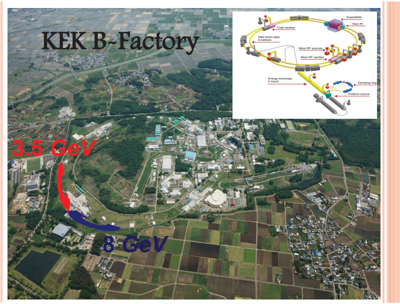
Application of Neural Networks for the Belle II Experiment

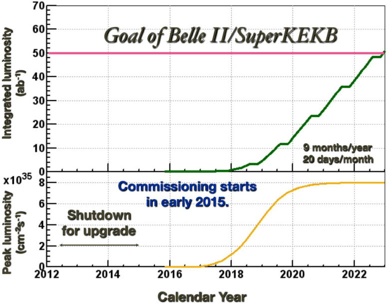
Fernando Abudinén

July 16 2014

$\Delta p \cdot \Delta q \geq \frac{1}{2} \hbar$

Upgrade of KEKB to SuperKEKB (Tsukuba)





$e^+ \rightarrow \Upsilon(4S) \leftarrow e^-$

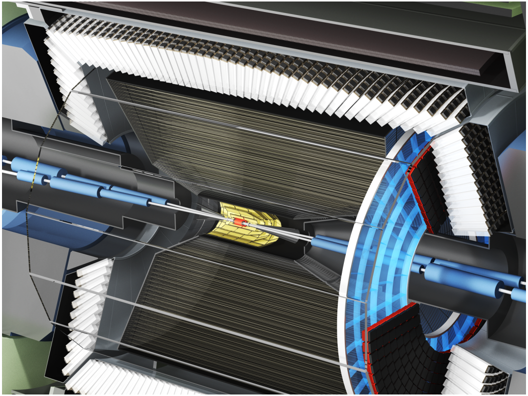
$B\bar{B} \;\Rightarrow\; \sqrt{s} = 10.58$ GeV

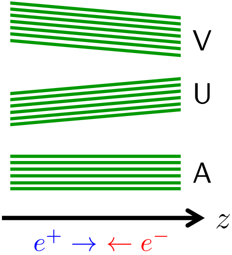
- Instantaneous luminosity of $L = 0.8 \cdot 10^{36}$ cm^{-2}s^{-1}

$\Rightarrow$ 40 times higher than the world record reached by KEKB.

$\Rightarrow$ 50 times larger data sample

The Belle II Detector





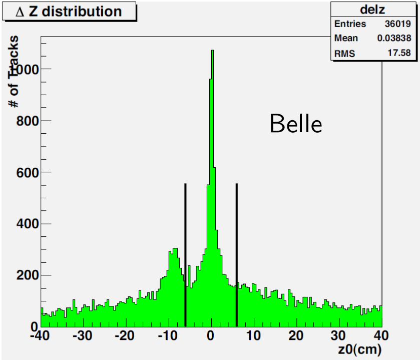
- CDC: Provides track information for the trigger.

$\Rightarrow$ 15000 sense wires in 9 superlayers

$\Rightarrow$ Config. AUAVAUAVA "A" $\hat{=}$ Axial, "U" and "V" $\hat{=}$ stereo

Why a z-Vertex Trigger?

- Undesired scattering processes (Touschek, Beam-Gas scattering, etc) $\Rightarrow$ **background**
- Background events not from collision point ($z = 0$)
- Higher luminosity $\Rightarrow$ higher background (factor $\sim$ **30**)
- Data prod. rate $\gg$ transfer $+$ record capacity $\Rightarrow$ L1 Trigger



- Filter out events with vertex ($z_0 \neq 0$)

$\Rightarrow$ Goal: High resolution z-vertex trigger ($\sigma \leq 2$ cm)

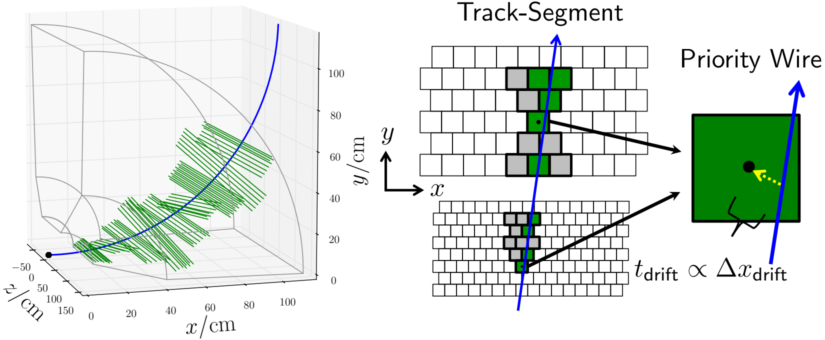
$\Rightarrow$ Cut at $\pm 3\sigma$

- Trigger latency ($\sim 5\ \mu s$)

Why a z-Vertex Trigger?

CDC Tracking:

- 15000 wires $\Rightarrow$ 2336 track-segments (TS)



Which information is available for triggering?

a) Identification numbers of active track-segments (TS-IDs)

$\Rightarrow$ Position of priority wire

b) Drift times of priority wires

Multi Layer Perceptron

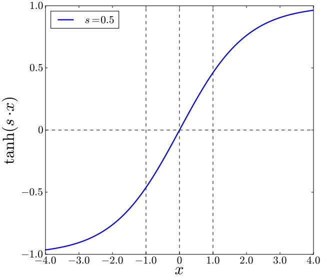
a) TS $\hat{=}$ Input-Neurons

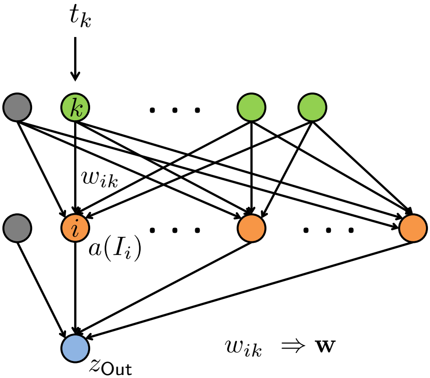
b) Drift times $\hat{=}$ t_k Input values

$\Rightarrow$ Hidden layer: $n_{\text{hidden}} = 3 \cdot n_{\text{input}}$

$\Rightarrow$ Input: $I_i = \sum_{k=0}^{n_{\text{input}}} w_{ik} t_k$

$\Rightarrow$ Output: $a_i = \tanh(I_i)$





$\Rightarrow$ Output neuron:

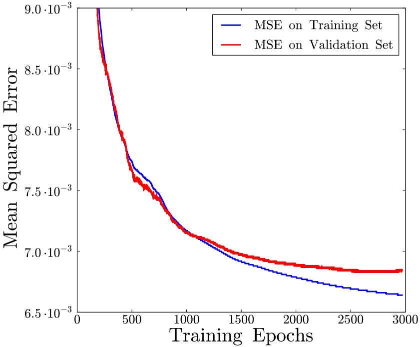
$z_{\text{Out}} = a(\sum_{i=0}^{n_{\text{hidden}}} w_{\text{Out},i} \cdot a(I_i))$

- Real vertex from simulation: z_{True}

Training Process

⇒ Training Sample: $N_{\text{train}} \sim 20000$ MC events

- Training: Modify $\mathbf{w}$ stepwise ⇒ $z_{\text{Out}} \to z_{\text{True}}$
- Training Algorithm: BP (Back Propagation)
- BP evaluates cost function $E(\mathbf{w})$ (MSE) at each step n:

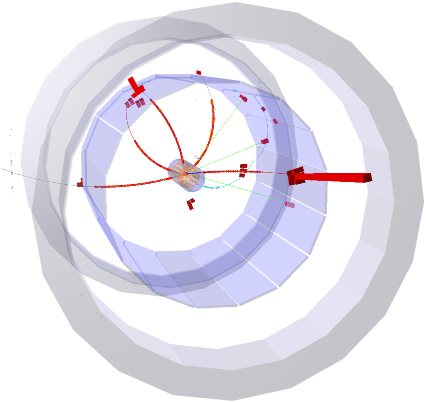


$$E(\mathbf{w}) \equiv \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left(z_{\text{True}}^{j} - z_{\text{Out}}^{j} \right)^2$$

$$\Rightarrow \quad \Delta\mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}}$$

$$\Rightarrow \quad \mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}$$

Method



- Decompose events in single tracks $\Rightarrow$

CDC Phase Space:

- $\phi \in [0°, 360°]$
- $\theta \in [17°, 150°]$
- $p_\mathsf{T} \in [0.2, 5.2]$ GeV$/c$
  $p_\mathsf{T} \propto \kappa^{-1}$

- Whole CDC phase space too much input for a single MLP

$\Rightarrow$ Divide it in sectors:
$\Delta\phi \sim 1°$, $\Delta\theta \sim 6°$
$\Delta p_\mathsf{T} \sim 0.05$-$0.6$ GeV$/c$

$\Rightarrow$ $\sim 2 \cdot 10^6$ sectors

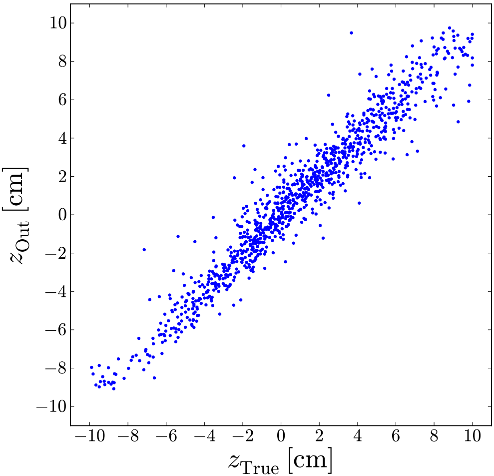
- Find the sectors which are firing!

MLP Test

Train a MLP for each specific sector!

Test of the MLP after training:

⇒ Compare MLP output with true value! (simulated tracks)

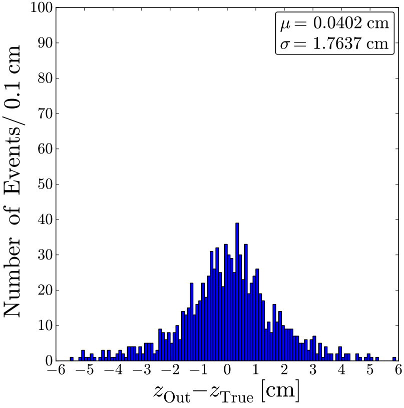


- To evaluate the performance consider:

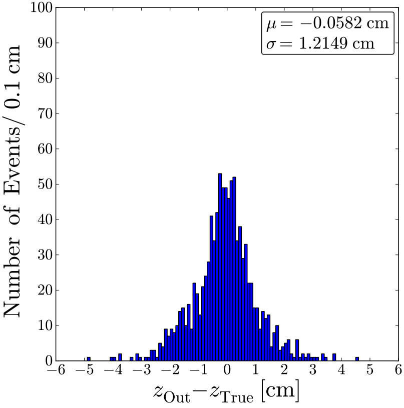
$$\mu = \text{mean}(z_{\text{Out}} - z_{\text{True}})$$
$$\sigma = \text{std}(z_{\text{Out}} - z_{\text{True}})$$

Results without Background

$p_{\mathsf{T}} \in [5.0, 5.2]$ GeV/c $\Rightarrow$ vary θ



(a) $\theta \in [30°, 36°]$

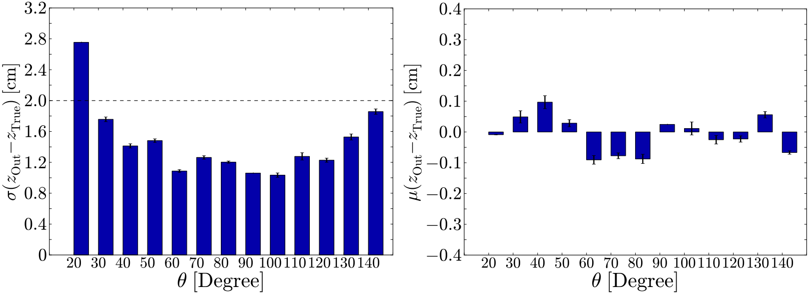


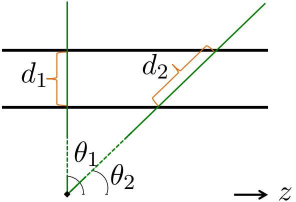
(b) $\theta \in [80°, 86°]$

Results without Background

Network resolution as a function of polar angle θ:

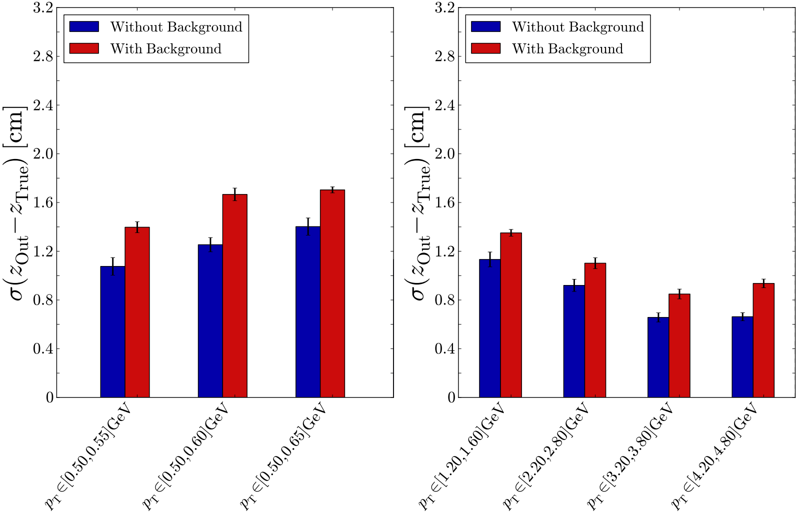
$\Delta\theta = 6°$





Results without and with Background

$\theta \in [56°, 62°]$ $\Rightarrow$ vary p_T

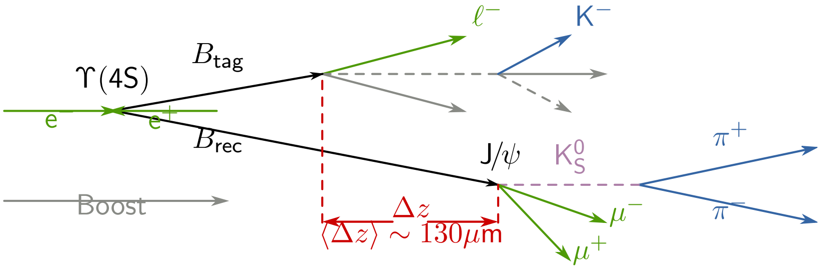


Outlook of the z-vertex trigger

$\Rightarrow$ With and without background: Resolution significantly better than required (< 2 cm). ☺

- Background leads to an average resolution loss of $\sim 25\%$
- Parallelism inherent to the computations makes the MLP suitable for L1 trigger $\Rightarrow$ realizable in FPGA. ☺
- The number of required MLPs ($\sim 2 \cdot 10^6 \hat{=} 10$ Gb) is a challenge for the hardware implementation.
- Next step: Decomposition of a specific event in separate tracks? (Look-Up-Table, 2D Trigger of Belle II?)

- Goal: Measurement of time dependent violation of CP symmetry in the B-Meson system



- B_{rec}: Fully reconstructed B-meson in decay to CP eigenstate
- B_{tag}: B-meson used for flavor determination

General approach

- B^0 has a high number of decay channels

$\Rightarrow$ full reco. of B_{tag} not feasible

- Some charged final state particles (Targets) correspond to

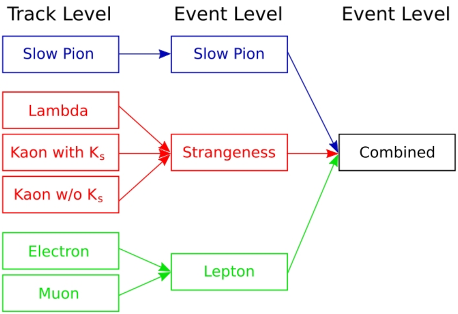
$\Rightarrow$ Flavor specific decays

- Examples:
  - $\bar{B}^0 \to X l^- \bar{\nu}$ $\quad (b \to c l^- \bar{\nu}) \to$ +(-) charged l tags B_0 ($\bar{B}_0$)
  - $\bar{B}^0 \to X K^-$ $\quad (b \to c \to s) \to$ +(-) charged K tags B_0 ($\bar{B}_0$)
  - $\bar{B}^0 \to X D^* \pi^-$ $\quad (b \to c) \to$ +(-) charged π tags B_0 ($\bar{B}_0$)
  - ...

$\Rightarrow$ Group flavor specific signatures into different categories!

Event and track level

- Two Steps needed for each category:
  Track Level → Event level



- Final step:

⇒ Category combiner

Each step $\hat{=}$ Multivariate Method

Flavor tagging

For each category:

- Assumption: remaining tracks in B_{rec} belong to B_{tag}

1. Track level output: probability of being the target track

$\Rightarrow$ Select the track with highest probability

2. Event level output: $y_{\text{Event}} = q \cdot r$:

   $q = \text{sgn}(y_{\text{Event}})$: flavor of B_{tag}

   $r = ||y_{\text{Event}}||$: expected flavor dilution factor

   $r = 1 - 2\omega$ with ω wrong tag fraction

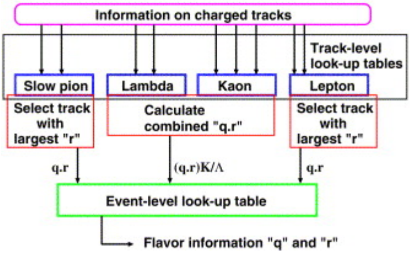
   $r = 1$ $(r = 0)$ indicates (no) flavor information

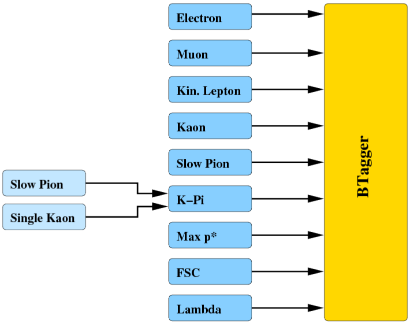
Combining all event level outputs:

3. Category Combiner output: combined dilution factor

$\Rightarrow$ $y_{\text{Combiner}} = q \cdot r$

Belle vs. Babar Scheme





⇒ Each step: Trained neural network (NN) or other TMVA method.

Semimuonic Category

- Primary Muons: $\bar{B}^0 \to X\mu^-\bar{\nu} \qquad (\, b \to c\mu^-\bar{\nu})$
  $\to$ pos (neg) charged muon tags B_0 ($\bar{B}_0$)
- Variables for Track Level (calculated only for each track):
  $q_{\text{MC}},\ p^{\text{cms}},\ \theta_{\text{lab}},\ \mathcal{L}_\mu$
- Variables for Event Level (calculated only for target track):
  $q_{\text{Class}},\ M_{\text{recoil}},\ p^{\text{cms}}_{\text{miss}},\ \cos\theta_{\text{miss}},\ E^W_{90}$

First Studies:

- Generate Semimuonic MC events: $\Upsilon(4S) \to B^0_1 B^0_2$
  $B_1 \to J/\Psi K^0_S \qquad B_2 \to X\mu\nu$
- Define function y_{MC} (track):

$$y_{\text{MC}}\,(\text{track}) = \begin{cases} 1 & \text{if } (\text{ track} \mathrel{\hat{=}} \mu) \,\&\, (\mu \to \text{mother} \mathrel{\hat{=}} B^0) \\ 0 & \text{else} \end{cases}$$

Training of MLP

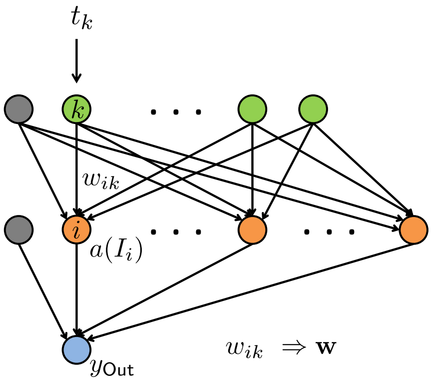
a) Input Variables $\hat{=}$ Track level Variables

b) Target $\hat{=}$ y_{MC}

$\Rightarrow$ Hidden layer: $n_{\text{hidden}} = 3 \cdot n_{\text{input}}$

$\Rightarrow$ Activation function
$a_i = \tanh(I_i) \in [-1, 1]$

$\Rightarrow$ Output neuron:
y_{Out} = Probability of being the target Track

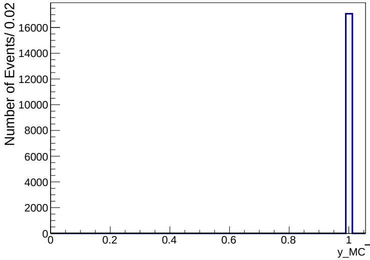


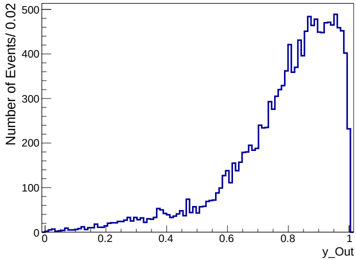
$w_{ik} \Rightarrow \mathbf{w}$

Training:

- Modify $\mathbf{w} \Rightarrow y_{\text{Out}} \rightarrow y_{\text{MC}}$
- Cost function= MSE
- Training Algorithm= BP

First Results





- Particles classified as target:

	μ	$\pi^{\pm}$	$K^{\pm}$	e	p
%	94.1	4.3	1.0	0.5	0.1

- Mothers of classified target:

	B^0	D	K	π	η	ρ
%	91.3	5.2	2.0	1.3	0.1	0.1

- Wrongly class. particles with right flavor ($\sim 3\%$):

	μ	$\pi^{\pm}$	$K^{\pm}$	e	p
%	24	36	36	3	1

Outlook

- Only the Track level of the Semimuonic category has been studied yet.

⇒ About 94% of N_{Signal} are correctly classified and nothing has been optimized yet!. ☺

Numbers for Evaluation

After Training the MLP is Tested with a Testing Sample.

- Some numbers to evaluate Performance:
  - N_{Events}: Total Number of Events.
  - $N_{\text{noParticle}}$: `ListSize` of `ParticleList` is 0 (No B_{reco}).
  - $N_{\text{Background}}$: Target muon is not in `RestofEvent`.
  - N_{noTracks}: No `Track` could be fitted.
  - N_{Signal}: `Track` of Target muon in `RestofEvent` can be reconstructed.
  - N_{Corr}: Correctly classified Events.
  - N_{Wrong}: A wrong Track is classified as Muon.
  - $N_{\text{Wrong\_rightFlavor}}$: Wrongly classified `Track` tags the right flavor.

First Results

- Sample of $N_{\text{Events}} = 62766$:
  - $N_{\text{noParticle}} = 44202$
  - $N_{\text{Background}} = 2201$
  - $N_{\text{noTracks}} = 55$
  - $N_{\text{Signal}} = 16308$
  - $\frac{N_{\text{Corr}}}{N_{\text{Signal}}} = 0.9127$
  - $\frac{N_{\text{Wrong}}}{N_{\text{Signal}}} = 0.0873$
  - $\frac{N_{\text{Wrong\_rightFlavor}}}{N_{\text{Signal}}} = 0.0272$
  - $\frac{N_{\text{Corr}} + N_{\text{Wrong\_rightFlavor}}}{N_{\text{Signal}}} = 0.9399$

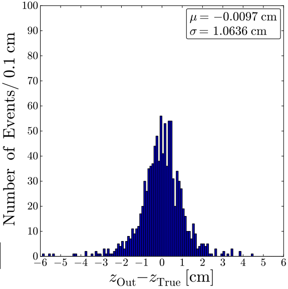
$\Rightarrow$ About 94% of N_{Signal} are correctly classified and nothing has been optimized yet!

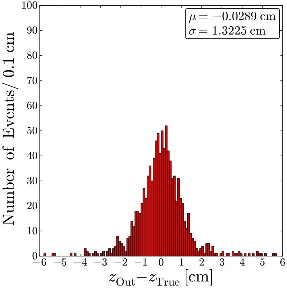
Results without and with Background

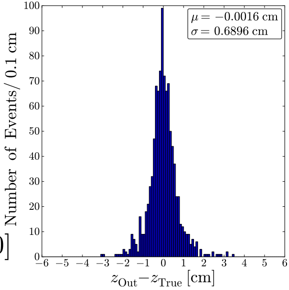
$\theta \in [56°, 62°]$

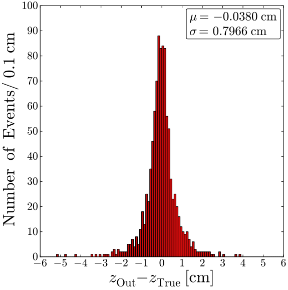
$\Rightarrow$ vary p_T

$p_\mathsf{T} \in [0.50, 0.55]$ GeV/c



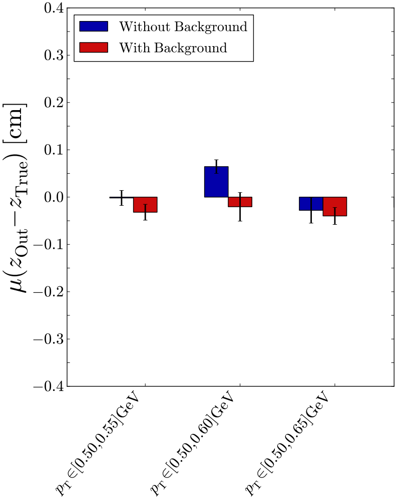


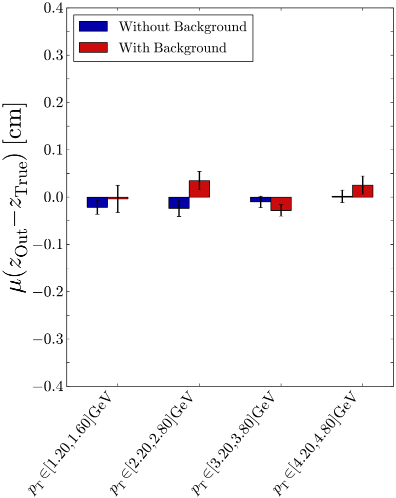




$p_\mathsf{T} \in [3.20, 3.80]$ GeV/c

Results without and with Background

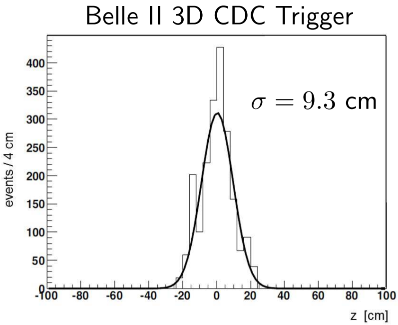




Why a z-Vertex Trigger?

- Belle II: Present z-vertex trigger uses **only TS-IDs** (Hough transformation)

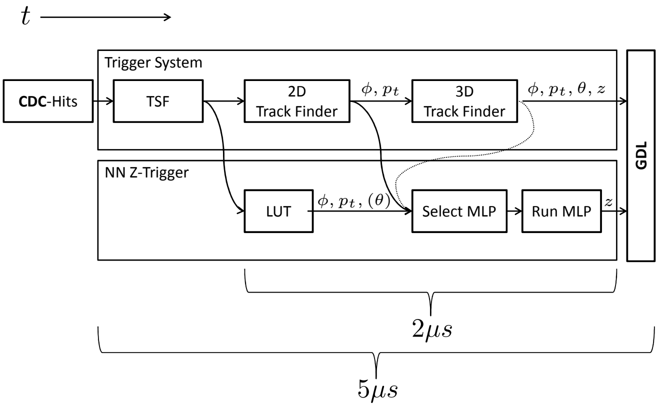
- No time for full track reconstruction !!!



- Significantly better results expected from neural networks (MLP) using **drift times**

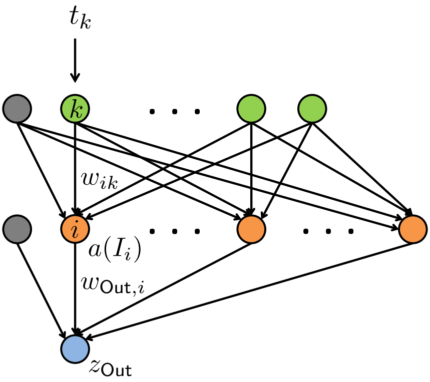
$\Rightarrow$ Parallelism of MLP computations suitable for L1 trigger

The Whole Trigger



Multi Layer Perceptron

How does it work?



- Input I_{Out} for output neuron:
  $I_{\text{Out}} = \sum_{i=0}^{n_{\text{hidden}}} w_{\text{Out},i} \cdot a(I_i)$

- CDC-Track Segments (TS) $\hat{=}$ Input-Neurons (Input Layer)
- Input values t_k $\hat{=}$ Drift times
- Neurons in the middle layer $n_{\text{hidden}} = 3 * n_{\text{input}}$ (Hidden Layer)
- Connection weights w_{ik}
- Activation function $a_i = \tanh(I_i) \in [-1, 1]$
- Input I_i for a neuron i:
  $I_i = \sum_{k=0}^{n_{\text{input}}} w_{ik} t_k$
- Output of the output neuron:
  $z_{\text{Out}} = a(I_{\text{out}}) \in [-1, 1]$

Multi Layer Perceptron

- From each simulated event: real vertex position on the z axis: z_{True} $\Rightarrow$ needs only to be scaled to $[-1, 1]$.
- Training means to iteratively modify all weights $\mathbf{w}$, in order for z_{Out} to converge to z_{True} using a training algorithm (iRPROP^{-}).
- The iRPROP^{-} algorithm evaluates at each iteration (training) step the Mean Squared Error function:

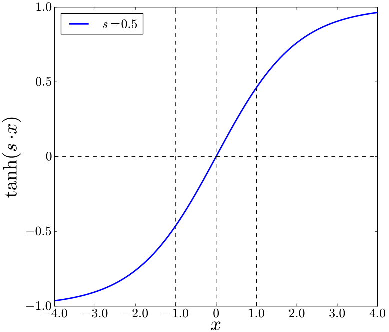
$$E(\mathbf{w}) \equiv \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left(z^{j}_{\text{True}} - z^{j}_{\text{Out}} \right)^2$$

- Training Sample $N_{\text{train}} = 20000$ events

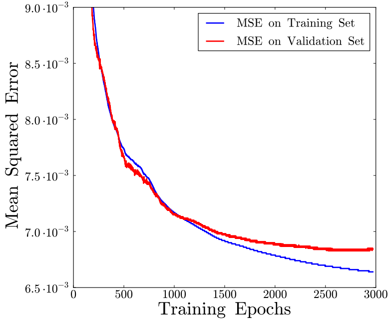
$$\Delta\mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}} \Rightarrow \mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}$$

- $n^{\text{weights}}_{\text{total}} = f_{\text{hidden}} \left(n^2_{\text{input}} + 2n_{\text{input}} \right) + 1 \ \sim 900 - 3000$

Activation Function and Training Process



(a) Tangens Hyperbolicus



(b) Training Process

Method

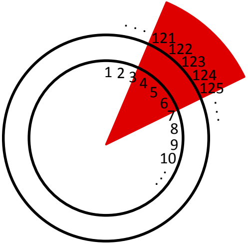
Train a MLP for each specific sector!

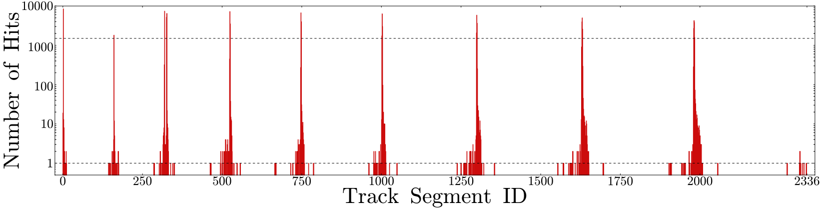
- Illuminate uniformly each sector

$\Rightarrow$ Look for active TS

- Select TS which are active in more than 15% of the events

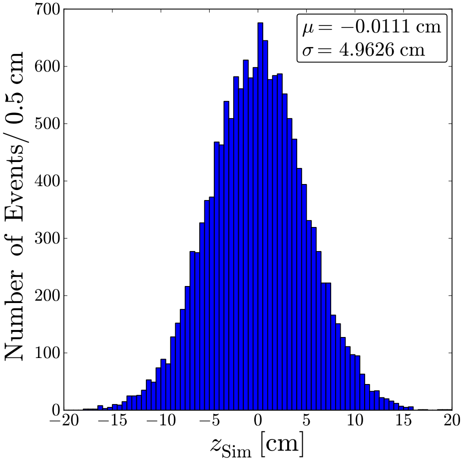
$\Rightarrow$ 16-28 Input neurons



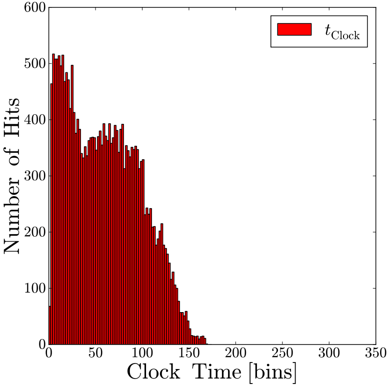


$\Rightarrow$ Drift times of selected TS $\Rightarrow$ Input values for MLP.

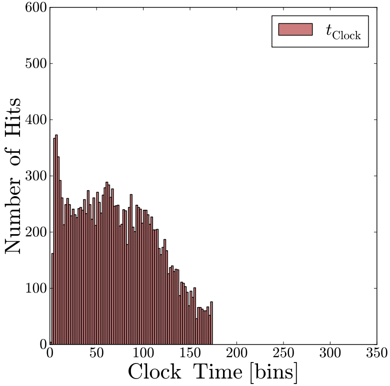
z-Vertex Distribution for all Experiments



Drift Time Distributions with TSF Resolution

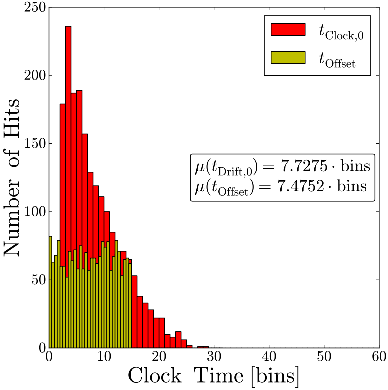


(a) No Background

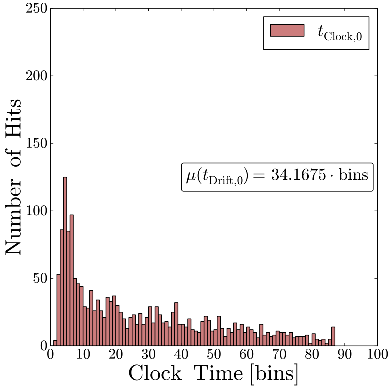


(b) Pure Background

First Hit Time Distr. with TSF Resolution



(a) No Background



(b) Pure Background