Machine Learning Carpentry

Colin Sauze
<cos@aber.ac.uk>

Links to material:
https://tinyurl.com/ml-ccmcr19

Vague Plan

- Background
- What should/does this course cover?
- My experience of teaching it
- Current issues
- Things to do:
  - Write some personas
  - Write some concept maps
  - Find a good dataset
  - General bug fixing
- Future development

Background

- I’m seeing demand for ML/AI training from:
  - Geographers doing remote sensing
  - Plant biologists with pictures of plants/bits of plants
  - Computer Scientists doing medical imaging
  - And a few others
- Mixture of backgrounds
- Want to know how to apply ML to their problems
  - Use latest libraries

More urgent need

- PI: “Can you teach something on data science, python and AI to some DProf students over easter?”
- Me: “I can do a Software Carpentry Python Course. I was thinking about creating an AI course but it won’t be ready in time, how about we do that in September?”
- PI: “I’m sure you can come up with something, it only needs to be half a day”
- Me: “erm ok, it might be a bit rushed and basic”
- PI: “Basic is fine, i’m sure it will be ok”
- Me (to self): “What have I just let myself in for? I better write something.”
- The result: https://tinyurl.com/ml-carpentry

My Vision

- Carpentry style lesson on machine learning
  - Half or whole day
  - Practical “teach me enough to do something useful”
  - Avoid too much background theory
  - Bust hype, show what is/isn’t realistically achievable
- Using a common dataset
- Using a single library
  - I chose Scikit Learn + Python

What to cover?

- Core ideas:
  - ML is subset of AI
  - ML builds predictors/models and classifiers
- What is ML?
  - Examples of ML in everyday life and in research
- Linear regression models
  - Testing accuracy with test/training data
- Logarithmic and polynomial regression models
- Clustering
  - Unsupervised learning example
  - Kmeans and spectral

What to cover (part 2)

- Neural networks:
  - Perceptrons
  - Perceptron Learning Algorithm
  - Multilayer perceptrons, backpropagation
  - Cross Validation
  - Deep learning image recognition via Cloud API examples
- Ethics
  - Bias in training data
  - Explaining algorithmic decisions

What does this miss?

- What important things weren’t covered?
- What shouldn’t be included?
- Is the order sensible?

My experience of teaching this

- Ran this course once during Easter 2019
- Around 20 students
- Half day
- Python Gapminder course run over two half day sessions in prior days

Course feedback

- 15 people filled out pre-workshop survey
  - 10 biologists, 1 mathematician, 1 geographer, 1 business, 1 computer science
  - 12 PhD students, 3 postdocs
- Reasons for coming:
  - “hopefully to understand be basics behind machine learning, which is the basis of my PhD analysis”
  - “understand what ML can do for my optimisation problems”
  - “Get the general idea of the topic”
  - “To know initial implementation of Machine learning”
  - “analysing data”
  - “new knowledge in teaching environment”
  - “renewing my skills”

Pre-workshop survey, existing knowledge.

- Which of the following can you do already?
  - use a linear regression: 6 no, 4 maybe, 5 yes
  - use logarithms or polynomials to model non-linear data: 9 no, 2 maybe, 4 yes
  - measure the error between a model and test data: 7 no, 5 maybe, 3 yes
  - find clusters in data using a technique such as kmeans: 10 no, 5 maybe
  - train a neural network to classify data: 13 no, 2 maybe
  - cross validate a machine learning algorithm's output: 12 no, 3 maybe

Post Workshop survey

- Only 5 responses
- Comments:
  - The workshop was interesting, although I think the time was not enough. I was more interested in the last bit neural networks. I believe that topics as regression needed to be shorter, expecting most of the people are ok in statistics. And it will be good next time to have more time for NN.
  - The last part of the workshop was quite fast(Neural Networks). You could have Machine Learning Workshop2 to go in depth for CNN and bit of practical for Deep Learning Such workshop would be great.
  - More time on neural nets
  - I enjoyed experiencing the carpentry from the other side. I particularly liked the perceptron bit. It was just a shame to have to rush the clusters and NN stuff. I reckon I could have a good stab at using sklearn to building a NN model now.
- From the helpers:
  - “Jyputer notebooks would be well suited to this.”
  - “I really like the carpentry style of teaching and that’s why I volunteered to help”

Post workshop survey

- How much of the information was new to you?
  - all of it: 1
  - some of it: 1
  - half of it: 2
  - none: 1

Current Issues

- Half a day wasn’t long enough
  - Had to rush neural nets and ethics section
- Old version of Anaconda on our public workstations.
  - Version of sklearn didn’t have the handwriting digits data.
- Linear regression and perceptron sections have full code implementations and sklearn versions.
  - Is this a good idea?
  - Not everything is library magic
  - Un-necessary detail?
- Perceptron learning algorithm exercise confused people
- All exercises need improving

Wider issues

- No narrative of a researcher solving a problem
- No common dataset
  - Gapminder + Worldbank for regression
  - Random data for clustering
  - Handwriting digits for neural networks
- A bit of deviation from standard Carpentries formula
  - Intro section: Use of some videos + Links to some papers
  - A bit more background
  - Some maths knowledge required
  - Somewhat more advanced material?

Activity 0: Concept maps

Activity 1: Write some personas

- Name
- Research discipline
- Type of data
- Type of analysis needed
  - Model or classify?
- Types of algorithms
- Existing programming knowledge

Activity 2: Datasets

- Is there a common dataset that works across the entire lesson?
- Might a typical researcher use it?
- What kind they want to discover?

Activity 3: Test/debug a section

- https://tinyurl.com/ml-carpentry
- Take a section or two per table
- Pull request or github issue any problems
- Known bugs:
  - Formatting of Python code is incorrect, no colour highlighting

Activity 4: What's missing?

- What techniques/algorithms should be included?
  - Random Forest?
  - Deep Learning
    - Caffe/Keras/Tensorflow?
    - How to do GPU support?
      - Google Collab?
- Can the material be more generic and platform agnostic?
- Switch to Jyputer Notebooks

Activity 5: What next?

- Any volunteers to become maintainers/developers?
- Anyone want to run it themselves?
- Infrastructure for development:
  - Mailing list, slack channel, github page etc
- How do we make it an official lesson?
- Has this been done before? Are we duplicating effort?