Hello,
and once again welcome to the “Getting Started With Machine Learning” -workshop next Wednesday!

It took a little longer than I expected to write this, but as I promised, here’s some preliminary material for the workshop. Please take an hour or so to go through this email and some of the links it contains. If you don’t have the time to process everything, just make sure that you have all of the software installed, that’s the most important part to save time during the event.

Machine learning is not the easiest topic in the world, but neither is it something you couldn’t wrap your head around with some work. If there’s something in this mail or in the links you don’t understand, don’t worry, we’ll be helping you in the workshop.

But first things first. The event will be held on Wednesday February 7th at Hub Turku (http://hubturku.com/en/co-working-in-the-heart-of-the-city/) at Eerikinkatu 7. The tentative schedule for the workshop is:

SCHEDULE
18:00, Doors open. Drinks.
18:15, Introduction by the organizers and the coaches
18:30, Basics of Machine Learning by Ilmari Ahonen (Vincit)
19:30, Break
19:45, Quick go-through of the workshop project
20:00+, Split into groups to work on the project

THIS IS A BRING-YOUR-OWN-LAPTOP EVENT, so before the workshop you should have the following programs installed on your machine (everything necessary is available for Windows, OSx and Linux, you can use the operating system you’re most comfortable with):

**Python**. (https://www.python.org/download) The machine learning code we'll be using is available for both Python2 and Python3, so you can use either one. I'll be using Python3 myself, so if you don't have a strong preference, I'd suggest that.

There’s a distribution called **Anaconda** that contains most of the Python stuff (+ additional resources) you will ever need for machine learning. So if you don’t already have Python and a packaged manager for Python installed, you should definitely get that from https://www.anaconda.com/distribution/.

As we are mostly running readymade code, you don't have to know Python. Also, the key concepts are explained in the workshop. However, if you want to get a feel of the language, you can check out these short programs https://wiki.python.org/moin/SimplePrograms.
There’s tons of other learning resources too, you can find links from python.org.

Alongside with Python, R is a go-to programming language for ML / data science. Some of the linked tutorials below use R. You can download R (https://www.r-project.org/) and R Studio IDE (https://www.rstudio.com/) but these are not required in the workshop. They are also included in Anaconda.

**Git** (https://git-scm.com/)**.** This source control program has a steep-ish learning curve, but it's also the de facto standard in any kind of development, machine learning included. We'll be using it in the workshop. Unless you're familiar with the command line git, install a git GUI too. I'm using Sourcetree (https://www.sourcetreeapp.com/) myself and can recommend it.

You only need to use the basic git functionality, so if you have ever worked with any kind of version control, you should be fine. If you're a complete newcomer to the world of version control, check the beginners tutorials from Atlassian git help pages https://www.atlassian.com/git/tutorials (or alternatively from GitHub https://guides.github.com/).

**Code editor** of your choice. I'd suggest Sublime Text (https://www.sublimetext.com/) or Visual Studio Code (https://code.visualstudio.com/), but other options are available. Again, use the one you're comfortable with.

THE ACTUAL PROJECT we'll be going through together involves handwriting recognition, or more precisely recognizing handwritten numbers. This is done by building a machine learning model called neural network. We'll be using MNIST database of handwritten digits (http://yann.lecun.com/exdb/mnist/) to train and test our machine learning model. The end result will recognize digits with a good accuracy.

The workshop project is following the Chapter 1 of the book "Neural Networks and Deep Learning" by Michael A. Nielsen. You can browse through the chapter here: http://neuralnetworksanddeeplearning.com/chap1.html. This will probably take a couple of hours to go through, so you can just skim through it or skip it altogether. Ee'll be covering the key parts in the workshop. You can find the associated code from Github at https://github.com/mnielsen/neural-networks-and-deep-learning.

The reason we use this project is as is stated in the book: It hits the sweet spot between being complicated and interesting but not so complicated to require PhD level solutions or extensive computer power. Also, it gives you a good start to dive deeper into deep learning, the machine learning method *de jour*.

Don't be intimidated by the math in the text. Going through it is not required to implement this machine learning algorithm in practice. As the author states, it's more of a technical detail than a barrier to understanding. You will do just fine even if it's been a while since your last maths exercise.

We’re going to divide into groups to work on this project. The coaches will help the groups to get started and guide them through the key concepts and help with questions and problems. Here’s a rough outline how we are planning to proceed:

- Understand what a neural network is (and isn't) and what it does.
- Understand different (hyper)parameters of a neural network.
- Understand how to evaluate the performance of a machine learning model, neural networks included.
- Run "plain" neural network code in Python.
- Try out different network parameters to see the effect on its performance.
- Explore how the neural network handles your own handwriting.
- If you have time, try out neural networks with Keras (https://keras.io/).

Now, you might ask why we finish rather than start with Keras, isn’t this *the* framework for this kind of stuff? I think it's good to get some intuition on how this stuff actually works before moving up one level of abstraction. So that's why we are using a relatively low-level code here. It also gives you a better starting point to learn more about (deep) neural networks.

After we have completed this project (it’s mainly running readymade code), the obvious question is where to go next?

Depending on how quickly you finish the neural network exercise, you’ll probably have some time to get started with some other project. The coaches can help you with these too! You can for example check some of these projects: https://elitedatascience.com/machine-learning-projects-for-beginners. For more in-depth machine learning resources / tutorials, you can check https://github.com/ujjwalkarn/Machine-Learning-Tutorials. That’s a huge list though, so it could feel a bit overwhelming.

The most important thing is to select a topic that interests you! Remember that machine learning is much more than neural networks and deep learning. More important than the method or algorithm used is that you get good results out of your project.

As a final note, if all this looks scary complicated, don’t worry! We took in a limited number of participants for this exact reason, so that we can provide each one of you with personal assistance when needed. Also, we are working in groups and the idea is to help each other out, which is always a good way to learn!

If you got any questions, shoot! I’m quite busy the next week, but Il try to answer in a reasonable time.
Cheers,
Lasse Rintakumpu
+358 40 196 8642
rintakumpu@gmail.com