The totem Neurochip and HEP Applications

Antonio Sidoti
INFN Padova

- Neural Networks in HEP
- The TOTEM Neurochip
- HEP Applications
  - Case Study #1: Higgs at LHC
  - Case Study #2: The hadronic *top* at CDF
- Conclusions

Neural Networks

Considerable **overlap** between neural networks and statistics (some examples from `comp.ai.neural-nets`):

NN Jargon	Statistical Jargon
Architecture	Model
Pattern, Example	Observation, Case
Training, Learning	Estimation, Model fitting, Optimization
Classification	Discriminant analysis
Supervised learning	Regression, Discriminant analysis
Training set	Sample, Construction sample
Test set, Validation set	Hold-out sample
Generalization	Interpolation, Extrapolation,Prediction

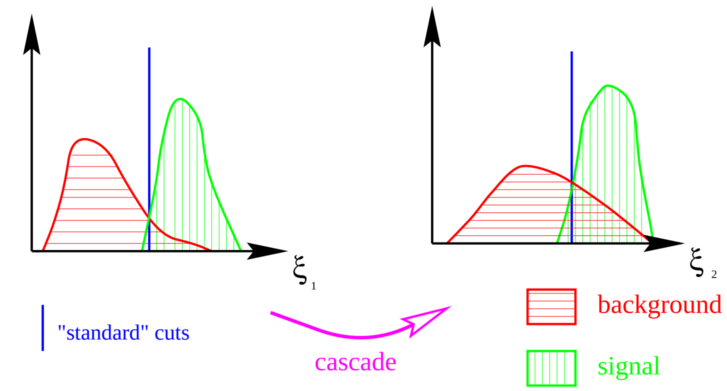
FFNN applications in HEP:

- Deriving function parameters from examples (*e.g.* fitting).
- Pattern Recognition
- Classification of events in classes. Can be used in:
  - **On-line** as triggers.
  - **Off-line** as data analysis.

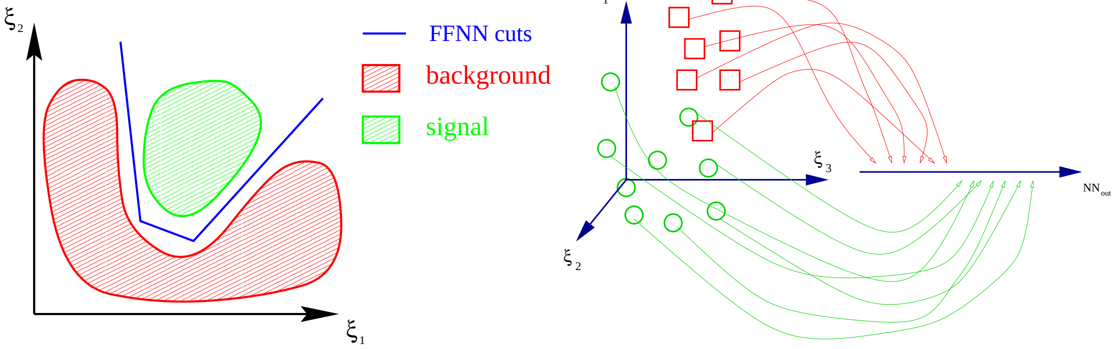
Neural Networks in HEP

Multivariate selection: $\Rightarrow$ Classsification of events in classes.

Standard cuts:



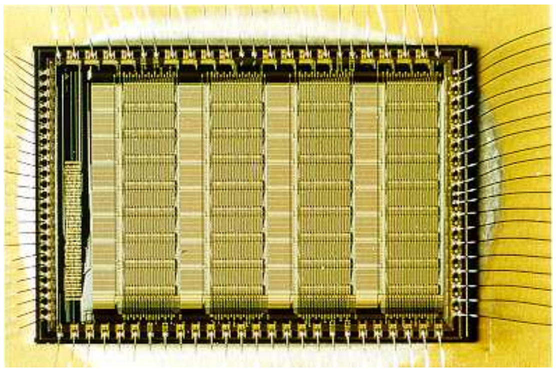
FFNN:

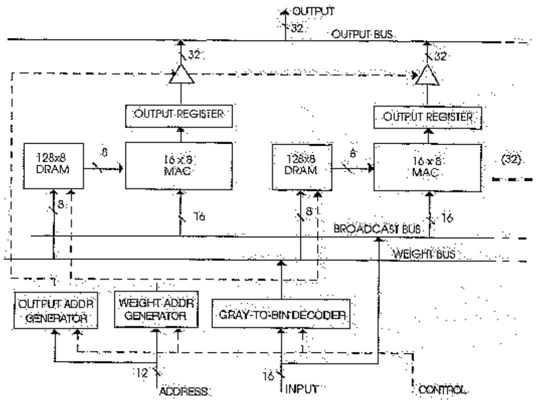


Exploiting correlations among variables
$\Rightarrow$ increase efficiency, purity.

The TOTEM NeuroChip

- Neurochip Produced by Neuricam (www.neuricam.com)
- Digital VLSI parallel processors $\Rightarrow$ Fast learning and recognition.





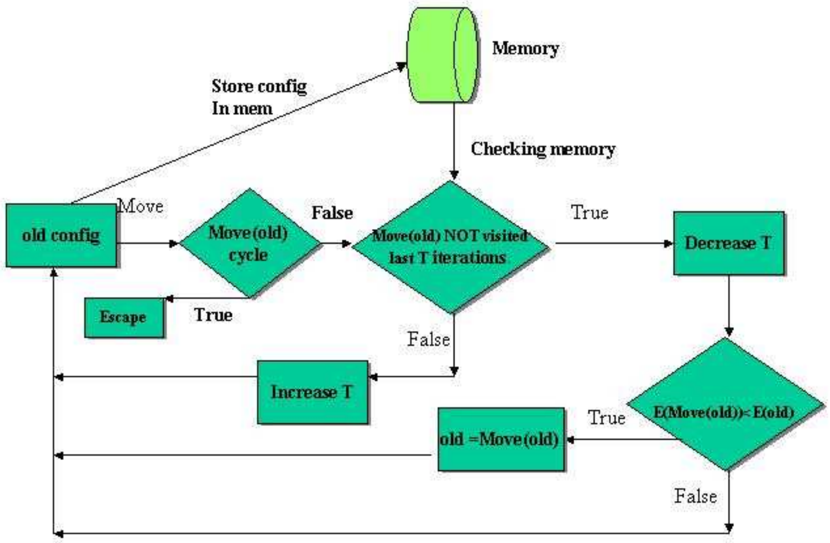
- Several boards are available: ISA, VME and PCI architecture.
- Chip architecture optimised for implementation of "Reactive Tabu Search".
- **Fast:** Performance of 1.5 GMAC/s (10^9 **M**ultiply and **AC**cumulate operation per second) with a clock of 50 MHz $\Rightarrow$ a 64-128-64 FFNN can evaluate an event in about $9\mu s$

The RTS Training algorithm

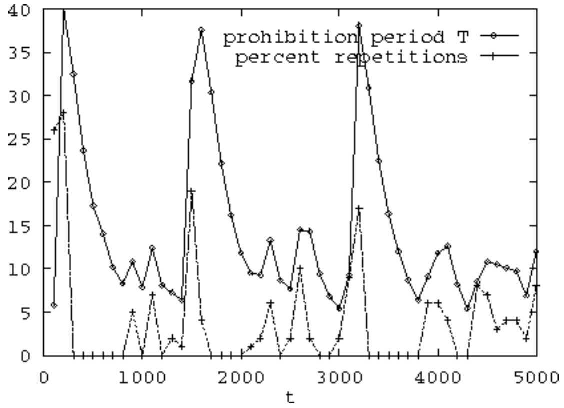
- High processing speed $\Leftarrow$ no floating point arithmetic; integer weights.

- **Drawback:** Backpropagation cannot be used $\Rightarrow$ the **R**eactive **T**abu **S**earch algorithm has been used.
- Tabu Search (from F. Glover) based on *temporary prohibition* of moves in the weight configuration space. Open problems:
  - Determination of *prohibition* period.
  - Adoption of algorithms for using memory
  - Robustness of technique for a wide range of problems
- In RTS prohibition period determined with feedback reactive mechanisms.
- All configurations found during the search are stored in memory.
- Time access $\sim O(1)$ (indipendence from number of iteration) through "hashing" techniques.
- Mapping of weight space $\longrightarrow$ Message space (`Grey` encoding))

The RTS Algorithm: **RTS**



Reactive Mechanism:



Advantages

- Research of minima is more **aggressive** ⇒ In the same time a larger region is explored.
- Avoid trapping in local minima and cycling

Case study #1: Higgs at LHC

- **Objective:** Higgs (with $M_H = 200 GeV/c^2$) at LHC ($\sqrt{s} = 14\,\mathrm{TeV}$) (Phys. Lett B **409** (1997) 517-522).
- Process: $pp \to H \to ZZ \to 4\mu$.
- Backgrounds: $t\bar{t}$ and $Z^0 b\bar{b}$.
- Generator: Herwig
- No detector simulation $\Rightarrow$ Parton level study
- Input variables:

  $X_1 - X_4$ Trasverse momentum of four muons.

  $X_5 - X_8$ Invariant mass for the different $\mu^+\mu^-$ pairs.

  X_9 The four muons invariant mass

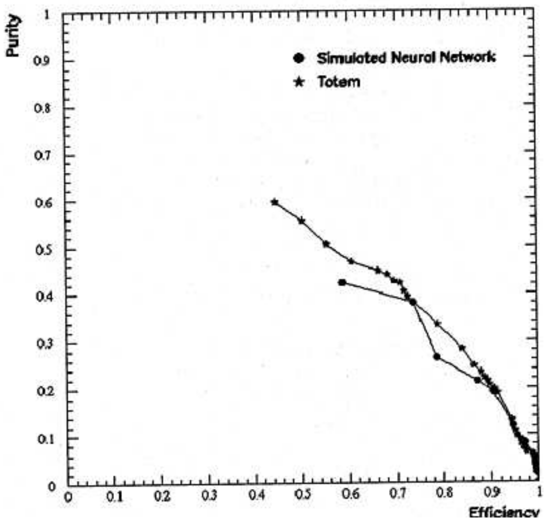
  X_{10} Hadron Multiplicity

The Higgs Case

⇒ Evaluation of Efficiencies and Purity.

- Training and testing sets:

Process	Num train. evts	Num gen. evts
$HX \rightarrow 4\mu X'$	2000	2000
$t\bar{t}X \rightarrow 4\mu X'$	2000	5.6×10^6
$Z^0 b\bar{b}X \rightarrow 4\mu X'$	2000	4.2×10^6



Comparison of performances between "standard" backprop and TOTEM

HEP Application: All hadronic channel at CDF

Objective: measurement of $t\bar{t}$ production cross section and *top* mass in the all hadronic channel decay (CDF Note 4722 and NIM A to be published).
Final state considered:

$$t\bar{t} \rightarrow bq_1 q_2 \bar{b} q_3 q_4 \rightarrow 6 \ \ jets$$

- 44 % of all $t\bar{t}$ decays;
- signal overwhelmed by QCD multijet background;

Both CDF and D0 analysed this channel ⇒ Compare obtained results.

Data selection and Choice of examples

Data collected during Run I at CDF
Multijet trigger requirement;

- Real data events: events collected at CDF passed the data selection:
- Events with at least one b-tagged jet ($N(tag) \geq 1$) $\Rightarrow$ *final data sample* (signal enriched).
- Events with **NO** b-tagged jet $\Rightarrow$ *multijet background* (signal depleted (0.5%)).

Monte Carlo generated events:

- **Signal** $t\bar{t}$ with different M_{top}.
- **Background** QCD generated background events (as checks)
- Input variables are chosen looking at their discriminating power ($S/\sqrt{B}$).
- Check if variables are linearly correlated:
  1. Identifying couples of variables with linear correlation coefficient.
  2. Train the FFNN without one of them and check NN performances.
  3. From geometrical interpretation $\rightarrow$ check the weights of a 2-N-1 FFNN trained with the two linear correlated variables.

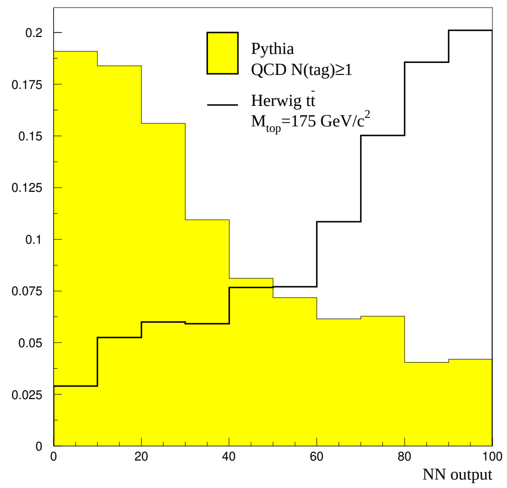
Choice of Variables

Only kinematical information from jets' $\vec{P}^{\mu}$.

$\sum E_T$	the scalar sum of jets' transverse energy.
Aplanarity	$\propto$ lowest eigenvalue of momentum matrix.
Centrality	$\sum E_T/\hat{s}$.
$\sum E_T^3$	scalar sum of non leading jets' E_T.
$< \eta^2 >$	quadratic dispertion of jets in $< \eta >$ direction.
$\sqrt{E_{T_5} E_{T_6}}$	geometric mean of fifth and sixth jets' E_T.
M_2 and M_3	two lowest eigenvalues of Parisi matrix.

Training and Generalization

- Avoid sensitivity on *top* mass $\rightarrow$ Mixture obtained with different M_{top} for $t\bar{t}$ training events.
- Traning set: 900 *multijet bckg* evts + 900 MC $t\bar{t}$ evts.
- Generalization results.

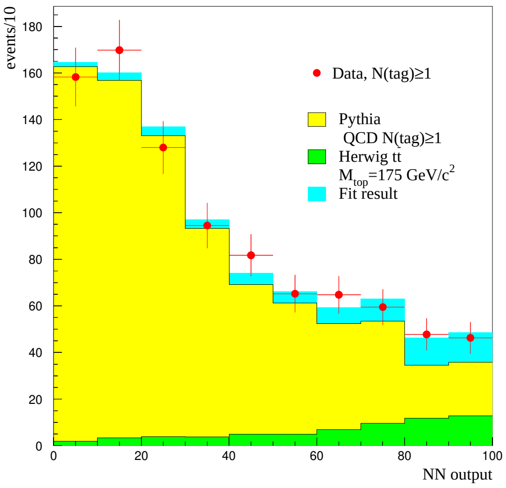


$\Rightarrow$ NN_{out} discriminates between signal and background events.

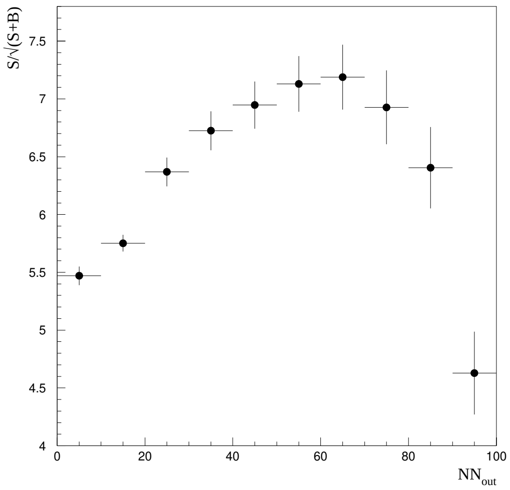
Different methods to measure the cross secction:

- **Counting Experiment** Choose a *cut* value on NN_{out} that maximize the $S/\sqrt{B}$ on a *test* set and perform a counting.
- **Two Component Fit** Estimate the fraction of signal and background in the NN_{out} distribution of the *final data sample* with a two component fit.

Results



Results of the two component fit assuming $M_{top} = 175$ GeV$/c^2$.



A value of *cut* = 60 maximize the ratio. Using the same cuts of *traditional* analysis $\longrightarrow S/\sqrt{B} = 6.93 \pm 0.43$.

Systematics

"Traditional" systematics:

- Luminosity;
- b-tag efficiency;
- selection efficiencies;
- b-tag bias $\Leftarrow$ how much the NN_{out} distributions change requiring different *tag* informations?

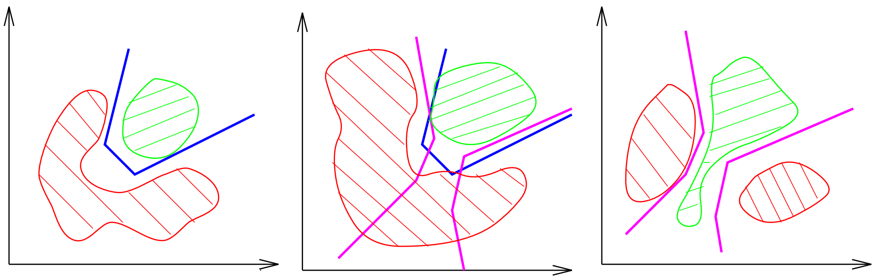
"Specific" systematics of NN approach:

? Choice of the architecture (*e.g.* Number of hidden layers, number of neurons in hidden layers, shape of sigmoid)

- **Main issue** Finite number of training examples $\Rightarrow$ how many training events are necesssary and quantify the systematics uncertainty?

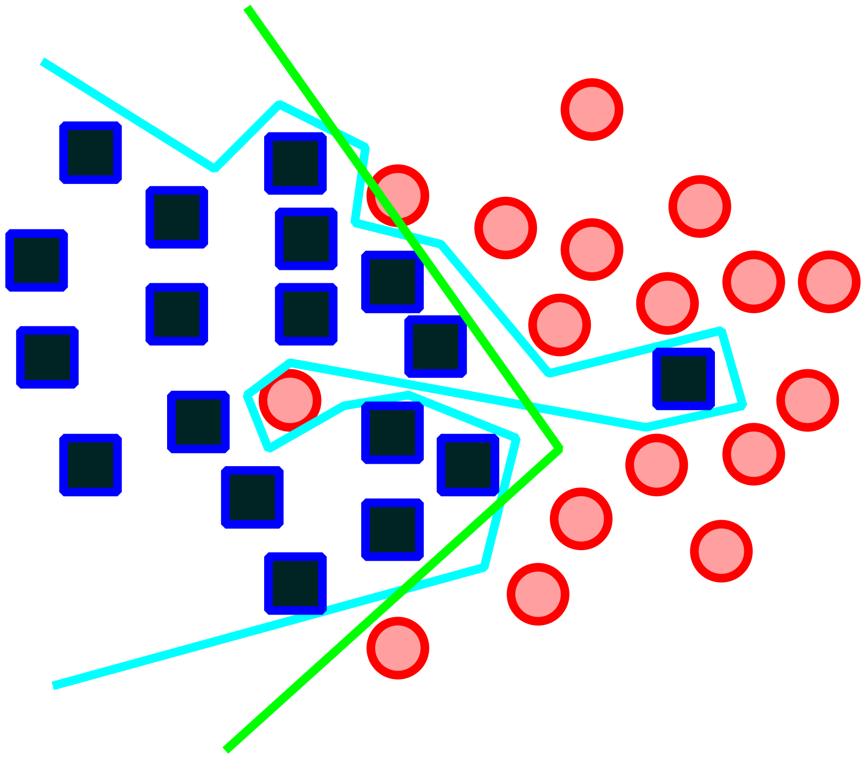
Size of Training events

- If the number of training events is small ⇒ Large statistical fluctuations.



⇒ Wrong Classification.

- Large complexity in architecture ⇒ Many cuts.



Check of NN systematics

Problem dependent issue.

Normally one tries one runs more MC to increase the number of training events. In this case that was impossible as we were using “real”” events.

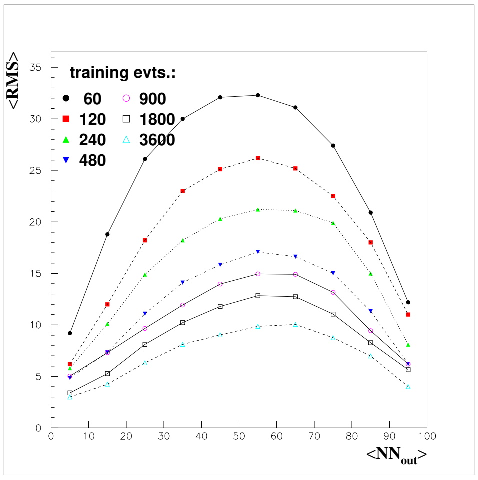
Idea for the evaluation of systematics:

- Create non overlapping training sets of different cardinality (loosen the selection cut to increase the sample statistics).
- perform training on these using a FFNN with the same architecture.
- generalization on a common *test* set.

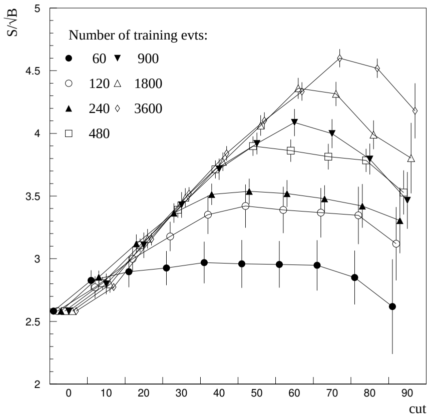
Checks:

- **How much the shapes change?** $\Rightarrow$ Difference in NN_{out} shapes obtained for the various generalization?
- **How much a specific event is classified differently?** $\Rightarrow$ $< RMS >$ check.

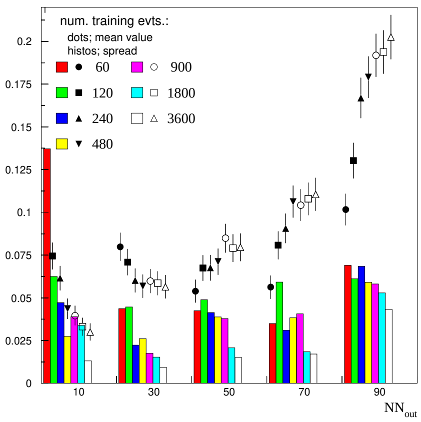
Check Results



Reshuffling: Differences of classification of events by FFNN trained with different training sets with same number of events.



Significance: Modifications in $S/\sqrt{B}$ increasing the number of training events.



Shape modification: Modifications of NN_{out} shapes increasing the number of training events.

Results

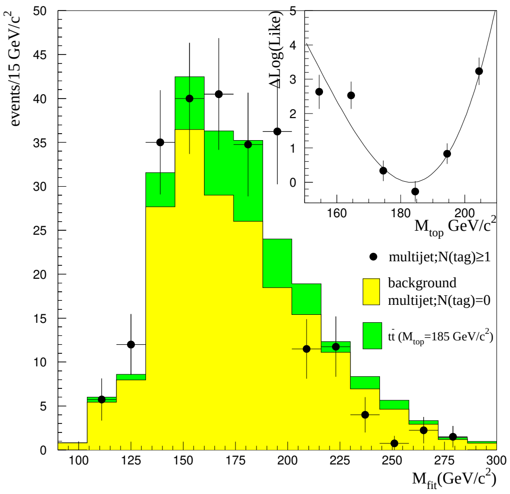
SYSTEMATICS (%)			
	2 Comp Fit	Counting	Traditional
	5.8	5.7	9
Generator mod.	6.0	4.2	9
Gluon rad.	5.0	9.3	11
b-tag signal	13	13	13
background est.	4.1	6.6	6.5
Luminosity	3.9	3.9	3.9
Training	*4.7*	*5.7*	n.a.
TOTAL	*17.7* 17.4	19.9 *19.1*	22.6
Cross Sec.(pb)	$8.6 \pm 1.9 \pm 1.5$	$7.9 \pm 3.1^{+2.1}_{-1.1}$	$9.6 \pm 2.9^{+3.3}_{-2.1}$

- Measurements are in agreement with the "Standard" ones.
- Systematic uncertainties are lowered also adding the "Training" ones $\Leftarrow$ the only straight cut is $NN_{out} > cut$.
- The "Two Component Fit" method decreases the systematics (**NO** straight cuts at all.)

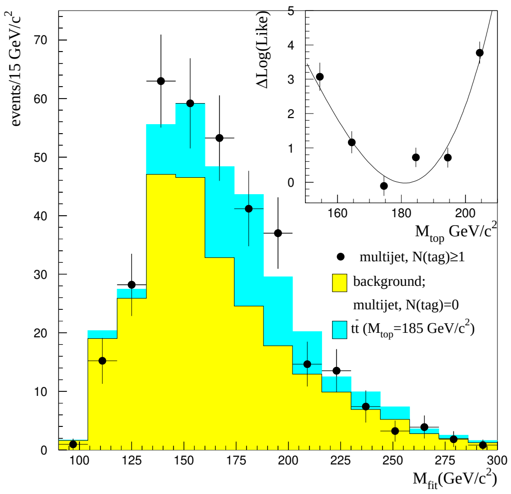
Mass Measurement

- For each event a reconstructed mass is calculated.
- Two methods to get M_{fit} distributions.
- Two Comp fit on mass measurements $\Rightarrow$ Likelihood fit $\Rightarrow$ Top mass measurement

- Method `cut`: Plot M_{fit} for events with $NN_{out} \geq 60$.
- Method `all`: Plot M_{fit} distributions considering **all events** weighted with their NN_{out}.



Cut



All

Results

SYSTEMATICS (%)			
Source	cut $\Delta(\%)$	all $\Delta(\%)$	*Standard*
Jet Energy scale	2.7	2.6	2.7
Init.-Final state	1.1	1.0	0.97
Monte Carlo mod.	0.2	0.2	0.11
Monte Carlo gen.	0.47	0.34	0.43
Background shape	0.5	0.4	0.91
Monte Carlo stat.	0.3	n.a.	0.32
training sample	*1.6*	*1.5*	n.a.
TOTAL	3.0 *(3.4)*	2.8 *(3.2)*	3.1

- NN Mass measurements are in agreement.

$$\begin{aligned} M_{cut} &= 183.4 \pm 9.5 \pm 6.2 \mathrm{GeV}/c^2 \\ M_{all} &= 181.1 \pm 8.8 \pm 5.7 \mathrm{GeV}/c^2 \\ M_{stand} &= 186.0 \pm 10.0 \pm 5.7 \mathrm{GeV}/c^2 \end{aligned}$$

- For the mass measurement no significant improvement using NN methods $\Rightarrow$ the uncertainties affects prinipally the values of reconstructed mass for each event.

- As expected, straight cuts increase systematic uncertainties.

Conclusions

- We have learned and applied NN methods to HEP ⇒ possible methods to quantify uncertainties connected with the finite size of training events.
- TOTEM board ⇒ proved to work in a HEP environment.

PRO:
- Fast, reliable.
- need only a PC with Linux or Windows NT.

⇒ can be used as trigger.

- Coupled with **RTS** ⇒ Very efficient training algorithm.

CONTRO Missing software interface specific to HEP ⇒ Need to integrate it (*e.g.* using the Root/JetNet interface, `AC++` Module,...).