A Neural-Network Clusterisation Algorithm for the ATLAS Silicon Pixel Detector

Katharine Leney on behalf of the ATLAS Collaboration

University of the Witwatersrand, South Africa

16th May 2013
Two beams of protons circulating in opposite directions, traveling at 99.99% of the speed of light.

27km long tunnel, 100m underground.

Beam is controlled by 1800 superconducting magnets (8T) operating at 1.9 K.

Beams collide 40 million times per second.

200 billion protons per bunch.

3000 ‘bunches’ of protons per beam.
The ATLAS Detector

Diameter = 25m
Length = 40m
Weight = 7000 tonnes

16th May 2013
Katharine Leney
The ATLAS Detector
Data Events

$Z \rightarrow \mu\mu$ event + 24 other p-p interactions
Pixel Detector

- Innermost layer of ATLAS detector.
- Provides precision measurements of positions of charged particles.
- Crucial for identification of long-lived particles via reconstruction of secondary vertices and estimate of number of proton-proton interactions per bunch crossing.
- 1744 identical modules.
- 46080 silicon pixel sensors per module.
  - Thickness: 250 \( \mu \text{m} \)
  - Transverse length: 50 \( \mu \text{m} \)
  - Longitudinal length: 400 \( \mu \text{m} \)
- Total of 80.8 million read-out channels.
Tracking Overview

Track Pattern Recognition

- Start with blue “seeds” from 3 space point hits in silicon
- Use seed to build track candidate from inside out
- Use ambiguity solver to keep only most probable tracks
- Silicon track candidate connected to TRT extension

Track Fitting

- Fit track trajectory to collection of hits in track
Standard Clustering

- Particle traversing detector typically deposits charge in more than one pixel.
- Pixels with deposited charge are grouped into clusters if they have a common edge or a common corner.

\[
\begin{align*}
x_{cs} & = x_{\text{center}} + \Delta x \cdot \left( \Omega_x - \frac{1}{2} \right) \\
y_{cs} & = y_{\text{center}} + \Delta y \cdot \left( \Omega_y - \frac{1}{2} \right) \\
\Omega_{x(y)} & = \frac{q_{\text{last row(col)}}}{q_{\text{first row(col)}} + q_{\text{last row(col)}}}
\end{align*}
\]

- Charge deposited in a pixel measured using pulse-height time-over-threshold.
- Position of crossing is computed from the signal heights inside the cluster of pixels.
Tracking in Dense Environments

• Standard clustering provides excellent resolution for most clusters.

• Inadequate for dense environments with multiple charged particles:
  ‣ Charge deposited in neighbouring pixels.
  ‣ Clusters are shared.
  ‣ Track parameters are mis-estimated.

Jets with transverse momentum $> 1$ TeV typically produce merged clusters.
Tracking in Dense Environments

ATLAS for Approval

Anti-$k_t$, $R=0.4$, EM+JES Jets, $0<\eta^{\text{jet}}<1.2$

Histo=Pythia 8, Points=Data

$1000 \text{ GeV} < p_T^{\text{jet}} < 1200 \text{ GeV}$

$310 \text{ GeV} < p_T^{\text{jet}} < 400 \text{ GeV}$

$110 \text{ GeV} < p_T^{\text{jet}} < 160 \text{ GeV}$

More energetic jets $\rightarrow$ more tracks in core.
Tracking in Dense Environments

![Graph showing the sum of reconstructed tracks per bin as a function of ∆R_{trk,jet}.

**ATLAS** for Approval

Anti-κ, R=0.4, EM+JES Jets, 0<η_{jet}<1.2
Histo=Pythia 8, Points=Data

- 1000 GeV < p_{T}^{jet} < 1200 GeV
- 310 GeV < p_{T}^{jet} < 400 GeV
- 110 GeV < p_{T}^{jet} < 160 GeV

Most energetic tracks in jet core.

16th May 2013

Katharine Leney
Tracks in core have fewer hits associated to them.
Energetic tracks in jet core share more hits with neighbouring tracks.
Neural Networks

- Powerful tools for pattern recognition problems.
- Can handle non-linear correlations between input variables.
- Attractive for problems with many degrees-of-freedom.
- Inputs are differently weighted in the hidden layers of the NN to finally determine the output.

Good choice for pixel clustering algorithm:
- Many cluster properties are nearly meaningless when alone (e.g. charge of a single pixel).
- Combine cluster properties to put into context (e.g. knowing charges of adjacent pixels).
- Variables then contain all information required for successful pattern recognition.
Neural Network Cluster Splitter

Feed-forward multi-layer perceptron network:

\[ F_i(\bar{x}) = h\left( \sum_j \omega_{ij} g \left( \sum_k \omega_{jk} x_k + \theta_j \right) + \theta_i \right) \]

- \( F_i(\bar{x}) \): output node
- \( \omega_{ij} \): weight parameters
- \( g(x) = \left( 1 + \exp^{-2x} \right)^{-1} \): activation functions
- \( x_k \): input nodes/variables (\( k = [0, \text{N}_{\text{inputs}}] \))
- \( \theta_j \): threshold parameters

Characteristics:
- Output nodes are confined between 0 and 1
- Neural networks used to compute:
  - Number of particles per cluster.
  - Cluster position and error.

\[ g(x) = \left( 1 + \exp^{-2x} \right)^{-1} \]
\[ h(x) = x \]
Number of Particles Per Cluster

60 input nodes

- 7x7 pixel matrix of collected charge of each pixel
- Vector of longitudinal size of pixels in the matrix
- Direction of the candidate charged particles traversing the cluster

Hidden layer

3 output nodes

- One particle per cluster
- Two particles per cluster
- Three particles per cluster

16th May 2013

Katharine Leney
Cluster Position & Error

Additional set of neural networks used to estimate:

Cluster position:

• Configured for interpolation.
  ‣ Obtain a function where outputs get as close as possible to one or more continuous target variables.
  ‣ Exploit dependence of such targets on the input variables.
• Different neural networks for different number of particles scenarios.
  ‣ Trained on true number of particles in simulation.
• Same input variables as for classification neural network.

Probability density function for residual of estimated impact point:

• $\Delta \vec{P} = \vec{P} - \vec{P}_{\text{true}}$
• Separate neural networks for transverse and longitudinal directions.
• Translated to the cluster rest frame.
Training

• Ten neural networks needed for up to three sub-clusters:

<table>
<thead>
<tr>
<th>Number of charged particles traversing cluster</th>
<th>Particle 1 position</th>
<th>Particle 2 position</th>
<th>Particle 3 position</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Particle 1 x-error</td>
<td>Particle 2 x-error</td>
<td>Particle 3 x-error</td>
</tr>
<tr>
<td></td>
<td>Particle 1 y-error</td>
<td>Particle 2 y-error</td>
<td>Particle 3 y-error</td>
</tr>
</tbody>
</table>

• Trained on simulations of pair produced top-quarks, and highly energetic di-jet events.

• Simulations divided into test samples and training samples.
  ‣ Number of training patterns exceeds number of network parameter by at least 1000.
Cluster Splitting

Figure 3: Example for a merged cluster created by two particles using Monte Carlo simulation: The two arrows show the paths of the particles through the silicon, the centre boxes indicate the true intersection with the mid.sensor plane: The black dot illustrates the non.split cluster position, while the two stars show the estimated cluster positions after splitting: The circles indicate the according error estimates, and $p(N=i)$ denote the probabilities for the cluster to be created by $i$ particles as estimated by the neural network: Effects caused by the Lorentz angle drift in the silicon sensor have been removed in this illustration:

Wrongly split 1 particle clusters

Wrongly non-split 2 particle clusters

Figure 4: Dependence of the fraction of incorrectly split 1 particle clusters on the fraction of wrongly split 2 particle clusters in simulation: The distribution is shown both for the NN using only the cluster information and for the NN also including the track information:

4.1 Performance in Data and Simulation

As the NN was trained using simulated data, it is vital to demonstrate similar performance in data: The charge deposition pattern after the readout conversion from ToT is the main input of the NN training: Because of this the performance on data very much depends on how well the interaction of the particle with the silicon, and the signal collection is modelled by the detector simulation and digitization: Figure 6 compares the hit residuals in data and simulation in the transverse $\phi$ and longitudinal $\eta$ direction for both:

Non-split cluster position

Cluster positions after splitting

True direction of particles

True intersection with mid-plane
Split Clusters

**ATLAS** for Approval

Anti-\(k_t\), \(R=0.4\), EM+JES Jets, \(0<\eta^{\text{jet}}<1.2\)

Histo=Pythia 8, Points=Data

- 1000 GeV < \(p_T^{\text{jet}}\) < 1200 GeV
- 310 GeV < \(p_T^{\text{jet}}\) < 400 GeV
- 110 GeV < \(p_T^{\text{jet}}\) < 160 GeV

Figure 5: Mean number of pixel clusters split by the neural network pixel cluster splitting algorithm as a function of \(\Delta_{\text{trk,jet}}\).
Two-Particle Separation

• Track is allowed to share a pixel cluster with another track only if cluster is not already split, and the neural network output is compatible with a possible merged cluster.

• Most noticeable improvements in innermost layer of pixel detector (b-layer) where particle density is highest.

• Ambiguities reduced by order of magnitude when using neural network.
Cluster Resolution

- Dramatic improvement in resolution.
- Non-linear treatment of charge resolution allows recovery of single peak in track-to-hit residuals.
Track Resolution

- Improved cluster resolution leads to improved track parameters.
- 15% improvement in longitudinal impact parameter.
  - Used for identification of long-lived particles (e.g. heavy flavour quarks).
Summary

• Neural network approach used to boost detector performance and make full use of detector design potential.

• All correlations inside pixel cluster are taken into account.

• Identify and split merged clusters created by multiple charged particles.

• Sizeable improvement in track measurements, particularly in dense environments such as in jet cores.

• Non-linear treatment of charge collection improves impact parameter resolution even for isolated tracks.

• Improved two-particle separation will become even more important during future upgrades as particle density increases.
Back-Up
The ATLAS Detector
How much is an eV?

A single electron accelerated by a potential difference of 1 volt will have a discreet amount of energy, $E = qV$ joules, where $q$ is the charge on the electron in coulombs and $V$ is the potential difference in volts.

$$1 \text{ eV} = (1.602 \times 10^{-19} \text{ C}) \times (1 \text{ V}) = 1.602 \times 10^{-19} \text{ J}.$$
In a semi-conductor, the gap between the valence band and conduction band is less than 1GeV.

When high energy particle hits semi-conductor, some of the energy is absorbed by electrons which are then promoted to the conduction band.

Number of charge carriers (both electrons and holes) is increased, and so resistance decreases.
To achieve better performance used:

- The standard approach: charge sharing
- The neural network approach

Introduction

Parameters.

Visible on moderately sized clusters.

Parameters.

- Interpolating the collected charge

Cluster reconstruction is usually performed the prompt particles from high-to-hit residuals is recovered charge distribution a distribution of the collected charge.

Thanks to the non-linear treatment of the NN recognizes the non uniform These clusters are mainly due to large clusters.

Resolution on the production.

Charge sharing variables:

Independent projections

Charge sharing variables:

As a result, the NN is able to make full use of the 2D distribution of the read-out signals.

Position correction:

Position correction:

A set of neural networks trained to separate heavy flavours from particles the very dense core of high energy jets at the LHC.

Neural network based cluster creation in the ATLAS Pixel Detector

Note: The diagrams show the distribution of charge for different scenarios: Two very close particles and Three overlapping particles. The neural network approach is highlighted by the increased resolution and ability to differentiate between closely spaced particles.
Abstract

We present a novel technique using a set of artificial neural networks to identify and split merged measurements created by multiple charged particles in the ATLAS pixel detector. Such merged measurements are a common feature of boosted physics objects such as tau leptons or strongly energetic jets where particles get highly collimated. The neural networks are trained using Monte Carlo samples produced with a detailed detector simulation.

The performance of the splitting technique is quantified using LHC data collected by the ATLAS detector in 2011 and Monte Carlo simulation. The number of shared hits per track is significantly reduced, particularly in boosted systems, which increases the reconstruction efficiency and quality. The improved position and error estimates of the measurements lead to a sizable improvement of the track and vertex resolution.