Springboard

Guidelines for Cloud Resources

AI/Machine Learning Career Track

April 2019

Summary of resources

As part of this course, students can choose among several free and paid cloud resources. We'll update this section (and the rest of this doc) as we continue to build our partnerships and provide more resources to our students.

Domino Data Lab (Student Tier)

Domino Data Lab is a popular Data Science and Machine Learning development platform that enables collaboration in Data Science teams. In this course, we use Domino for hosting our mini-projects, and some students also use Domino for their Capstone Projects. Students have indefinite free access to Domino's student tier.

Domino's student tier provides a containerized environment with several useful features:

- Python 2.7 or a Python 3.6 environment
- Common ML libraries such as scikit-learn, Tensorflow, Keras and Spark/SparkML
- Facilities to easily deploy an application with an API

Limitations

- Suitable only for data sets up to ~50GB. Beyond that, Domino encourages using their Data Sets feature which is not enabled in the student tier.
- Some other Domino features may not be available in the student tier
- Can't create customized environment (by installing custom libraries, for example)
- Only CPU, no GPU

Documentation for Domino

Google Colaboratory

Google's Colaboratory (aka Colab) tool provides a Jupyter environment for prototyping Machine Learning applications in Python. It provides some really awesome FREE facilities that should be enough for most Capstone Project:

- CPU + GPU
- 350GB of disk storage
- 13GB of RAM
- Interfaces directly with Google Drive and Github

- Install Python packages on the underlying system as you need

Limitations

- Colab is a prototyping tool, not a deployment one. It's good for research, not necessarily for deployment
- Can't save customized containers containing specific new libraries or tools
- No way to upgrade to a paid tier or better hardware configurations; what you see is what you get

Documentation for Google Colab.

Google Cloud (Free Tier)

Google Colab provides a free packaged environment with a single GPU, but if you want more flexibility to work with the plethora of tools that Google Cloud Platform (GCP) provides, you might consider using the GCP free tier. The free tier provides a mixture of the following:

- Tools that are always free (with limitations on amount of resources)
- Tools with a 12-month free trial
- $300 of GCP credits

Limitations

- Obviously, there are limitations on how much you can use the tools, and it's up to you to ensure that you don't exceed usage limitations
- You'll have to figure out how to set up each tool on your own, and what combination of tools you might need
- You have to put in your credit card information and keep track of any charges that you might eventually incur.

It's not practical for us to document all of the available tools in this document, but you can find detailed documentation here about the various resources and constraints.

AWS (Free Tier)

AWS is the original cloud platform, and similar to GCP, they also provide a free tier. The free tier provides a mixture of the following:

- Tools that are always free (with limitations on amount of resources)
- Tools with a 12-month free trial

- Shorter-term free trials

Limitations

- Unlike GCP, AWS does not provide any starter credits, so you're constrained to the free tools
- There are limitations on how much you can use the tools, and it's up to you to ensure that you don't exceed usage limitations
- You'll have to figure out how to set up each tool on your own, and what combination of tools you might need
- You have to put in your credit card information and keep track of any charges that you might eventually incur.

It's not practical for us to document all of the available tools in this document, but you can find FAQs here about the various resources and how to use them.

AWS Educate

In order to compensate for the lack of free credits offered by AWS, Springboard has partnered with AWS via their AWS Educate program to **offer every student $50 of AWS credits**. You should receive an email from AWS Educate to claim your credits within two weeks from the date your course starts. Please contact your student advisor if you haven't received your AWS Educate email after two weeks of your course start date.

How to make the most of your resources

Why conserve resources?

Cloud resources can sometimes feel very abstract, given that so much of our life nowadays runs 'in the cloud'. We constantly use email, social media and storage that rely on massive amounts of data stored in the cloud and transported to and from our devices. It's easy to forget that all this has a cost.

As future Machine Learning Engineers, we will insist that you learn how to conserve cloud resources for three important reasons:

1. **Costs matter to employers:** Cloud resources can add up in cost really quickly. An MLE who can get a product working with fewer and efficiently used resources is much, much

more valuable than one who cannot. Most employers will not provide unlimited resources, so you have to learn to work under constraints and within budget.

2. **It'll help you stand out in the job search:** A candidate who can demonstrate a clear understanding of how cloud platforms work, how they price their offerings, and shows awareness of how to estimate resource consumption, will stand out in their job hunt as a more responsible and mature engineer.
3. **It's good for the world:** Cloud resources ultimately consume a huge amount of energy; chips work on electricity and data centers need to be cooled. It's good for the environment and the planet not to waste energy in general.

So what are some ways you can be smart about the resources you use? We provide a few suggestions.

1] Prototype and test locally first

It's easy to look at a large data set and think that you need to start in the cloud right away with a powerful CPU, one or more GPUs and a humongous amount of data storage. But do you really need that at an early stage? Think (and discuss with your mentor) about how you can make the most of your local machine. Laptops and desktops nowadays are enormously powerful, with multi-core CPUs, plenty or RAM and disk storage.

2] Use smaller instances before trying larger ones

Once you're in the cloud, try using less powerful CPUs/GPUs first. This will help you stay within the free tiers of various cloud providers. If you're using up credits on AWS or GCP, this will help you burn your credits more slowly.

3] ALWAYS shut down your unused instances

The biggest source of wastage in cloud usage are unnecessarily running instances of machines. ALWAYS remember to shut down an instance when you're not using it. If this is an instance you're using free credits or paying for, it could burn through your wallet. And even if it's a resource you're using for free, it's important to be a good citizen of the cloud and develop good engineering habits. Your future managers and colleagues will thank you for it.

Using cloud resources for your Capstone Project

There are a variety of tools provided by the cloud providers in the market, and new tools are added all the time. In this section, we make a few recommendations based on our

conversations with employers and hiring managers, taking into account the resource constraints that you'll have to work under.

Overall, here's the strategy that we strongly recommend for your Capstone Projects:

1. First, use your local machine as much as possible to prototype
2. Then, use all the free tier cloud resources you have access to
3. Then, use your available credits on GCP or AWS, only if you need to

You're welcome to combine these resources in any way that makes sense to you and your mentor. For example, it's perfectly acceptable to use free computation resources from Domino Data Lab combined with paid storage on AWS S3 buckets.

Unfortunately, once you run out of these options, Springboard will be unable to cover you any more. You're welcome to pay for using more cloud resources at that point.

Getting started

When you're first getting started with your projects, you'll be spending a lot of time collecting and cleaning your data, and then exploring it. At this stage, we recommend that you really conserve and prototype on fewer resources. Some resources we recommend at this stage:

Your local machine

Whether you're working on your Capstone Project or at a company, it's always advisable to prototype on your own machine first. Not only is it easier, but it can also be quicker to fix bugs and try several things on your machine first instead of wasting cloud resources on erroneous code.

Google Colab

This is a great choice when you have a large amount of data (> 50GB) and you know you definitely need a GPU. However, remember that Colab is a research and not a deployment tool, and you'll have to spend some effort moving your code to a different platform down the line. Also, Colab has a fixed hardware tier, so you won't be able to run it any faster than what it provides, even by paying.

- Run your Python scripts on Google Colab
- PySpark on Google Colab

Domino Data Lab

For projects that are relatively smaller in terms of data size (< 50GB) and are CPU-only (no GPU needed), Domino Data Lab is an excellent free option to start with. You'll already be using it for your mini-projects, so you'll be familiar with it. Domino has excellent options to manage Machine Learning code and deploy applications.

Building and scaling an ML/DL prototype

As you get further in your project, you'll probably need to start seriously experimenting with ML/DL algorithms and thinking about how to create a full-scale ML/DL system with your entire data set. **You can still continue to use the free tools** that we suggested in the previous section, which include your local machine, Domino Data Lab and Google Colab, but depending on the needs of your project, there are a couple of other tools that we recommend you could consider.

Google Cloud ML Engine

The Cloud ML Engine lets you train, deploy and monitor ML/DL models on GCP. It provides a lot of built-in facilities to host your trained model in the cloud and simplify a machine learning workflow.

Note that when you use the Cloud ML Engine, you'll need to spend your free GCP credits on training and prediction, so remember to use only what you need and shut down instances when you're done with them! This pricing guide will help you estimate your resource consumption.

- Introduction to Google Cloud ML
- Classification of Signature and Text images using CNN and Deploying the model on Google Cloud ML Engine

AWS SageMaker

Similar to Cloud ML Engine, AWS provides its own ML/DL model management and deployment tool called AWS SageMaker. As of the current time, Amazon provides 250 hours free of AWS SageMaker use on a non-GPU system. If you want to go beyond that, you'll need to spend your AWS credits.

- Build, Train, and Deploy a Machine Learning Model with AWS SageMaker
- Building fully custom machine learning models on AWS SageMaker: a practical guide

Deploying your application

Finally, once you've trained your model and scaled it, what's next? You have to deploy it to a production system. **Many of the tools we have covered above, including Domino, Google Cloud ML Engine and AWS SageMaker offer one-click deployment options** i.e. once your code runs

independently and your model has been trained, you can simply package up your application as a container and deploy it on the platform you're using. These platforms also make it easy to create APIs to interface with your application. **We recommend you go with one of these tools if you want to keep things simple.**

Going beyond managed ML platforms

Some of you might want to go beyond these tools for any of the following reasons:

1. You would like to understand cloud systems in some more depth; you might be excited by the challenge of setting up your own servers, packaging up your own containers and deploying them.
2. You want to understand how to really scale and optimize a large-scale ML/DL application on a larger or more customized cluster setup, and think about engineering issues such as load balancing, fault tolerance and availability.
3. You're using some highly customized tools or libraries that are not provided by any of the 'standard' tools mentioned earlier

So here are some additional deployment options you can look at:

EC2 and Docker on AWS

A simple deployment solution is to set up an EC2 server on AWS, install and configure everything you need on there, and then get your application running. There are lots of tutorials out there on how to set up EC2 machines.

- Setting Up AWS EC2 Instance for Beginners
- Deploying a Python Web App on AWS

A raw EC2-based deployment works, but it's tedious and error-prone because it will require you to manually set up and configure the system. It's also not very portable i.e. if you lose the machine, you'll have to do all the work to recreate the setup.

That's where containers come in. Containers let you create virtual setups containing all of the tools pre-packaged, along with your application. AWS provides various facilities for using containers, including AWS Elastic Container Service (ECS). AWS also provides containers ready for Deep Learning.

- What is Docker?
- Deploying a Deep Learning pipeline with Docker on AWS
- AWS Deep Learning containers

Containers on GCP

Google's Cloud ML Engine allows developers to use their own customized containers that bring their own tools and libraries. Google has been internally using containers for over a decade, so GCP incorporates containers very naturally.

- What are containers?
- Using containers on Cloud ML Engine

Conclusion

We've covered many different options for using cloud resources in your Capstone Project. Please work closely with your mentor to pick the option that's right for you. We'd like to leave you with three takeaways.

1] Keep your deployment architecture as simple as possible

Building and deploying an ML/DL application is hard. While it's tempting to do everything perfectly and get into the weeds so you can show off your skills, it's impossible to learn everything that's out there in such a short amount of time. Use the tools, services and platforms that will get you to your goals and your project done as quickly as possible.

2] Prototype first on smaller and cheaper resources

Yes, we know. You want to run your fancy Deep Learning algorithm on the Terabyte of data you've acquired, and use 16 GPUs and 256 GB of RAM to do it. But please prototype and experiment on your own machine, or on a smaller cloud system first. Use the free resources available before spending your valuable credits.

3] When you're not using a cloud instance, TURN.IT.OFF.