Machine Learning 02 : From Paper to Product

SLQ Wiki Fabrication Lab 2024/11/04 00:50

Machine Learning 02 : From Paper to Product

With State Library closed to the public due to COVID-19, for this Machine Learning (ML) workshop we won't be able to use the Digital Media Lab at The Edge. Instead will be exploring examples of how a Machine Learning idea becomes a product that we can use, following projects from a research paper through to commercial software product.

In the second half of the workshop, we will take a look at Google Colab - a free ML toolkit that we can test out online. The workshop is not an introduction to coding or math, but we will give a of general overview of how ML is defined, how it is developed into a product and where it is commonly used today.

Requirements

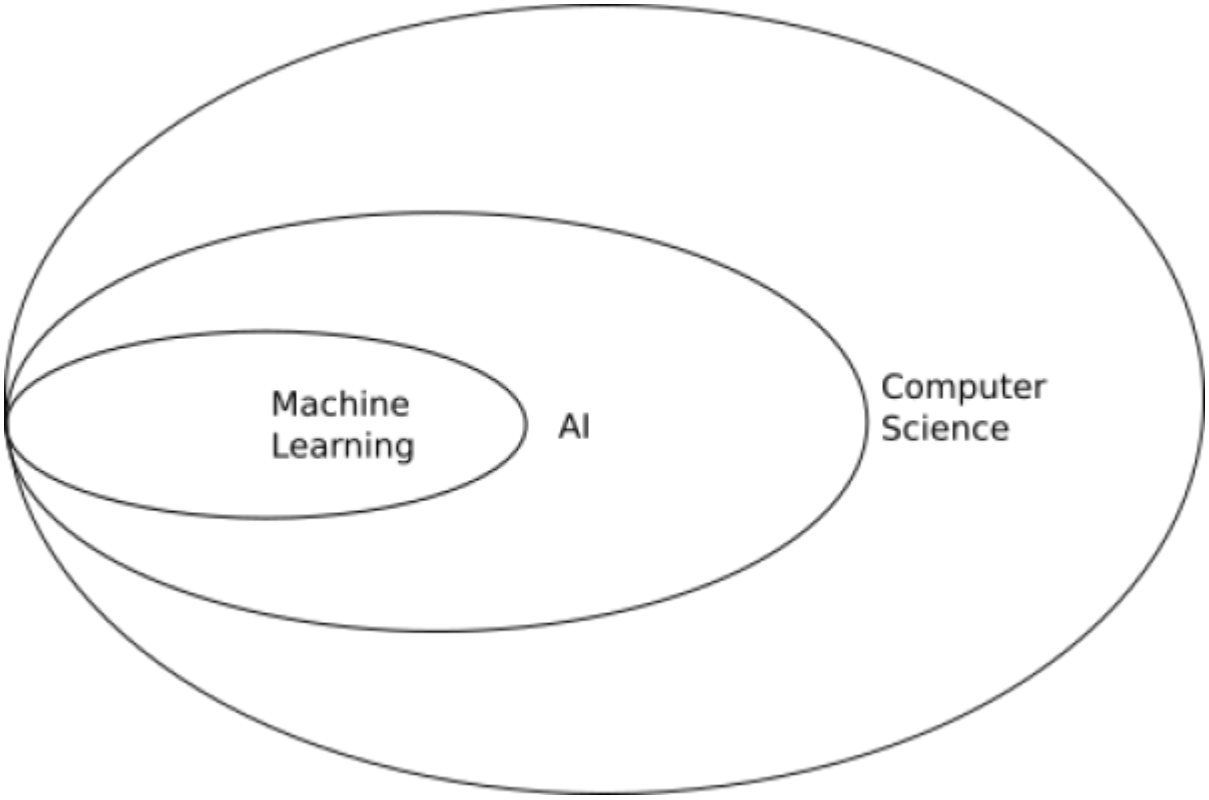
All we need to get started for this workshop is a Google account to access Google Colab in the second half of the workshop. If you don't have one you can quickly sign up. If you don't want to create a Google account, you can always just follow along with the examples.

Outcomes

- A basic Machine Learning background
- Short history of speech synthesis
- ML speech synthesis examples:
    - Commerical
    - Open Sourc research
- using Google Colab
- using Spleeter (audio source separation)

Background

Machine Learning(ML) is a subset of Artificial Intelligence (AI) which is is a fast moving field of computer science (CS). A good way to think about how these fields overlap is with a diagram.



Machine Learning - Why Now?

While many of the concepts are decades old, and the mathematical underpinnings have been around for centuries, the explosion in use and development of ML learning has been enabled by the creation and commercialisation of massively parallel processors. This specialised computer hardware most commonly found in Graphics Processing Units (GPUs) inside desktop and laptop computers and takes care of the display of 2D and 3D graphics. The same processing architecture that accelerates the rendering of 3D models onscreen is ideally suited to solve ML problems, resulting in specialised programming platforms, Application Programming Interfaces (APIs) and programming libraries for AI and ML.

Common Machine Learning Uses

One way to think of ML is as a *recommendation system*.

Based on input data (a lot of input data[1)])

- a machine learning system is trained
- a model is generated
- the model is used can make recommendations (is implimented) on *new* data.

:workshops:prototypes:machine_learning:ideepcolor:8725096366_d1fe677cc5_o.jpg

One extremely common application of this is image recognition.

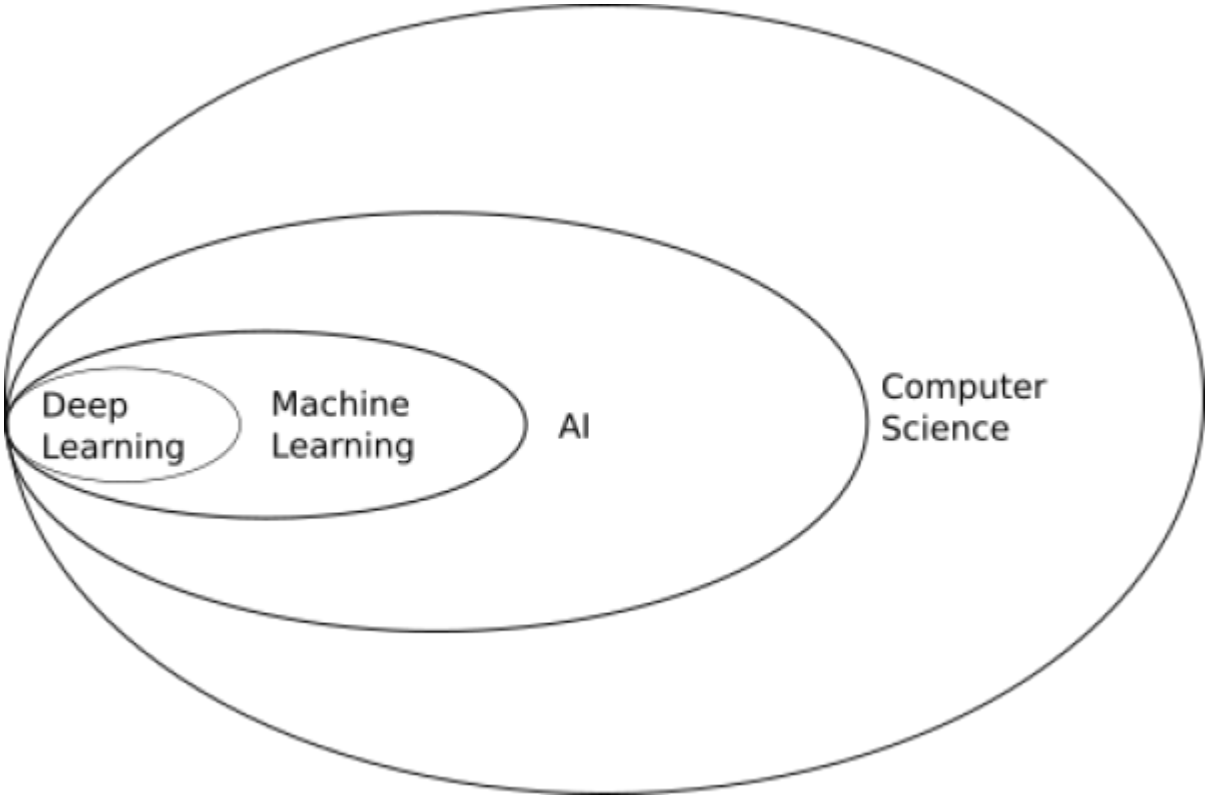
:workshops:prototypes:machine_learning:ideepcolor:2020-02-21_14_19_57-8725096366_d1fe677cc5_o_fb.jpg

When facebook asks you to tag a photo with names, you are providing them with a nicely annotated data set for **supervised learning**. They can then use this data set to train a model than then recognises (makes a recommendation) about other photos with you or your friend in it.

Snapchat filters use image recognition to make a map of your features, then applies masks and transformations in real-time.

Deep Learning

Today we are going to go a little “deeper” inside ML, exploring deep learning. Deep learning used multiple layers of algorithms, in an artificial neural network, inspired by the way the human neural networks inside all of us.



ML - From Paper to Product

We'll be exploring a few ML ideas, but to start with lets follow "Real-Time User-Guided Image Colorization with Learned Deep Priors", by Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, Alexei A. Efros from paper to product. Its not a recent paper in ML terms where it seems every month brings another breakthrough, but we can follow this paper right through to its release in Adobe Photoshop Elements 2020.

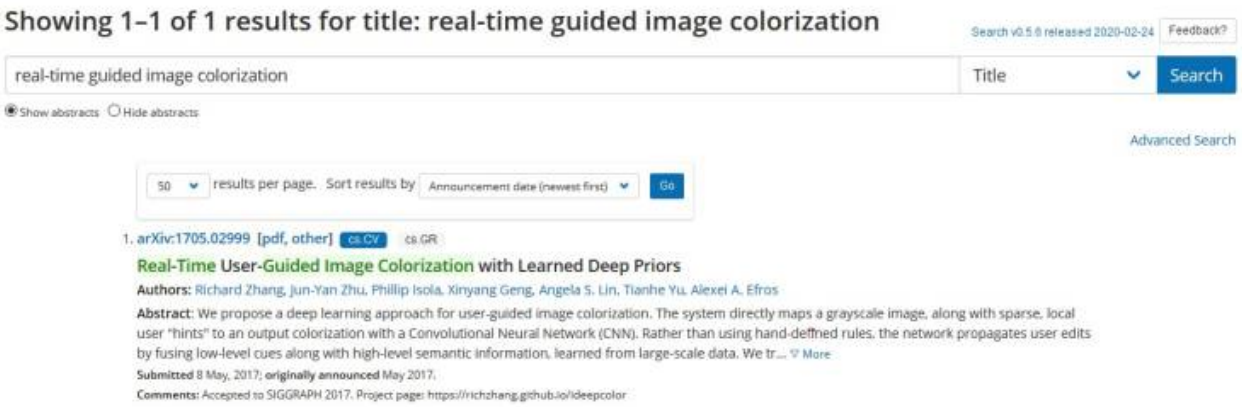
Research Papers on arXiv.org

arXiv.org is probably the worlds biggest and fastest growing collection of preprint electronic scientific papers from mathematics, physics, astronomy, electrical engineering, computer science, quantitative biology, statistics, mathematical finance and economics. All of the ML ideas we will be looking are either first published on arXiv.org, or reference papers on the site.

Finding a Paper

Papers on arXiv.org are moderated but not peer-reviewed, which means the speed and volume of publishing on this open-access repository is overwhelming. But to get started, lets say we are interested in re-colourisation of black and white images of our grandparents at their wedding, but we don't want to do all the work ourselves. We'd also like to interact in real-time and guide the process, so we can get the colour of grandad's suit and grandma's bouquet just right.

So lets search for “real-time guided image colourisation” but we'll use the American English spelling “ colorization”. Searching for “real-time guided image colorization” brings up our paper straight away, with handy PDF link[2].



Examining the Abstract

All research papers begin with an abstract, and a well written abstract will tell us all we need to know about whether the paper is relevant for you, particularly if we are looking for working demonstration. This time we are in luck - there is a link to a an ideepcolor demo site at the end of the abstract.

Check out the Demo

The demo for ideepcolor looks great and we've got a link at the top of the page, where ideepcolor is implemented on github.

Code Implementation on github.com

Github.com is a website used by software developers to create, collaborate and share source code, and is most likely the largest repository of source code in the world. Github is named after git, a free and open-source(FOSS) distributed version-control system for tracking changes in source code during software development. Git means that developers from all over the world can work on the same code and if the project is open source, build on, expand and re-purpose shared codes[3]. But lets back up a bit and cover off on what source code is.

Using the Source

source code is the instructions for a computer program contained in a simple text document.

For a computer to run a program, the source code either has to be

***compiled** into binary machine code by a compiler:

- his file is executable - in this case execute just means can be read, understood and acted on by the computer, or
- **interpreted** by another program, which directly executes the code

Here is a example of source code. in this case its a simple program in the C programming language that shows on the screen “Hello, World”

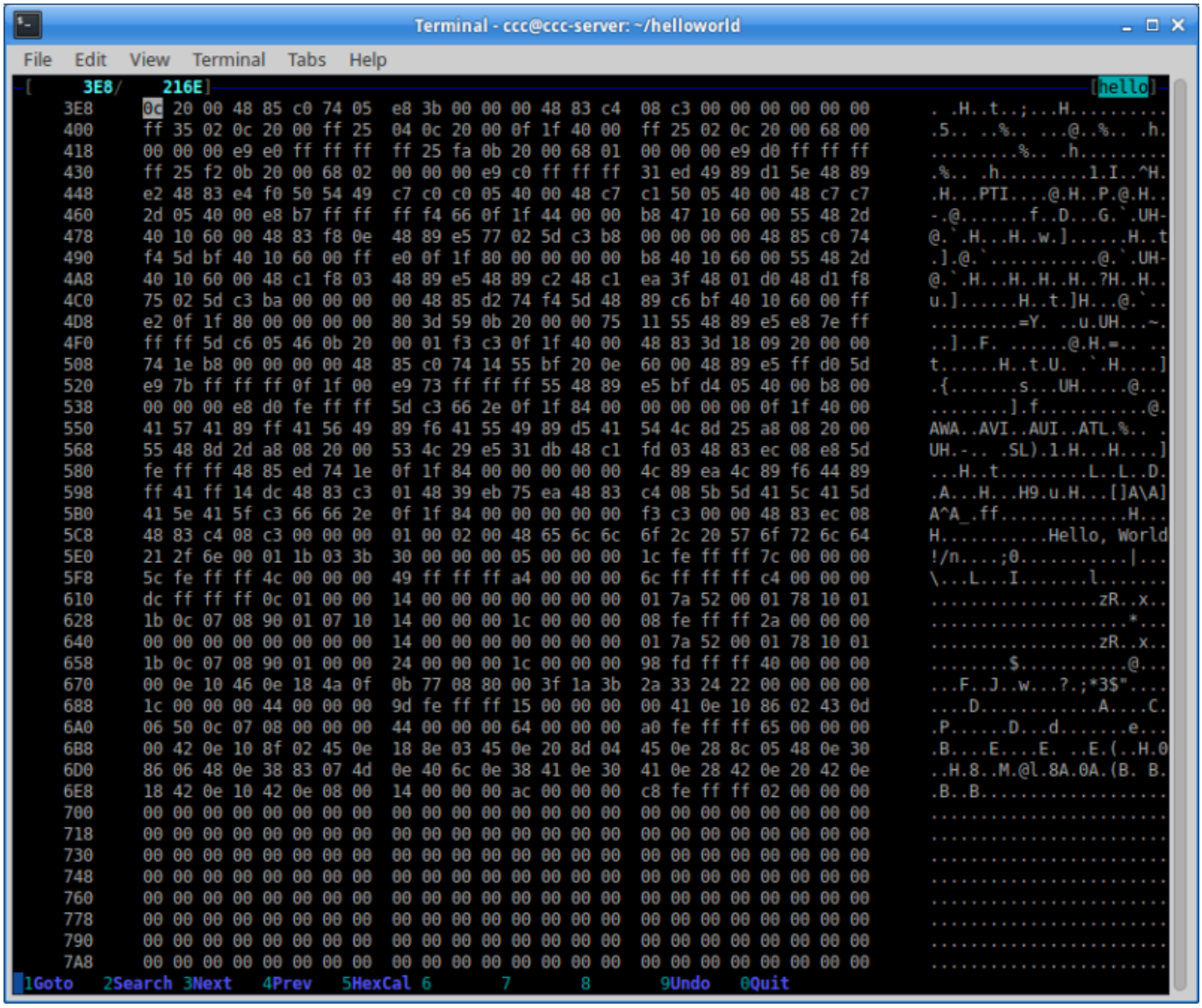
```
 #include <stdio.h>

 int main(void)
{
 printf("Hello, world!\n");
 return 0;
}
```

Despite the strange symbols, if you know how the C language is written, this program is **human readable**.

Once this code is run through a compiler, we get a binary executable file - which is **machine readable**.

But with the right tools (like a HEX editor) we can still open the file and edit it.

Here is the binary for our “Hello World!” program.



Open Source

Dokuwiki (the software we are using for this wiki) is open source, and developed publicly, and freely available on the internet. Anyone is able to grab the source code and run it, modify it or redistribute it.

Below is and example of the **open source** code for this wiki, which is written in a language called php.

```
  // define all DokuWiki globals here (needed within test requests but also helps to keep track)
  global $ACT, $INPUT, $QUERY, $ID, $REV, $DATE_AT, $IDX,
  $DATE, $RANGE, $HIGH, $TEXT, $PRE, $SUF, $SUM, $INFO, $JSINFO;
  if(isset($_SERVER['HTTP_X_DOKUWIKI_DO'])) {
  $ACT = trim(strtolower($_SERVER['HTTP_X_DOKUWIKI_DO']));
  } elseif(!empty($_REQUEST['idx'])) {
```

```
$ACT = 'index';
} elseif(isset($_REQUEST['do'])) {
$ACT = $_REQUEST['do'];
} else {
$ACT = 'show';
}
```

How did we get hold of the source code for this wiki? In this case all we did was look in the dokuwiki source found on github pick bit of code at random and throw it in our wiki.

So, finding the source for open software is easy. but to do the same thing with closed source program is usually difficult or impossible. Either you purchase or are given access to the code. Any other method may break all manner of licenses and laws.

ideepcolor on Github

For a project like ideepcolor, Github is where researchers and developers describe how they achieved their results with real, working code. There is generally an introduction, which should contain any major updates to the project, then we go through the prerequisites, setting up or getting started, installation, training and (hopefully) application. There are number of ways to demonstrate application of a model, ideepcolor has built a custom Graphical User Interface (GUI), which we demonstrated this project in our first ML workshop. More commonly demos are done with a jupyter notebook or Google Colab notebook. Now, lets look at the updates at the top of the ideepcolor repository - which tell us:

> 10/3/2019 Update: Our technology is also now available in Adobe Photoshop Elements 2020. See this blog and video for more details.

So it looks like this project has moved to the next stage - integrating ideepcolor into a commercial product.

Comercialisation:Standalone or Software as a Service

Many ML projects stay as research only, but for those that make it into commercial production, there is generally two paths:

- As a component of an existing software project as illustrated by ideepcolor.
- Offered as Software as a Service (SaaS), usually as as subscription

We can expect a commercialised ML product to be faster, more stable and more refined than a demo on github, with a price tag to match.

Other ML Examples

Given the process outlined above - lets explore a few more ML ideas. Instead of starting on arxiv.org - lets jump to searching github instead, so we know the projects we find have a good chance of a functioning demo.

Searching Github

- real-time voice clone
- Text To Speech (TTS)
- Audio Source Separation (spleeter)

1)
the ideepcolor training set is 1.3 million images
2)
this is obviously a contrived example of course, but the principle applies regardless
3)
if made available under an appropriate license