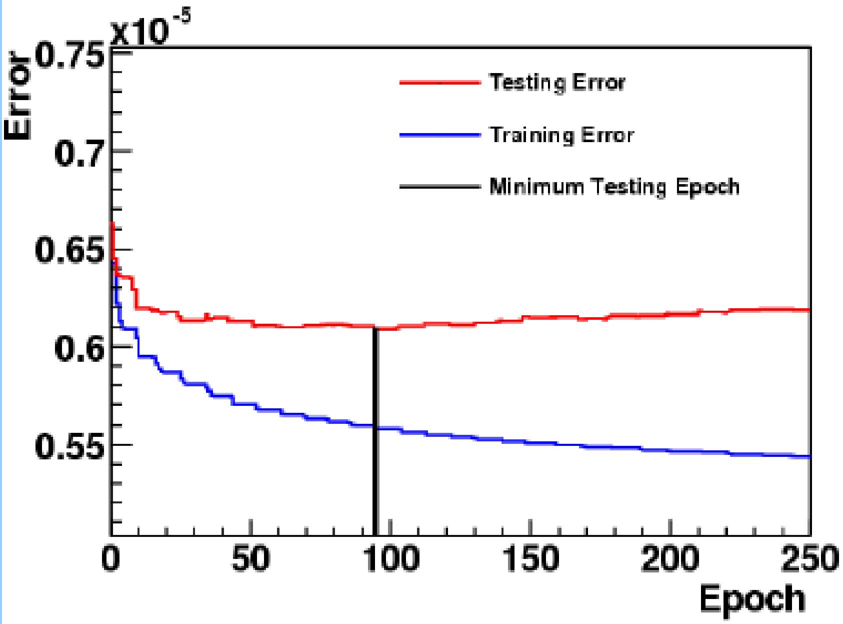


DØ single top search

Reinhard Schwienhorst, Michigan State University 40

Neural Network Training

- Initialize NN weights
- Read in signal and background model events
  - Training sample
- Compute NN error
  - $\Sigma\,(f_{\text{observed}} - f_{\text{expected}})$
- Adjust all NN weights as result
- Compute NN error again
- Apply NN to independent set of signal and background
  - Testing sample
- Stop training when error from testing sample starts increasing

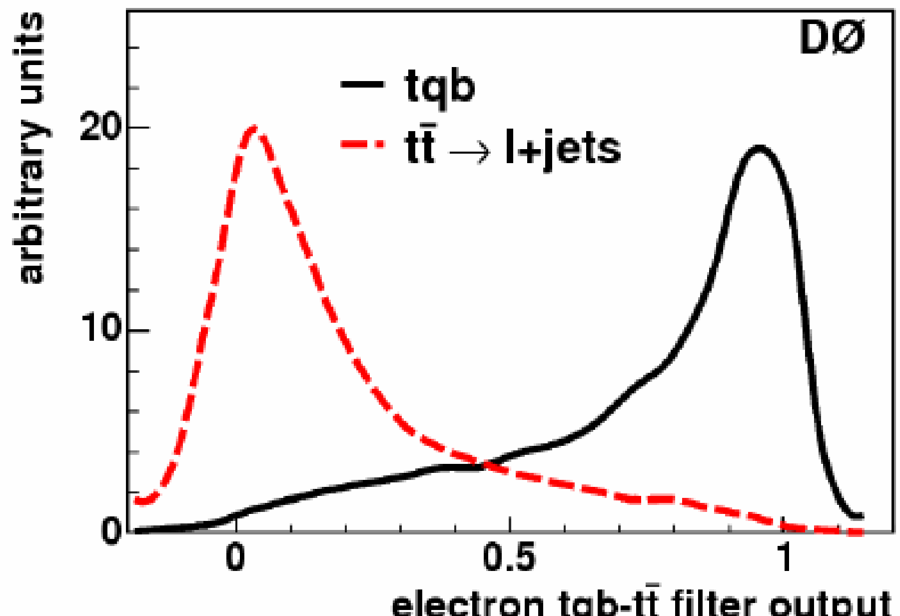


DØ single top search

Reinhard Schwienhorst, Michigan State University 41

Neural Network Result

- Train on signal and background models (MC)
  - Stop when signal-background separation stops improving
    - Independent MC training sample
- For each data event, compute NN output
- Result is almost a probability distribution
  - But not necessarily constrained to [0,1]

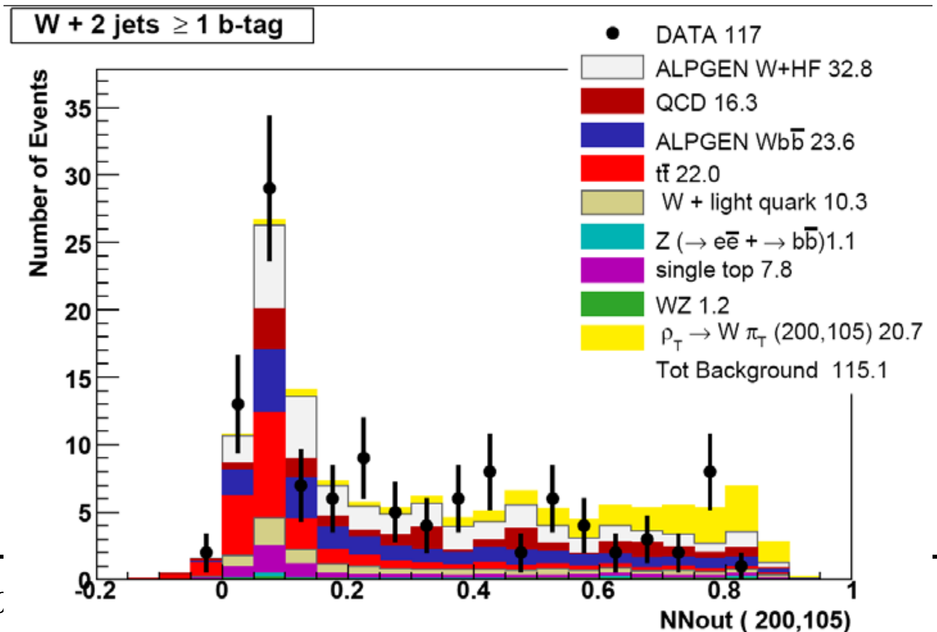


DØ single top search

Reinhard Schwienhorst, Michigan State University 42

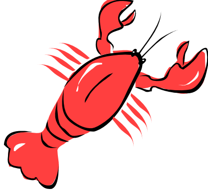
Neural Net Analysis Project

- From the DØ search for new particles called technihadrons
- p pbar → ρ_T → W π_T → (e ν) (b bbar)
  - Just like the search for the Higgs but 20x larger cross section
  - Model parameters: M(ρ_T) = 200 GeV, M(π_T) = 105 GeV
- Signature is a high p_T electron, missing E_T, and 2 b jets
- Several backgrounds processes are important, we will consider only the standard model production of W + 2 jets



DØ- CONF note 5154

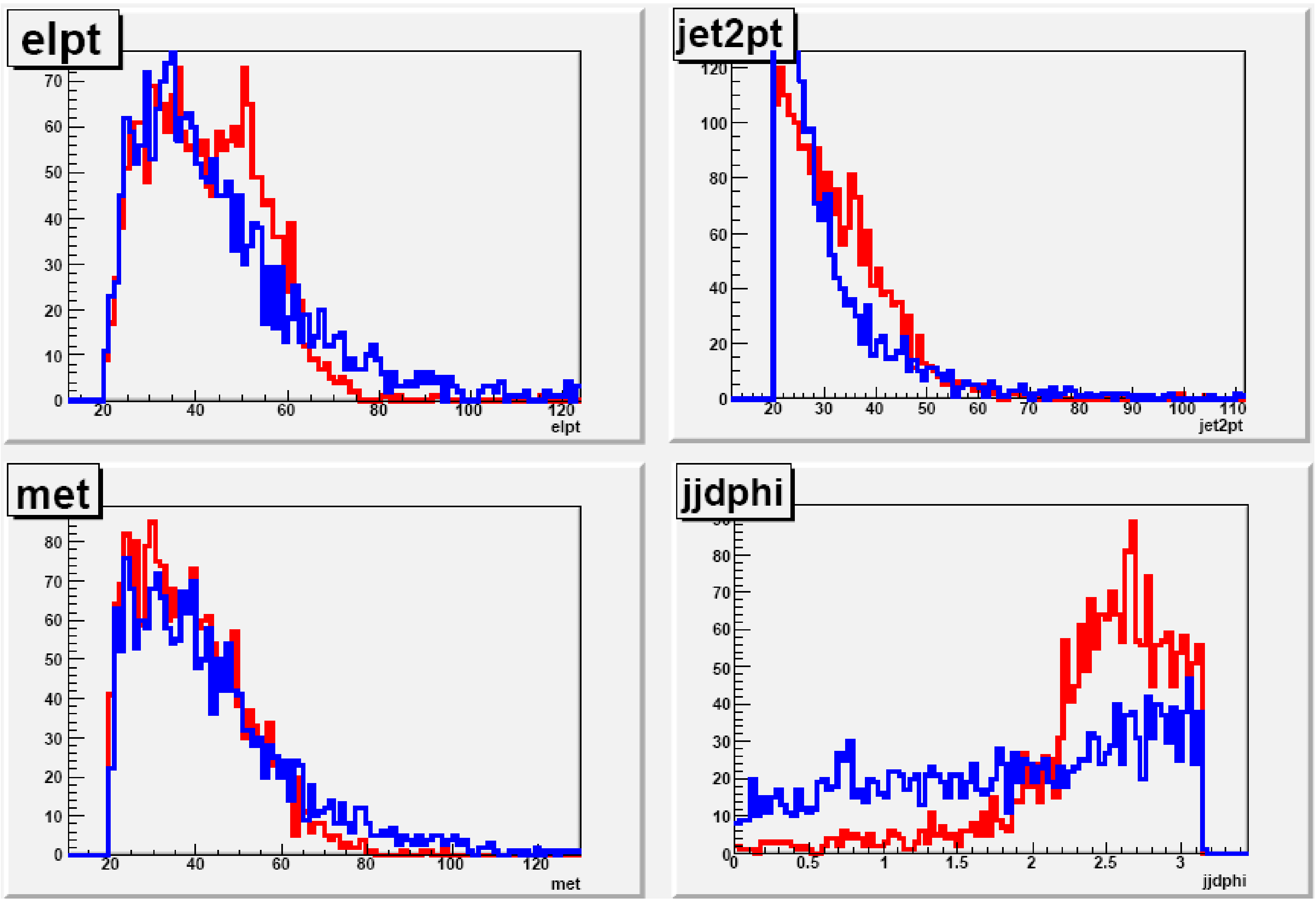
Neural Net Analysis Project



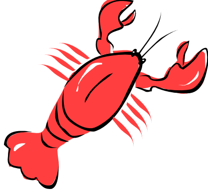
- Training data files
  - Signal: tc_pi0.root
  - Background: wjj.root
- Kinematic variables in the ntuples
  - elpt: Electron p_T
  - jet1pt: p_T of highest p_T jet
  - jet2pt: p_T of 2^{nd} highest p_T jet
  - met: Missing E_T
  - jjpt: p_T of the (jet1,jet2) system
  - hte: $H_T^e = \Sigma\, p_T(\text{jet}) + p_T(e)$
  - jjdphi: $\Delta\phi$(jet1,jet2)
  - meteldphi: $\Delta\phi$(Missing E_T,e)
- Plot these for yourself!
- Play with the network structure to optimize separation of signal and background
  - Variables selected for good signal-background discrimination but not necessarily against Wjj background!
  - You choose which variables to use as input to your network
  - Can have as many hidden layers and nodes as you want
- Challenge data files
  - mystery_n.root where n=1-4
- Unknown fraction of signal and background, up to you to determine

BOSTON UNIVERSITY

Sample of ntuple Variables



root Macros for Project

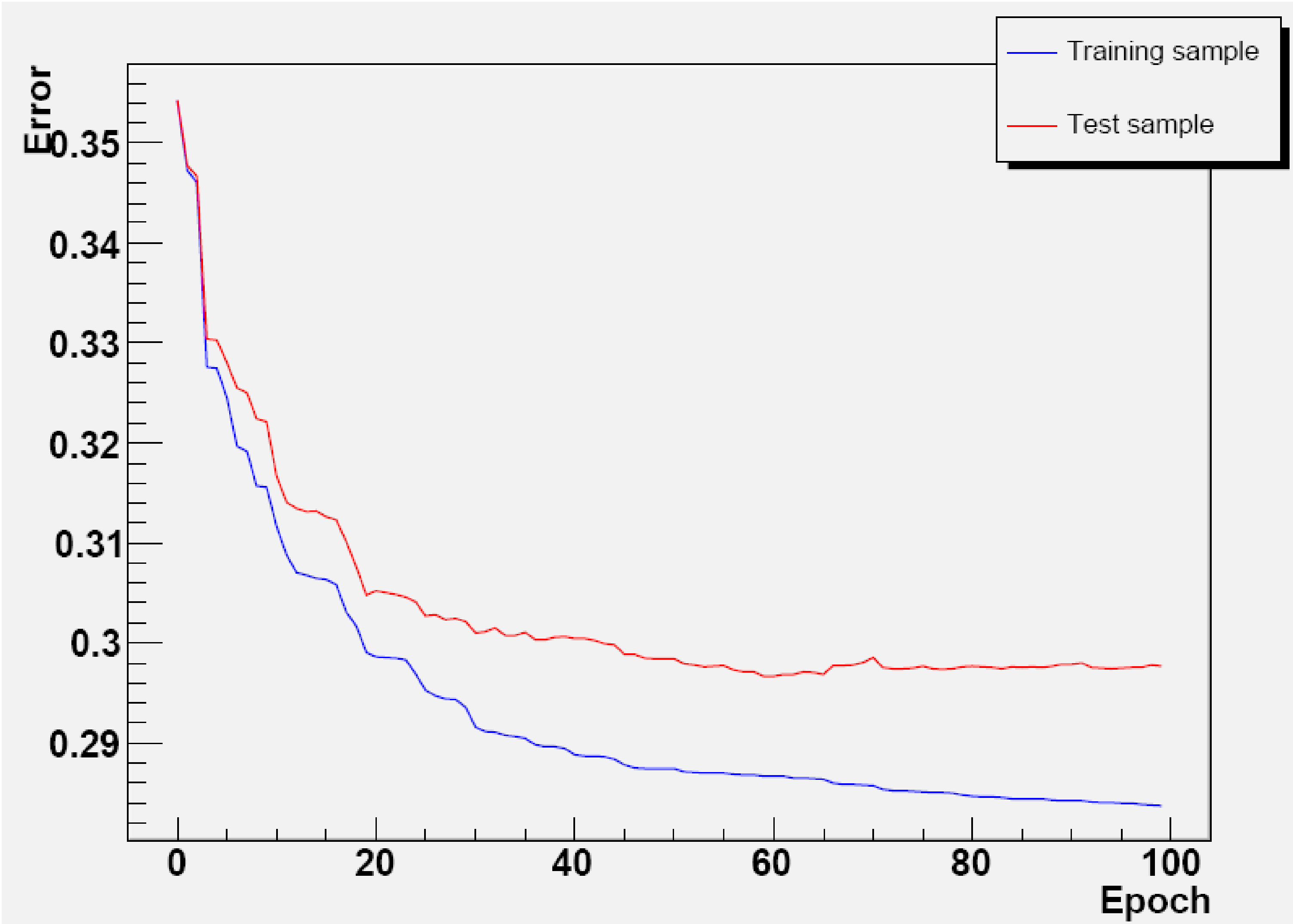


- mlp_train.C
  - Train the network and determine the node weights
  - Optimize structure for maximum S and B separation
- mlp_data.C
  - Run the signal, background, and challenge data through the network and produce NN output histograms
- mlp_fit.C
  - Here you insert your code that will fit the histograms to determine the signal fraction and plot the result
  - Two methods
    - Get bin contents, calculate χ^2 as a function of signal fraction, find minimum χ^2 – the way any *real* physicist would do it!
    - Use the root class designed for this kind of task – a black box, ugh!
- To run the macros, type

```
root –l mlp_x.C
```

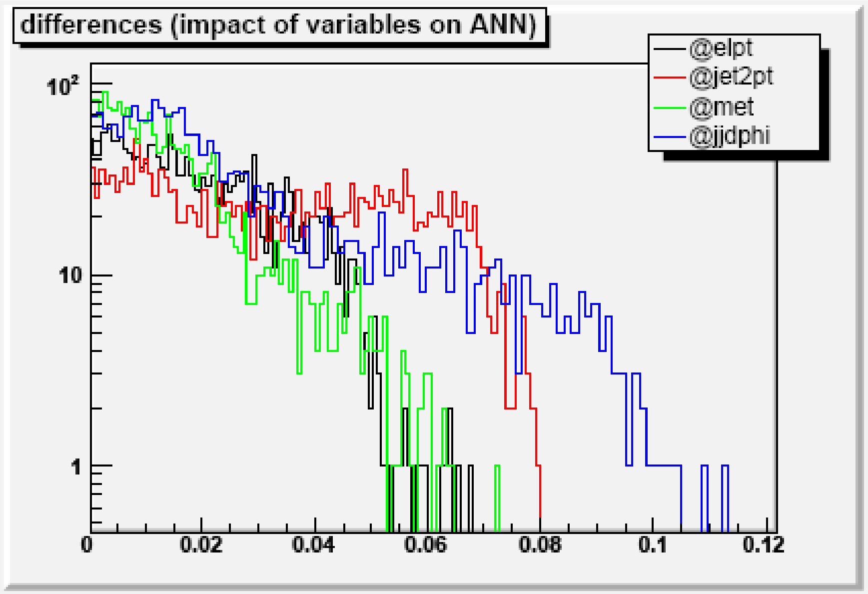
  where x = train, data, fit

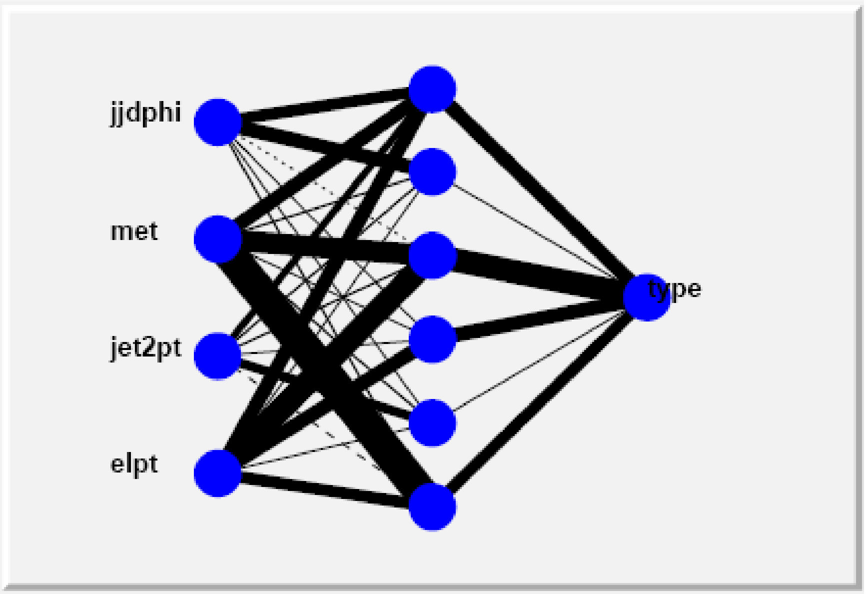
mlp_train.C Output

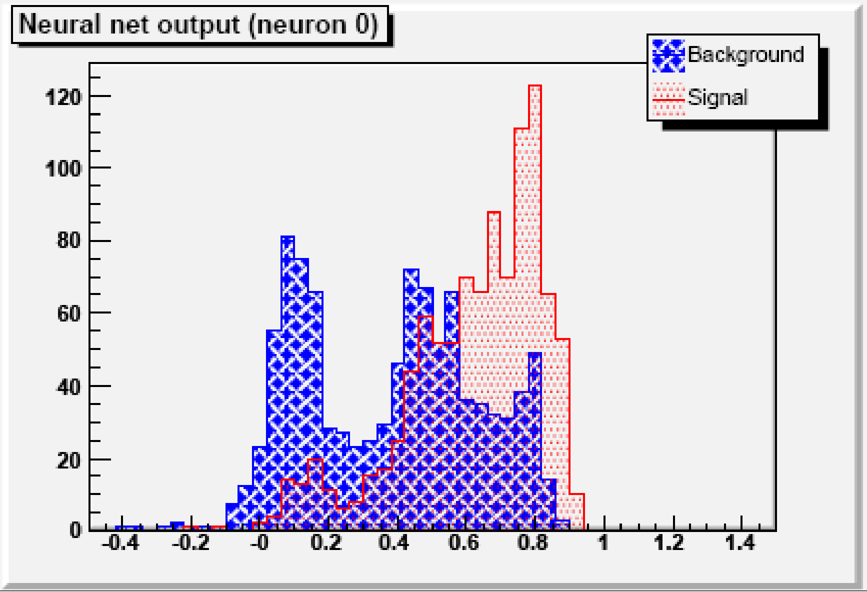


You can set the # of training epochs, 100 is a reasonable choice

mlp_train.C Output

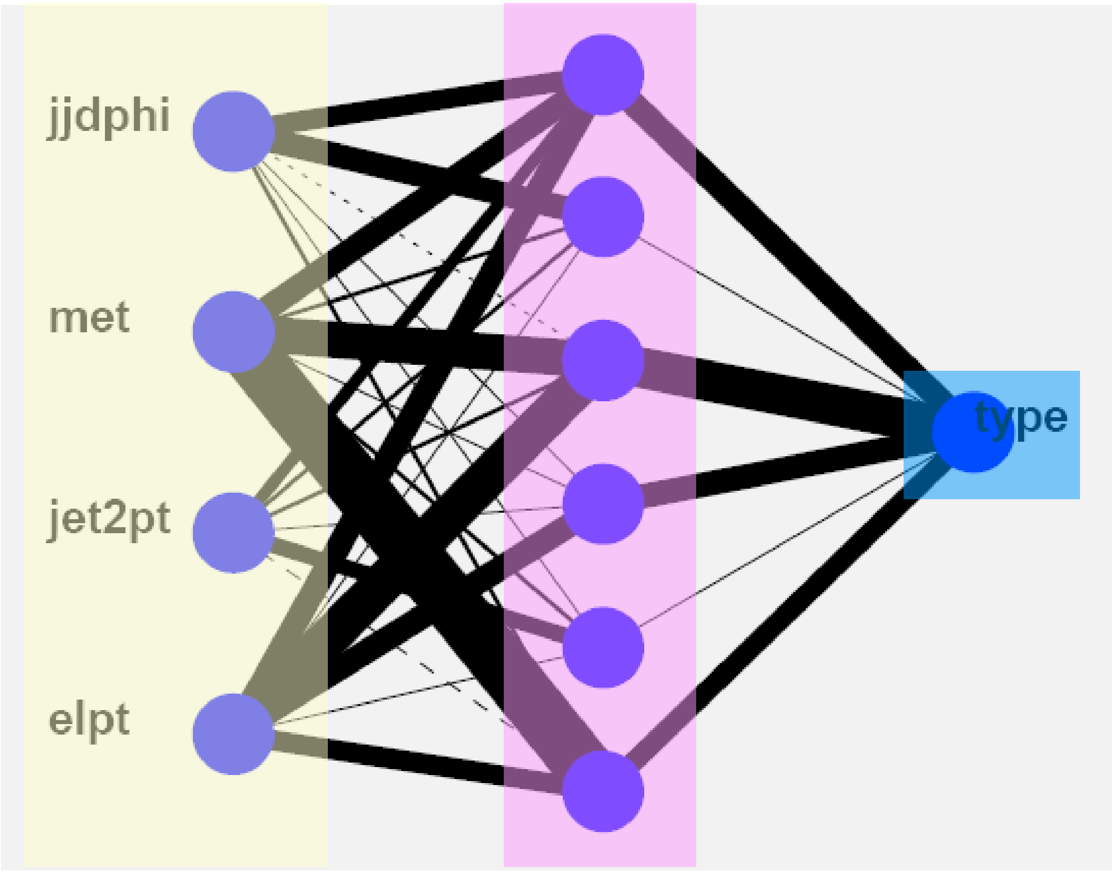






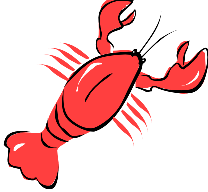
Watch it! The network changes every time you run the training

net_spec.txt



@elpt,@jet2pt,@met,@jjdphi:6:type

How to optimize the network?



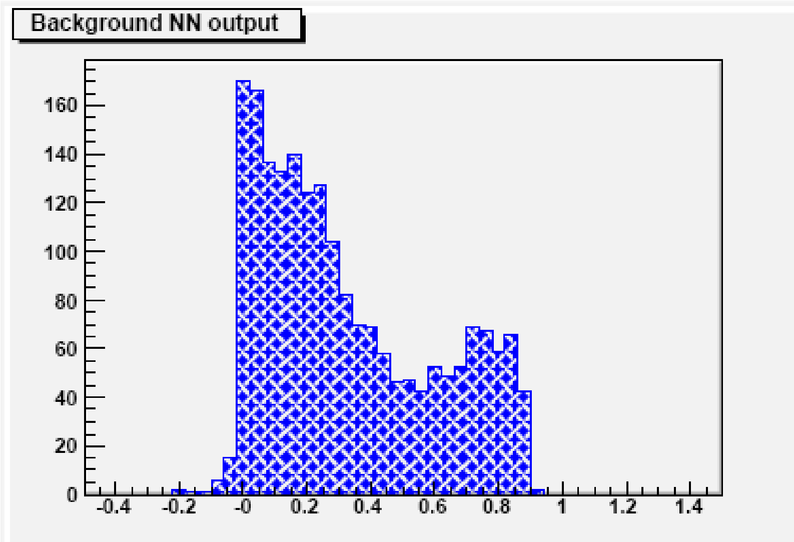
Advice from the root manual:

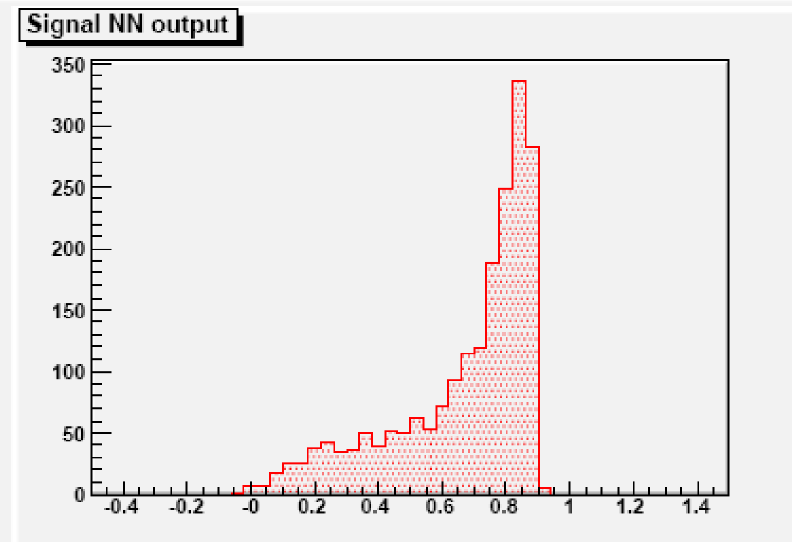
> Many questions on the good usage of neural network, including rules of dumb to determine the best network topology are addressed at ftp://ftp.sas.com/pub/neural/FAQ.html

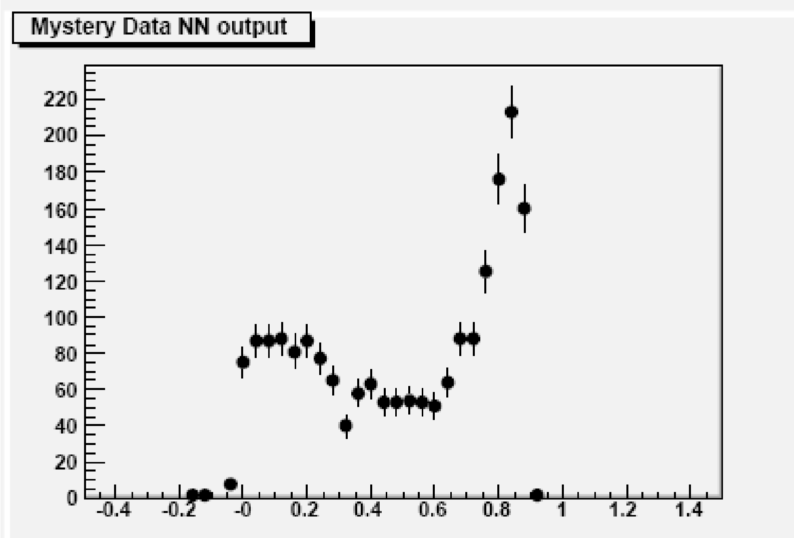
Play with the topology and see what you come up with!

BOSTON UNIVERSITY

mlp_fit.C Output



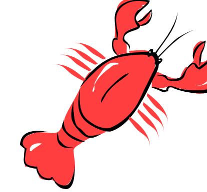




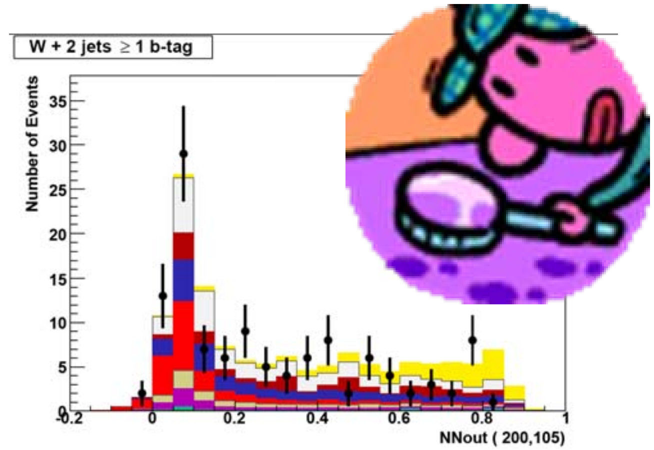
Your job is to fit the data histogram to the sum of the background and signal histos thereby determining the signal fraction

BOSTON UNIVERSITY

Time to get started!



- All macros and root files are available at http://budoe.bu.edu/~jmbutler/NEPPSR/
- Project Finale on Thursday
  - Send me your results by Wednesday evening for inclusion
  - Don’t be shy, there will be fabulous prizes for the best results!
- Stephane and I will be around to answer questions, provide hints, etc.
- Good luck in your search for new physics and do have fun!



BOSTON UNIVERSITY