Deep Neural Network based b-jet energy correction and resolution

Nadezda Chernyavskaya - ETH Zurich
On behalf of the CMS collaboration

SPS Annual Meeting, EPFL Lausanne
28 - 31 August, 2018
The Higgs boson was discovered by CMS and ATLAS in 2012.

Since then Higgs couplings to vector bosons, tau leptons and top quarks were established.

Observation of the Higgs boson decay to bottom quarks could only be announced very recently by CMS and ATLAS.

Despite large BR(58%) to bottom-quarks, it is very challenging to separate Higgs signal from large QCD background.

Reconstructing the invariant mass of the b jet pairs with a high resolution was a crucial ingredient for the observation.
A dedicated algorithm is needed to estimate the energy of b jets because b jets:

- originate from secondary displaced vertex
- have semileptonic decays (leading to undetected neutrino daughter particles)
- deposit energy over wider area than light jets

The goal is correct b jet energy and estimate jet resolution.

Improving b jet energy resolution is fundamental for analyses with b jets in the final state.
We reconstruct b jet energy using **multidimensional regression**:

- by combining all information about the tracks and individual energy deposits reconstructed by the different subdetectors of CMS
- using as **target** the true b jet energy at generator level from the simulated events

As a regressor use a deep neural network (**DNN**)
b jets from simulated events:

- Train on 100 M b-jets from top quark-antiquark pairs sample $t\bar{t}$ ($t \rightarrow bW$)
- Top quarks decay to a b quark and a W boson before hadronizing making these jets well suited as a training sample
- Loose requirements on jets: $p_T > 15$ GeV and $|\eta| < 2.5$
- **Target**: ratio of generator-level jet $p_T$ (including the neutrinos' contribution) $p_T^{\text{gen}}$ to the reconstructed $p_T^{\text{reco}}$

![Graph 1](image1.jpg)

![Graph 2](image2.jpg)
Studying jet features

- Jet kinematics
  - jet $p_T$, $\eta$, and transverse mass
- PU information
  - median energy density in event
- Jet leading track and soft lepton track
  - $p_T$ component and $\Delta R = \sqrt{\Delta \phi^2 + \Delta \eta^2}$ distance relative to the jet axis of the soft-lepton candidate
- Jet composition
  - jet energy fraction carried by em, charged, neutrals computed in four rings of $\Delta R$ between the constituent and jet axis
  - multiplicity of jet constituents
- Secondary vertex
  - $p_T$, mass and # of charged tracks associated to the secondary vertex
  - decay length and uncertainty of the secondary vertex
Regression Loss function

Loss function for DNN

• Get 2 outputs from regression: energy correction and resolution estimator.

• To get energy correction we use Huber loss:

\[
H_{\text{uber}}(y, F(x)) = \begin{cases} 
\frac{1}{2}(y - F(x))^2, & \text{for } |y - F(x)| < \delta \\
\delta |y - F(x)| - \frac{1}{2}\delta^2, & \text{otherwise.}
\end{cases}
\]

• For resolution estimator use two quantile loss functions for 25% and 75% quantiles. Quantile loss function:

\[
\rho_\tau(y, F(x)) = \begin{cases} 
\tau(y - F(x)), & \text{if } (y - F(x)) > 0 \\
(\tau - 1)(y - F(x)), & \text{otherwise}
\end{cases}
\]

Complete loss function

\[
Loss = H_{\text{uber}}(y, F(x)) + \rho_{75\%}(y, F(x)) + \rho_{25\%}(y, F(x))
\]
DNN architecture

DNN as a regressor - Feed-forward fully connected DNN

43 Inputs

3 Outputs
- energy correction
- q_{25\%} resolution estimator
- q_{75\%}

Dense Unit - linear combination of all outputs in the previous layer

Batch Normalization Unit - transformation of an input to zero-mean unit-variance output (speeds up convergence)

Drop-out Unit - operation that randomly sets to 0 given fraction of inputs (regularization)

Leaky ReLU activation

\[ \alpha x \cdot [x < 0] + x \cdot [x \geq 0] \]

(gradient is never 0)

- DNN is implemented in Keras with TensorFlow backend
- Back-propagation using stochastic gradient descent with ADAM optimizer

Nadia Chernyavskaya

DNN b-jet energy regression

31.08.2018
Optimizations

- Optimization is done for hyper-parameters (dropout and learning rate) and architectures was performed to choose the optimal:
  - 6 layers: [1024, 1024, 1024, 512, 256, 128]
  - dropout : 0.1
  - learning rate = 0.001

- Effects of training statistics:
  - $b$ jet $p_T$ spectrum spans over 6 orders of magnitude exposing the training to many more jets with lower $p_T$ than with high $p_T$
  - fraction of lower $p_T$ jets has been discarded giving more emphasis to the higher $p_T$
  - No improvement has been found for high $p_T$ jets and full training statistics is used for the final result
To evaluate the performance of the DNN:

- compare b jet resolution and scale before and after the energy correction on a statistically independent sample
- evaluate performance on large domain: b-jets from different physics processes

After DNN corrections distribution becomes narrower and its median exhibits a flatter dependence as a function of the jet $p_T$ and $\eta$.
**Results**

Quantify the improvements in **relative jet energy resolution** $\bar{\sigma}$:

- Energy resolution $\sigma$ estimated as half difference of 25% and 75% quantiles normalized to position of centroid.

\[
\bar{\sigma} := \frac{\sigma}{q_{40}} := \frac{q_{75} - q_{25}}{2q_{40}}
\]

- Improvement in jet resolution $\sim 16\%$ for b jets created in different physics processes.

- Improvement up to $p_T \sim 300$ GeV.

- 11-14% up to $p_T < 100$ GeV and down to $\sim 5-8\%$ for $p_T > 200$ GeV.

---

**MC sample**

<table>
<thead>
<tr>
<th>Process</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t\bar{t}$</td>
<td>15.74%</td>
</tr>
<tr>
<td>$Z(\rightarrow l^+ l^-)H(\rightarrow b\bar{b})$</td>
<td>16.42%</td>
</tr>
<tr>
<td>$H(\rightarrow bb)H(\rightarrow \gamma\gamma)$ SM</td>
<td>16.42%</td>
</tr>
<tr>
<td>$H(\rightarrow bb)H(\rightarrow \gamma\gamma)$ resonant 500 GeV</td>
<td>17.61%</td>
</tr>
<tr>
<td>$H(\rightarrow bb)H(\rightarrow \gamma\gamma)$ resonant 700 GeV</td>
<td>15.75%</td>
</tr>
</tbody>
</table>
DNN provides not only energy correction but also a **per jet resolution estimator**.

To check that b jet resolution estimator $\hat{\sigma}$ represents intrinsic jet resolution $\sigma$ (half difference of 25% and 75% quantiles):

- split b jets in several equally populated bins of $\hat{\sigma}$
- in each bin compute the intrinsic jet resolution $\sigma$
- check the correlation between $\sigma$ and the mean $\hat{\sigma}$

**Liner dependence**

The same test was performed in different bins in jet $p_T$ and the deviations from linear behavior do not exceed 20% of $\hat{\sigma}$ value.
Conclusions

• DNN was trained to provide energy correction and resolution estimator for b jets arising from pp collisions at the LHC
• DNN regression was trained based on jet-composition information
• The algorithm improved the b jet resolution by roughly 15% compared to baseline
• Improvement in energy resolution and the knowledge of per jet resolution is used in the physics analyses with b jets in the final state in the CMS (including observation of $H \rightarrow b\bar{b}$)

Thank you!
Additional Material
NN architecture: Feed-forward fully connected NN

NN architecture:

- **Inputs**: 43 Inputs
- **Batch normalization**: layer between the input and the first hidden layer
- **Each hidden layer has 4 operations**:
  1. Dense
  2. Batch Normalization
  3. Dropout
  4. Leaky ReLU activation with $\alpha = 0.2$
- **Target**: normalized to zero-mean unit-variance
Optimizations

- Optimization is done for hyper-parameters (dropout and learning rate) and architectures was

- Random sampling of 50 points for training from the grid used for optimization:
  - dropout: [0.1, 0.2, 0.3, 0.4]
  - lr: [0.01, 0.001, 0.0001, 0.00001, 0.000001]
  - layers with nodes:
    - [512, 256, 128], [1024, 512, 256, 128],
    - [1024, 1024, 1024, 512, 256, 128],
    - [1024, 1024, 1024, 1024, 1024, 512, 256, 128]

- To avoid biasing the training toward training/validation samples cross-validation with 5-folds is used.
• Evolution of mean validation error with 5-folds versus epochs for different architectures

• The best set of hyper-parameters is chosen
  * 6 layers: [1024, 1024, 1024, 512, 256, 128]
  * dropout : 0.1
  * learning rate = 0.001

=================================================================================
Total params: 2,845,743
Trainable params: 2,837,721
Non-trainable params: 8,022
• **Model performance scaling with complexity**

• Only small difference in performance in validation sample for different complexities (number of trainable parameters of the NN)
Jet resolution estimator

- Resolution estimator represents well jet resolution independently of $p_T$
- Nice dependence \textit{inclusively} and \textit{in different bins of $p_T$}

\begin{align*}
60 < p_T < 70 \text{ GeV} & \quad 80 < p_T < 90 \text{ GeV} & \quad 100 < p_T < 110 \text{ GeV} \\
\end{align*}

\begin{align*}
\text{Work in progress} & \quad \text{Work in progress} & \quad \text{Work in progress}
\end{align*}