Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data

Laura Domine (Stanford / SLAC)
Kazuhiro Terao (SLAC)

arXiv:1903.05663

ACAT 2019
Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
Particle Image Analysis with LArTPCs

Liquid Argon Time Projection Chamber (LArTPC) = particle imaging detector

~3mm resolution

Wire LArTPC (2D projections)

Neutrino interaction candidate from MicroBooNE experiment @ Fermilab

Pixel LArTPC (native 3D)

Cosmic rays in a 3D LArTPC charge readout (arxiv:1808.02969) @ LBNL
Particle Image Analysis with LArTPCs for neutrinos

Neutrino detectors & LArTPCs

Goal: Extract $\nu$ flavor + energy
Particle Image Analysis with LArTPCs for neutrinos

Neutrino detectors & LArTPCs

Goal: Extract $\nu$ flavor + energy
Convolutional Neural Networks

Now state-of-the-art technique in computer vision for complex image analysis tasks:
Convolutional Neural Networks

Now state-of-the-art technique in computer vision for complex image analysis tasks:

Object detection & classification

Semantic segmentation
Semantic segmentation of LArTPC data (2D)

Data (network’s input)

Predictions (network’s output)

Predicted as particle track

Predicted as electromagnetic shower

A Deep Neural Network for Pixel-Level Electromagnetic Particle Identification in the MicroBooNE Liquid Argon Time Projection Chamber. (arxiv:1808.07269)
Semantic segmentation of LArTPC data (3D)

2-classes (particle track vs electromagnetic shower) pixel-level segmentation on 512px 3D images.
Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense UResNet
LArTPC data is sparse, locally dense
LArTPC data is sparse, locally dense

CNNs rely on dense matrix multiplications

All pixels are meaningful for CNNs.

<1% of voxels are non-zero in LArTPC data

Zero voxels are meaningless!
LArTPC data is sparse, locally dense

- **Dense**
  - Computation waste
  - Potential performance degradation

- **Sparse (but locally dense)**
  - <1% of voxels are non-zero in LArTPC data
  - Zero voxels are meaningless!

**CNNs** rely on dense matrix multiplications

All pixels are meaningful for CNNs.
Submanifold Sparse Convolutions

Many possible definitions and implementations of ‘sparse convolutions’...


State-of-the-art on ShapeNet challenge (3D part segmentation)
Submanifold Sparse Convolutions

Submanifold = “input data with lower effective dimension than the space in which it lives”

Ex: 1D curve in 2+D space, 2D surface in 3+D space

Our case: the worst! **1D curve in 3D space...**
Submanifold Sparse Convolutions *aim to solve*...

1. **Resources waste** of dense convolutions on sparse data
2. **Dilation problem**
   - 1 nonzero site leads to $3^d$ nonzero sites after 1 convolution
   - How to keep the same level of sparsity throughout the network?
Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense UResNet
   a. Dataset, task, metrics & network architecture
   b. Results
Dataset & Task

Total: 100,000 simulated 3D images
Spatial size: 192px / 512px / 768px (~3mm/pix)

Semantic segmentation with 5 classes

- Heavily ionizing particles (protons)
- Minimum ionizing particles (muons and pions)
- Electromagnetic shower
- Delta rays
- Michel electrons

Publicly available: https://dx.doi.org/10.17605/OSF.IO/VRUZP
Metrics

- **Non-zero accuracy**: fraction of correctly labeled pixels, i.e. 
  \[
  \frac{\text{# nonzero voxels whose predicted label is correct}}{\text{# nonzero voxels}}
  \]
- **GPU memory** (hardware limitation)
- **Computation wall-time**
Network architecture: U-ResNet

U-ResNet = U-Net + ResNet
Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense UResNet
   a. Dataset, task, metrics & network architecture
   b. Results
Sparse U-ResNet trains 119x faster

Sparse = 10h
Dense = 8.8 days

~98%
~92%
Sparse U-ResNet allows larger batch sizes

@batch size 88 sparse uses 93x less memory than dense and computation is 3x faster

At train time [2D, 512px, 5-16]

16Gb = max. memory (P100/V100 GPU @ HPC)

= 210 images fits the whole MicroBooNE detector
Learning strategies

Sparse U-ResNet learns more uniformly across pixels than dense U-ResNet
Summary
Submanifold sparse convolutions...

- Run faster
- Use less GPU memory
- ... and outperform standard convolutions.

Better performance and better scalability!
https://arxiv.org/abs/1903.05663

Reproduce our results / start using SSCN:

- Open dataset: https://osf.io/vruzp/
- Software containers: https://www.singularity-hub.org/containers/6596
- Code on Github: https://github.com/Temigo/uresnet_pytorch
Backup
Network architecture: UResNet

UResNet = U-Net + ResNet
Network architecture: UResNet

UResNet = U-Net + ResNet

Encoder

Decoder

Residual connections

Concatenation

input

conv

conv-s2-finc

tconv-s2-fde

conv-fdec

softmax

Residual connections

Concatenation
Network architecture: UResNet

UResNet = U-Net + ResNet
Network architecture: UResNet

UResNet = U-Net + ResNet
Sparse U-ResNet learns more uniformly

Std of mean EM shower softmax score in an image, across training iterations
Inference duration

![Graph showing inference duration vs. batch size for Sparse and Dense models. The graph plots forward time in seconds on the y-axis against batch size on the x-axis. The Sparse model has a lower forward time than the Dense model for all batch sizes.](image-url)
Learning from mistakes: the case of Michel electrons
Learning from mistakes: the case of Michel electrons
Learning from mistakes: the case of Michel electrons
Learning from mistakes: the case of Michel electrons

Primary Ionization

Gamma rays
## Relabeling study

*Relabeled dataset = only primary ionization is labelled as Michel electrons*

<table>
<thead>
<tr>
<th></th>
<th>Train data</th>
<th>Test data</th>
<th>Test data</th>
<th>Test data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Regular</td>
<td>Relabeled</td>
<td>Relabeled + Weighting</td>
</tr>
<tr>
<td>HIP</td>
<td></td>
<td>98.0%</td>
<td>98.1%</td>
<td>98.1%</td>
</tr>
<tr>
<td>MIP</td>
<td></td>
<td>99.4%</td>
<td>99.2%</td>
<td>99.4%</td>
</tr>
<tr>
<td>EM shower</td>
<td></td>
<td>99.4%</td>
<td>97.9%</td>
<td>99.2%</td>
</tr>
<tr>
<td>Delta rays</td>
<td></td>
<td>85.7%</td>
<td>94.8%</td>
<td>96.0%</td>
</tr>
<tr>
<td>Michel electrons</td>
<td></td>
<td>56.6%</td>
<td>94.4%</td>
<td>94.7%</td>
</tr>
</tbody>
</table>
Number of pixels in candidate vs matched Michel cluster
Michel electrons energy spectrum reconstruction

**Sample size**: 7105

**Identification purity**: 98.8%

**Identification efficiency**: 93.9%

**Cluster efficiency**: 96.1%

**Cluster purity**: 97.3%

**First ML-based approach**
Simulation only, next step is data!