Lecture 15

Last time: top tagging
- HLF cut based
- HLF QNN
- QNN on 4-vectors of constituents
- QNN on jet images

Current SOTA: top tagging community comparison paper (link to arXiv on website)

Winner was top tagger based on graph neural networks
and placed as a CNN based on ResNet architecture

"Point clouds" for self-driving cars

https://rutersconnect-my.sharepoint.com/personal/shih_physics_rutgers_edu/_layouts/15/Doc.aspx?source=0c21e841de-3e15-43ad-8ded-ca103e6850f1&action=edit&wd=target%28Physics 694 Lecture 15 one%7C%22
Brief description of graph NNs

What is something permutation in t

Represent each data instance as a graph

each node is a constituent

general idea of GNN: learn set of connecting btw nodes

map from graph \rightarrow output

A GNN (graphs) = output.

Specific implementation for top tagging

\hat{\mathbf{x}}_i \in \mathbb{R}^F (F = 3 initially (F=2,1,0))

\hat{\mathbf{x}}_j \in \mathbb{R}^{F'}

jet \subseteq \{ \text{const}_1, \text{const}_2, \ldots \}

but order doesn't matter!

for each node, consider k nearest neighbors in \((x, \phi)\)

train to learn "edge conv." for h \in shared weights across all nodes

\hat{\mathbf{x}}_i = \sum_{j=1}^k h (\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \sum! 

permute in t
EdgeConv produces a new graph! & is stackable.

PatrickNet

1902.08570

An & Constantin

original jet constituents

EdgeConv

EdgeConv

EdgeConv

global avg policy

CNN

softmax
Beyond Classification - Decoloration

In practical applications of jet tags, raw performance not the only consideration.

- Many applications, jet reserved for sideband interpolation \(\rightarrow\) background estimation.

\[ \text{ACD} \rightarrow \text{jet} \]

- If tagger "learns" \(\text{jet}\) \(\rightarrow\) cut on classifier \(\rightarrow\) sculpt a bump in background \(\text{jet}\) distribution.

\[ \text{ACD} \rightarrow \text{jet} \]

Cannot combine tagger w/sideband method in \(\text{jet}\).
Wait! Classifier independent (statistically) of jet in background.

"AI fairness/bias"

"Domain adaptation"

ML training data ≠ application data

Need to carry over features from training data → other domain.

Commonly used metric: Jensen-Shannon Divergence (between two distributions).

$
\text{JS}(p, q) = \frac{1}{2} \left[ \text{KL}(p || m) + \text{KL}(q || m) \right]
$

| Measure of similarity between two distributions. |
| Can be computed from histograms. |
\[ JSD(P, \alpha) = \frac{1}{2} \left( KL(P \| M) + KL(Q \| M) \right) \]

\[
M = \frac{P + Q}{2} \text{ avg dist.}
\]

\[
KL(P \| M) = "KL divergence"
\]

\[
= \int dx \, P(x) \log \frac{P(x)}{M(x)} \text{ not symmetric in P \& M}
\]

\[
KL(P \| M) \geq 0 \text{ \& } \lim_{P \to M} = 0.
\]

(Easy to prove!)

Symmetrized KL divergence

\[0 \leq JSD \leq 1\]

\[
0 \text{ if } P = Q \text{ \& have no overlap.}
\]

\[
JSD \to 1 \text{ as } P \leftrightarrow M
\]
Need to choose a cut to define $JSD_0$ — common choices are cut at 50% signal eff. (30% 20% 10%)

$R = \frac{1}{fpr}$ rejection factor

$R_{50}$ rejection @ 50% signal eff.
How can we achieve decorrelation?

1. Modifying training data to remove correlations
   i. How about removing jet from list of features in training data?
      In general this doesn’t work!
      - Other features are highly correlated w/ jet
      - For deep NNs, low-level info like const, 4-vec or jet images
        m_{jet} is not explicitly provided!
   ii. Hand engineer features that are indep. of m_{jet}

\rightarrow \text{example} D0T - Designed Decorrelated Tagger

\text{\textit{doesn’t generalize to} ML w/ many features.
"Planning" - reweight sig & bg met dists to look like each other.

i.e. train classifier w/ weighted events 
\[ w(x, y) \]
on features \( x \)

- other works well in practice
- but not guaranteed
- removes "leading order" mass information \( \rightarrow \) guarantees marginal dists are sam
- doesn't generalize well
- to multiple features

\[ \frac{P_{ij}(x, m)}{P_{ij}(m)} \rightarrow \frac{P_{ij}(x)}{P_{ij}(m)} \]

\[ \text{classifier}(x) \text{ independent of } m \]