Preliminary Tutorial Programme

National CDT Meeting, Edinburgh

There are four sessions over the day and a half of the meeting. Each session is 90 minutes. You can register for the sessions you want to go to here: https://indico.cern.ch/event/770601/

Session 1: Tuesday 20th November 10:30

Tutor	Room	Topic
Rita Tojeiro	2	Planning your data science skills
Duncan Forgan	3	Git & Github
Deepak Mahtani	4	Getting started with Jupyter notebooks and machine learning
John Armstrong	5	Classical unsupervised learning with scikit-learn

Session 2: Tuesday 20th November 14:00

Tutor	Room	Topic
David Henty	1	Parallel Programming
Rita Tojeiro	2	Planning your data science skills
Duncan Forgan	3	Git & Github
Isaac Roseboom (DeltaDNA)	4	Data Science in the cloud
Duncan Little (ASOS)	5	Data Science & Machine Learning in e-Commerce

Session 3: Wednesday 21st November 09:00

Tutor	Room	Topic
David Henty	1	Parallel Programming
Steven Bamford	2	Neural Networks in TensorFlow
Duncan Little (ASOS)	3	Data Science & Machine Learning in e-Commerce
Stewart Martin-Haugh	4	Code profiling & optimization
Eric Tittley	Plenary	GPU Programming

Session 4: Wednesday 21st November 11:00

Tutor	Room	Topic
Steven Bamford	2	Neural Networks in TensorFlow
John Armstrong	3	Classical unsupervised learning with scikit-learn
Stewart Martin-Haugh	4	Code profiling & optimization
Eric Tittley	5	GPU Programming

Data Science & Machine Learning in e-Commerce

Duncan Little, ASOS

Big data in retail and e-commerce: what, why and how.

- Common machine learning methods in retail and e-commerce (recommender systems, customer lifetime value prediction, automatic product understanding)
- Practical aspects of deploying and using ML in retail and e-commerce (using large distributed computing systems such as Spark for example)
- 'Soft skills' for data scientists: stakeholder management, understanding business value, expectation management.

Code profiling & Optimization

Stewart Martin-Haugh,

The faster our code runs, the more data we can process. In this tutorial you, will learn how to measure and improve CPU and memory performance in Linux.

Classical unsupervised learning with scikit-learn

John Armstrong

Unsupervised learning is a subsection of machine learning where the computer is trained with unlabelled data and must pick out important features in the data on its own. In terms of classical machine learning this boils down to two main approaches: dimensionality reduction and clustering. We will look at these two concepts and apply them practically to a dataset to test how good the computer's intuition is.

Data Science in the Cloud

Isaac Roseboom

Cloud computing has moved quickly from simply providing easy access to hardware to supplying easy to use services that eliminate the need for any sys admin knowledge. ML and data science has been one of the key drivers of this, with all the major cloud providers (i.e. AWS, GCP and Azure) offering a suite of tools to allow the construction of everything from data pipelines to ML model fitting to APIs to categorise data in real time and everything in between.

In this session I will go over what is available in the cloud for data scientists and talk through some typical cloud tech stacks you will encounter in the world of big data. Finally I will talk about the pros and cons of different cloud technologies and how they apply to different real world applications.

Getting started with Jupyter notebooks and machine learning

Deepak Mahtani

Machine learning is one of the biggest buzzwords in the field of data science and it has many applications within both academia and industry. In this tutorial, Pivigo's community manager and a data scientist will take you through the basics of using Jupyter notebooks and how to get started with machine learning. Jupyter notebooks are a fantastic way to explore data and to conduct experiments on the data. He will be using the titanic dataset on Kaggle.

Neural Networks in TensorFlow

Steven Bamford

This workshop will briefly introduce supervised deep-learning with neural networks, before walking through an example of constructing and training such networks using keras, a high-level Python interface to TensorFlow. We will train both fully-connected and convolutional neural nets to distinguish hand-written digits from the standard MNIST dataset, and validate the results. I will also discuss the importance of understanding and preparing your training data, and illustrate the benefits of data augmentation.

GPU Programming

Eric Tittley

Graphics Processing Units (GPUs) are commonly available computing devices designed to enhancing computer game experiences. The underlying hardware can, however, be exploited to perform general calculations and has given rise to General-Purpose GPU (GPGPU) computing. In this tutorial, I will discuss 1) when it might be advantageous to develop code to run on a GPU, 2) the nuances of GPU hardware that affect the algorithm ported to the GPU, compared specifically with other forms of parallel programming, and 3) examples of GPU programming with CUDA, nVidia's extension to C/C++, highlighting ease and indicating pitfalls. Knowledge of C/C++ is advantageous, but not essential.

Version Control with Git

Duncan Forgan

I will quickly introduce the concept of version control, and distributed version control systems (DVCS), of which the most famous is git. We'll learn how to use git from the command line to create repositories which keep track of our projects (be they code, documents or otherwise). I'll end by describing how to store your git repositories on a remote server, the most famous of which is GitHub. If we have time, we'll get into more advanced topics such as branching.

Planning your data science career

Rita Tojeiro

This is an interactive session, designed to explore the skills and tools that a successful data-scientist should possess. By the end of the session you should be better equipped to choose taught modules to suit your specific needs, and have a better understanding of the many aspects and scope of “data-science”. No computers, just post-its.

Parallel Programming

David Henty

High-performance computing (HPC) is a fundamental technology used in solving scientific problems. Many of the grand challenges of science have depended on simulations and models run on HPC facilities to make progress, for example: protein folding, the search for the Higgs boson, and developing nuclear fusion.

In this short session I will explain why HPC is synonymous with parallel computing, which involves using many CPUs at the same time to solve a common problem. I will also explain the basic tools and programming techniques used to run scientific applications on large, parallel supercomputers.