Machine Learning
Monopoles
and MoEDAL

Jonathan Hays
Adrian Bevan, Tom Charman, Krzysztof Furman
Outline

MoEDAL and Monopoles
Convolutional Neural Networks
Preparing Training Data
Performance
Summary and next steps
MoEDAL and Monopoles

MoEDAL = The Monopole & Exotics Detector at the LHC

Search for highly ionizing particles such as magnetic monopoles and other exotic avatars of new physics

Nuclear track detectors and aluminium trapping detectors

18/08/2017
Jonathan Hays, j.hays@qmul.ac.uk

Queen Mary
University of London
MoEDAL and Monopoles

MoEDAL = The Monopole & Exotics Detector at the LHC

Search for highly ionizing particles such as magnetic monopoles and other exotic avatars of new physics

Nuclear track detectors and aluminium trapping detectors

Highly ionising particles passing through the NTD cause microscopic damage to the polymer that can be reveal with etching

18/08/2017

Jonathan Hays, j.hays@qmul.ac.uk
MoEDAL and Monopoles

MoEDAL = The Monopole & Exotics Detector at the LHC

Signal:
- Set of characteristic O(micron) sized holes that line up across multiple sheets

Small holes in large sheets:
- high resolution imaging needed
- large images to scan

Analysing the images could be susceptible to an automated approach using machine learning techniques
Convolutional Neural Networks

Deep artificial neural network inspired by biological function – has been successfully applied to image processing and categorization tasks.

Simple example consists of multiple layers of convolution and pooling before feeding outputs into a traditional fully-connected MLP with one or more output neurons.
Convolutional Neural Networks

Deep artificial neural network inspired by biological function – has been successfully applied to image processing and categorization tasks

Challenges in this context:
- image preparation and resources – how to deal with very large images
- optimizing network structure
- supervised learning technique needs (lots of) training data
Convolutional Neural Networks

image preparation and resources – how to deal with very large images

tiling of the input images and consideration of how to handle multiple layers

optimizing network structure

investigate different structures and variation in performance

Supervised learning technique needs (lots of) training data

Potential solutions include:

- lots of real examples of etched sheets with signals simulated with beams
- artificial samples generated using Monte-Carlo based simulation
- resample small collection of real examples for training

18/08/2017  Jonathan Hays, j.hays@qmul.ac.uk
Learn how to use CNNs to analyze images using API developed by A. Bevan using Tensor Flow from Google.

Investigate resampling as a technique to generate large sample of training images.
Resampling

Start from just two images of some holes

Select regions to sample from for background and signal

Randomly build new image from samples
Sampling regions

Regions outlined in red are background, green are ROI.

18/08/2017
Jonathan Hays, j.hays@qmul.ac.uk
Example images

Generated images have size: 1536 x 1536
Tiles also include a random rotation (integral numbers of 90 degree rotations)

This patch size: 128 x 128

Examples of images made by the sampling algorithm.
Jonathan Hays, j.hays@qmul.ac.uk
Training Workflow

Generate 1000 signal and 1000 background images

Split 50/50 into training and test samples

Process batches of 50 images at a time through the network and update weights using the ADAM optimiser (arxiv:1412:6980)

Repeat for fixed number (200) of training epoch

Training run on GPUs: NVidia Tesla K40 NVidia GeForce 1080 Ti (each with 12GB of memory)
Challenge: memory!

Challenge:
Running with 1536x1536 images: completely exhausts the available memory and training fails!

Solution:
down-sample the images

Training run on GPUs:
- NVidia Tesla K40
- NVidia GeForce 1080 Ti

(each with 12GB of memory)
Downsampling

Training run on GPUs:
NVidia Tesla K40
NVidia GeForce 1080 Ti
(each with 12GB of memory)
Performance: Timing

Note: 1080 Ti much faster for these jobs than K40 (and much cheaper)
Performance: Accuracy

Compare each prediction with its label for all 1000 images in the testing/training set.

Accuracy = \frac{\text{Correct guesses}}{\text{Images in set}}

Threshold = 0.5

18/08/2017
Jonathan Hays, j.hays@qmul.ac.uk
Performance: ROC

Vary threshold and generate curve of true positive rate versus false positive rate

Not terribly informative – accuracy is very high since this example problem is very easy for the CNN to deal with
Aside: TensorBoard

TensorBoard: tool within the TensorFlow suite that lets you visualize the machine learning process

Interactive view of the computational graph – can drill down to fine details
Aside: TensorBoard

TensorBoard conv layer outputs. Left smallest images (48 x 48), right largest (768 x 768).

Not only do smaller images seem to train just as accurately as larger ones, they are also quicker.
(Note that tailoring the size of the kernel filters in the convolutional layers to the size of the image could lead to improved performance.)
Summary and Future Steps

Training data generated using resampling of real data

CNN successfully finds signal in simplified model

Future steps:
- Look at more realistic images
- Develop techniques to handle very large images
- Simultaneous processing of double sided scans