Data and ML model management with PyTorch and AWS

By
Sushil Ram Achamwad

Cloud technology started off with providing remote storage and access to enterprise data in late nineties and has continuously evolved since then. Feature of performing data analytics on enterprise data stored in cloud to gain business insights is one such example. This feature gave rise to production scale analytics where data analytics programs or products are put into action for business needs in real time using cloud infrastructure.

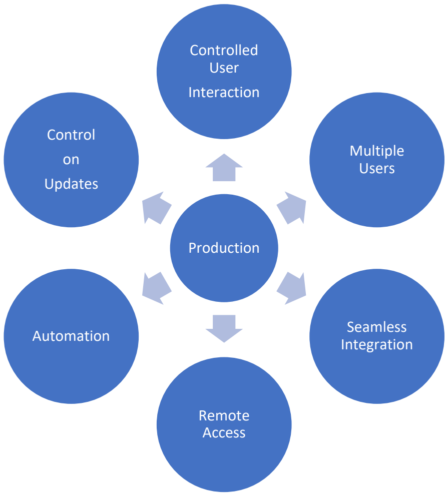


Fig 1

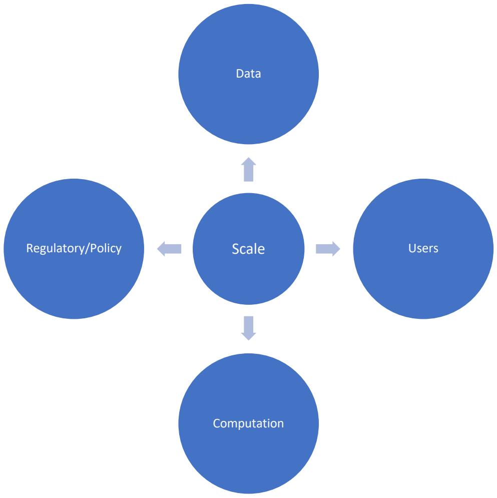
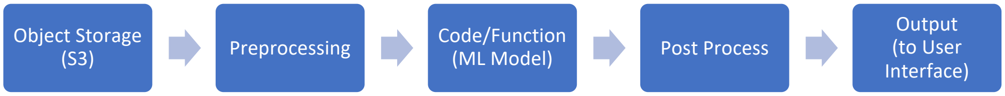


Fig 2

Production scale analytics leverages the features of cloud technology to perform data analytics over data stored in cloud with software's and hardware's stored in cloud infrastructure rather than using resources of local computers as in non-production scale analytics .

Components that make up "Production" and "Scale" part of the production scale analytics are depicted in above figures. As shown, production environment has enabled controlled user interactions which has scaled regulatory power of businesses significantly. With multiple users enabled to access and work on enterprise data it has also increased the user scale for businesses. They can satisfy almost unlimited storage requirements which provide scalability in terms of data and computational requirements of organizations. Facilitating remote access to hardware and software stored in cloud using internet, production scale environments have improved collaboration across teams at different locations across globe making collaboration much easy and efficient. It has enabled automating analytics processes which can be then tested, validated and controlled as they are updated as per business requirements.



Broad Level Analytics Pipeline

Often times machine learning is part of analytics pipelines. Machine Learning can be understood as an equivalent of a function which takes some inputs and performs specified logical operations on the input data with an objective of learning the inherent behavior, patterns and trends in the input data with respect to certain output measures. So in production scale analytics the input data is the data stored (ex. In S3), its pre processed for feeding to ML/AI code (in python, R etc.) Then the output is post processed and is fed to an output user interface. It can be hard to manage machine learning at production scale because the system should be able to handle huge volume and variety of data, should show low latency, should be scalable, should be flexible for changes or fine tuning and should be economical.

Production Scale Environments

Production scale environments demand storage independence, scalability, mobility and cost effectiveness. A local computer cannot fulfill all these requirements which brings organizations to utilize cloud infrastructure for production scale applications.

The cloud infrastructure saves significant costs for the organizations in the form of reduced licensing costs for multiple users, usage-based storage fees, flexible payment options for data maintenance etc. It provides almost unlimited storage capacity so that the organizations do not have to worry about running out of storage. Managing backups of enterprise data and recovering it in Cloud infrastructure is quite easier as compared to backing it up in physical devices such as hard-drives. This reduces the risks of losing essential business information also gives more control in managing the security of it. Businesses can expand or reduces their need of storage or computing resources in cloud as per their business requirements hence it’s scalable to user’s needs unlike local computers. Cloud environment can better manage fluctuating demands of computing, storage and maintenance resources and hence facilitates faster and efficient deployment of or running applications.

The extent of business applications of cloud or production scale environments have prompted many companies to provide cloud services to other business for a usage-based fee at the incentive of saving greater costs of buying and maintaining physical servers. Market share and annual market share growth is shown below for top 8, Market leaders in these services which are: Amazon, Microsoft, Google, IBM, Alibaba, Oracle, Salesforce and Rackspace.

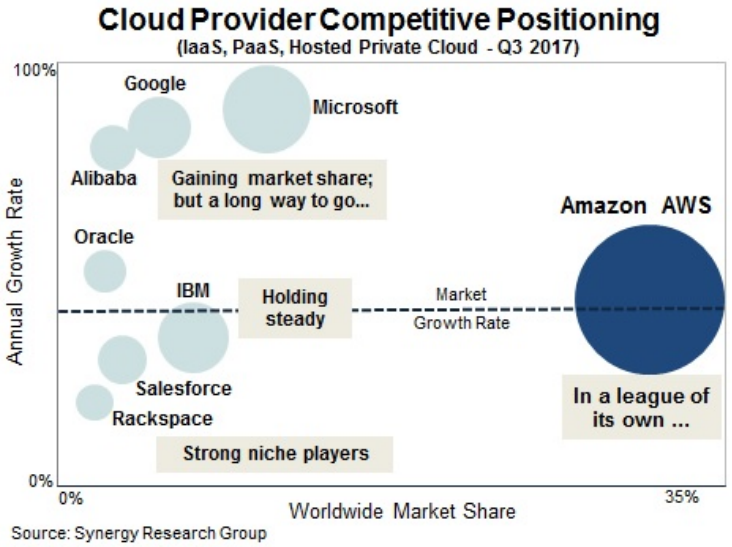


Fig. 3

With largest market share, amazon AWS has been a consistent leader in public cloud computing market. AWS can be used to Compute, Develop Tools, Store data, Manage Tools and Maintain Database. Further, it allows migration, networking and content delivery, media services, machine learning, analytics, customer engagement, business productivity, security, identity and compliance, mobile services, internet of things, application integration, AWS cost management and much more. It has the most comprehensive network of data centers worldwide and being one of the oldest service providers, it's more relied upon by organizations.

AWS provides variety of options catering to specific business needs like Elastic Compute Cloud (EC2), its flagship compute service provides resizable compute capacity in the cloud by offering different services such as assortment of instances, high performance computing, GPU instances and auto-scaling. AWS provides comprehensive container services that support Docker, Kubernetes, Fargate services. Its competitive pricing strategies such as providing free tier for EC2 for up to 12 months further makes it a lucrative choice for businesses.

Managed Services

One of the key USP's of Amazon AWS services is their managed service. In managed services, cloud service providers provide resources such as hardware and software tools, via multiple servers to maintain, automate or reduce cost of cloud infrastructure of customer businesses.

By automating common activities such as change requests, monitoring, security and backup services, AWS managed services reduce overhead operations costs and risks. They assess in implementing individual corporate and infrastructure policies of business for their AWS infrastructure.

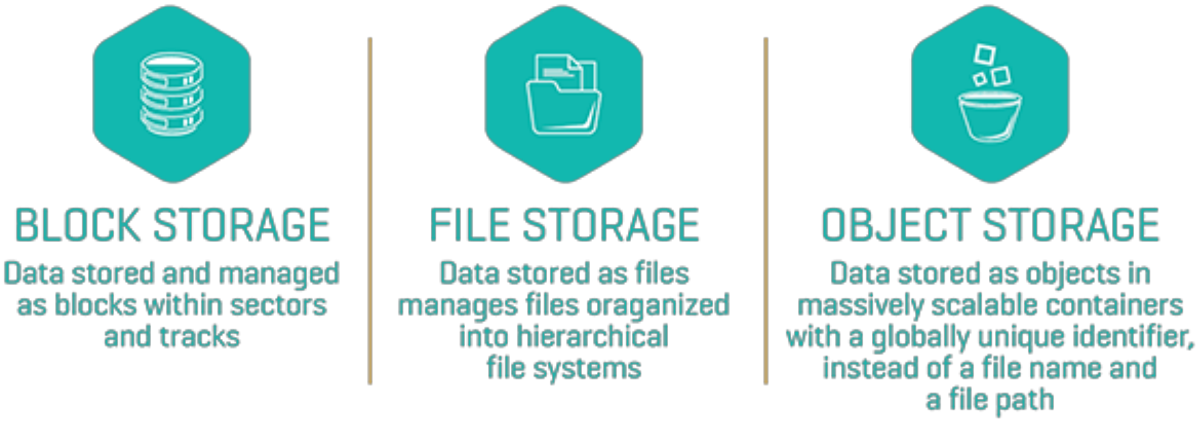
For example, being a user of Redshift, AWS will provide Massive Parallel Processing (MPP) capabilities of its data warehouse architecture for incredibly fast and scalable data warehousing applications.

This is quite different for Non-Managed services, for example, for a user developing a database system, each individual instance is required to spin up. So, it would involve first spinning up EC2 instances on the computer by user, then attaching EBS for storage and finally logging in to download or install software to get database service.

In general, Managed services are beneficial for reducing operations costs, consistent availability of services because of multiple servers, automated resource balancing, enhanced network security and protective environment through protected firewalls and intrusion detection system/intrusion prevention systems.

Some of the key benefits of AWS managed services include having low maintenance requirements, low setup time, greater stability of services and most importantly the responsibility of managed services lies on AWS. So, those businesses who use managed services can better focus on their business operations rather than investing time in server glitches and breakdowns.

Storage Resources


BLOCK STORAGE
Data stored and managed as blocks within sectors and tracks
FILE STORAGE
Data stored as files manages files oraganized into hierarchical file systems
OBJECT STORAGE
Data stored as objects in massively scalable containers with a globally unique identifier, instead of a file name and a file path

In general, three types of data storages are required to run data analytics at production scale viz. block storage, file storage and object storage.

In Block storage, a block is a raw storage volume filled with files that have been split into chunks of data of equal size. A server-based operating system manages these volumes and can use them as individual hard drives. Block storage is common in databases and other mission-critical applications that demand consistently high performance.

In File storage, files are stored in traditional way in which first we give files a name, tag them with metadata, then organize them in folders under directories and sub-directories. The standard naming convention makes them easy enough to organize. Many companies demand a centralized, easily accessible way to store files and folders. File level storage can deliver these perks at a cost that is typically affordable on a small business budget.

In object storage, object-based storage stores data in isolated containers known as objects. You can give a single object a unique identifier and store it in a flat memory model. The data could be physically stored on a local server, or a remote server. Object storage combines the pieces of data that make up a file, adds all its relevant metadata to that file, and attaches a custom identifier. Object storage is driven by metadata, and with this level of classification for every piece of data, the opportunity for analysis is far greater.

Amazon Storage Solutions

Amazon AWS's multiple storage solutions is another flagship service that enables AWS to continue being the leader in the market. These solutions enable businesses to choose storage service as per their business needs.

Amazon S3:

Amazon S3 is an object storage. S3 can be used to store and retrieve practically any amount of data from anywhere i.e. websites, mobile apps, data from IOT sensors and devices. Scalability is the USP of amazon S3 which makes it suitable for businesses having fluctuating storage needs. With Amazon S3, there is no need to move the stored data into separate analytics system for performing big data analytics. S3 makes the stored files available at 99.99% (almost all) the time and its small- mid size businesses can afford these services because of its competitive and affordable prices.

Amazon Glacier:

Amazon Glacier is one of the most affordable object storage service from AWS. But Glacier is designed for long term backup and archive of any data. So it's more useful for those users who need to archive greater volumes of data which they don't require to access regularly.

Amazon EFS:

Amazons Elastic File System stores and shares data in simple and scalable file system which is available for use with EC2 instance in AWS cloud. It was the first product to integrate with EC2 instances. EFS provides better administration by making it possible to grant certain file permissions across their storage. Although EFS is scalable to business demands its costs are a little on higher side. EFS are better for businesses having greater demand of file storage or organizations that work in great volumes with files and applications.

Amazon EBS:

Amazon Elastic Block Store(EBS) stores and process block data on persistent volumes for Amazon EC2 instances. So, EBS is designed to be used in conjunction with Amazon EC2. Here the data can be stored only when the two are connected. This storage solution is better for businesses having steady storage needs such as for applications like data analytics softwares and SQL databases but the scalability is limited.

AWS Snowball:

AWS Snowball physically migrates petabyte scale data sets into and out of AWS. Snowball makes frequent transfer of Data heavy IT environments much easier without incurring huge network usage fees. Snowball can perform the transfer operations even without fast internet connection so its ideal for locations having limited high speed internet.

Amazon RDS:

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It is easy to administer as it makes it easy to go from project conception to deployment. It doesn't require any infrastructure provisioning and also doesn't require installing and maintaining any database software. Using RDS, database's storage and compute resources can be easily scaled up or down as per business requirements. Further the user only pays for resources consumed by the user hence is quite inexpensive. Having options to choose between two SSD backed storage options and the other for cost-effective general-purpose use, Amazon RDS is fast and secure as it makes it easy to control network access to your database as well.

Compute Resources

Compute resource are the most importance component of any cloud infrastructure. Compute resources are analogous to CPU of local computer or laptop. Variety of compute solutions offered by service providers makes it easy to obtain exactly what the business requirements are. Within a cloud environment, compute nodes form a core of resources which supply the processing, memory, network, and storage that virtual machine instances need. When an instance is created, it is matched to a compute node with the available resources. A compute node can host multiple instances until all of its resources are consumed.

There are four major compute resources in Amazon AWS viz. Amazon EC2. Amazon LightSail, Amazon EC5 and AWS Lambda.

Amazon EC2:

Whenever the user wants to run any application, control and manage server or cluster level functions such as scaling and deployment, Amazon EC2 is used. EC2 offers a wide selection of instance configurations optimized for every use case.The payment option is flexible and is follows pay-per-use model.

Amazon Lightsail:

Whenever the user wants to run simple applications and websites on one or a few servers for a low, predictable price Amazon Lightsail could be used. Here, the user controls the server, OS, and other software through the Lightsail console but has less control and fewer options available than EC2. In Lightsail, users are billed a flat predictable rate for your plan each month.

Amazon EC5:

Whenever the user wants to run stateless or stateful applications packaged as Docker containers, EC5 can be used. Here the user decided the provision and scale of the server capacity and manages utilization and availability. Here, AWS manages the cluster state and container deployment.

AWS Lambda:

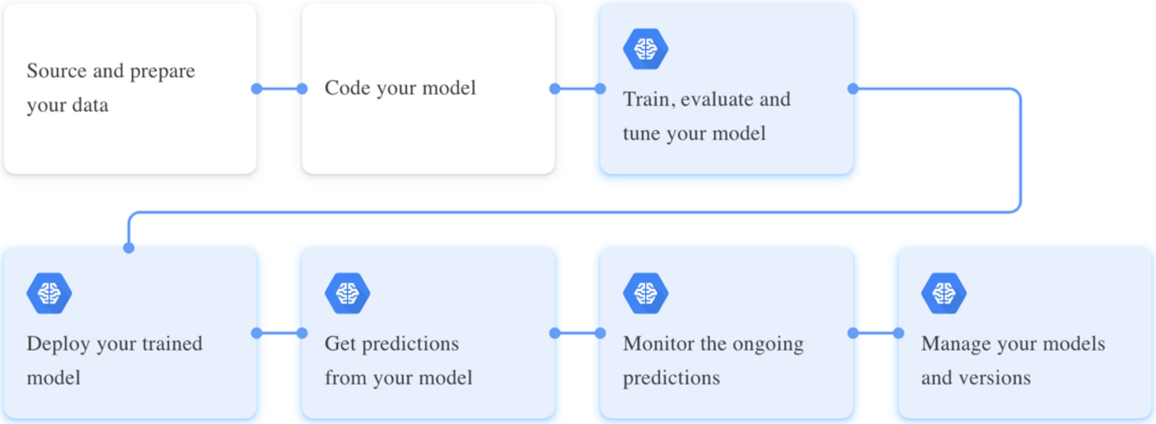
Whenever the user wants to run event-initiated, stateless applications that need quick response times, AWS Lambda can be used. Here, AWS decides provision and scale of the server capacity and manages utilization and also it manages the availability and fault tolerance of the application.

AWS Lambda does not provide any visibility into the server infrastructure environment used to run the application code, while Amazon ECS actively exposes the servers used in the cluster as standard Amazon EC2 instances and allows (or more correctly requires) the user to size and scale their fleet themselves

AWS Lambda functions must be written in one of a handful of supported languages and are restricted in the type of actions they can perform. Amazon ECS, on the other hand, can run any container using any code that is capable of running in a container (which is almost any application that runs on a typical Linux operating system).

Amazon SageMaker is a fully-managed service that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale. Amazon SageMaker removes all the barriers that typically slow down developers who want to use machine learning.

Running Machine Learning at Production Scale


Source and prepare your data
Code your model
Train, evaluate and tune your model
Deploy your trained model
Get predictions from your model
Monitor the ongoing predictions
Manage your models and versions

ML workflow

Usually Machine learning work flow is comprised of steps specified in above figure. But before jumping to source and prepare data user should formulate a well-defined problem to solve, then s/he should judge if Machine Learning is the best solution for the problem and decide upon success measurement parameters for the ML model.

Once these steps are over then one should start to source and prepare a training data which may include combining data from multiple sources, perform exploratory analysis through visualizations. Then while coding the model, the model should be developed using established analytics techniques. Next the developed model should be trained evaluated and tuned. In training, the outcome of the target variable is already known so different analytics techniques are tried to best fit the data and predict target outcome more accurately. Similarly, in evaluation the predicted outcome should be compared with actual outcome and then the accuracy of model should be judged. The model can be tuned for controlling the training process. Now the model should be tested in an environment as close as possible to final application.

Next step is hosting the model in the cloud which involves serializing the trained model. Then obtain predictions from the model, followed by monitoring them and as required different versions of the model should be managed.

PyTorch

PyTorch is a Python-based scientific computing package that uses the power of graphics processing units. It is also one of the preferred deep learning research platforms built to provide maximum flexibility and speed. It is known for providing two of the most high-level features; namely, tensor computations with strong GPU acceleration support and building deep neural networks on a tape-based autograd systems. Pytorch supports computational graphs and its framework works on cloud as it provides front end API for building applications with framework.

After training a machine learning model we serialize the model. Serialization is basically the process storing data structures or objects in a file or memory such that it can be then reconstructed later in any computer environment. Meaning a serialized model in onnx format could be supported by many programming languages.

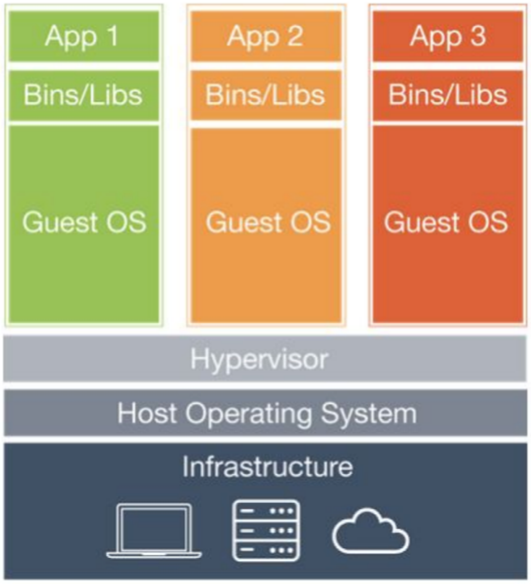
As far as using Pytorch on AWS is concerned, user can easily train and deploy your PyTorch deep learning models in Amazon SageMaker. This is the fourth deep learning framework that Amazon SageMaker has added support for, in addition to TensorFlow, Apache MXNet, and Chainer. Just like with those frameworks, now users can write PyTorch script like and rely on Amazon SageMaker training to handle setting up the distributed training cluster, transferring data, and even hyperparameter tuning. On the inference side, Amazon SageMaker provides a managed, highly available, online endpoint that can be automatically scaled up as needed.

Using PyTorch in Amazon SageMaker is as easy as using the other pre-built deep learning containers. Just provide your training or hosting script, which consists of standard PyTorch wrapped in a few helper functions, and then use the PyTorch estimator from the Amazon SageMaker Python SDK

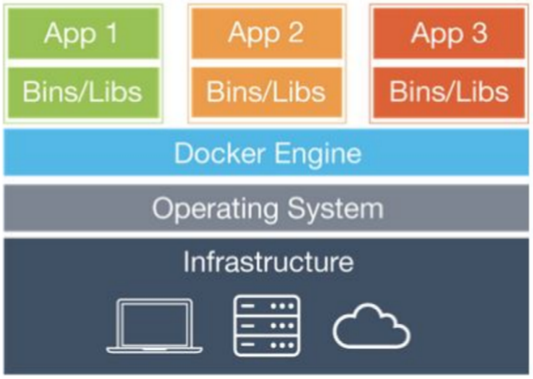
Another way to use Pytorch on AWS is with AWS Deep Learning AMI. The AWS Deep Learning AMI, lets user spin up a complete deep learning environment on AWS in a single click, with PyTorch, Keras 1.2 and 2.0 support.

Deploying PyTorch and other Processing Stages with Docker

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. By doing so, the developer can rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.



Virtual Machine



Docker System

In a way, Docker is a bit like a virtual machine. But unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of the application.

Docker allows multiple instances of the running model to share the common resources (i.e. OS) instead of creating multiple copies of the common required resource. This reduces the quantity of resources required to run the same number of instances.

Docker makes management and deployment easier by building containers. Containers allows developer to package application with all their required parts such as libraries and other dependencies. By making container it becomes easier and more dependable to run the package on the Linux operating system even though the machine may have different customization and setting from the machine that was used for writing and testing the code. Since package is installed at once the amount of time required for installing the pre-requisites is drastically reduced

AWS services such as AWS Fargate, Amazon ECS, Amazon EKS, and AWS Batch make it easy to run and manage Docker containers at scale. AWS also supports Kubernetes which is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery. Kubernetes will allow us to run required containers for multiple instances whenever an API request is sent to the API.

Now, after we serialize a model next step is to send the model to an API that can read and deploy the model for production. The options for doing so are to use TensorFlow, Torch, PyTorch or MxNet.

For using MxNet there are two advantages. First, it provides optimized numerical computation for GPUs and distributed system while allowing the user to build models in high level languages

like python and R. Second is that MxNet automates common workflow through which standard neural network can be expressed concisely in few lines of code. The disadvantage is that MxNet has a small community and very limited ready-to-use documentation.
TensorFlow has three advantages i.e. , first it is ever evolving and new features and functionalities keep getting added to it, second it comes with Tensorboard which provides easier Network Visualization. It has disadvantage that it is too complex to use without any interface and has been criticized on some communities for being slow.

Conclusions

Our discussion concludes by understanding: what are all the components of production scale analytics, how Amazon is dominating the public cloud service market, what all services are provided by AWS for cloud storage, computing and analytics, how Pytorch and Tensorflow are more suitable for production scale applications, how docker containers are part of the production scale system and finally how all these services are served under one umbrella of Amazon AWS .

In overall, AWS provides end to end solution for production scale analytics, however concerns of security of cloud infrastructure still exists and it is to see how these technologies shape up in near and long term.

Reference:

1. https://www.cbtnuggets.com/blog/2017/12/5-awesome-aws-storage-types/
2. https://aws.amazon.com
3. https://www.tensorflow.org/serving/serving_basic
4. https://cloudian.com/blog/object-storage-vs-file-storage/
5. https://searchstorage.techtarget.com/definition/object-storage
6. https://docs.aws.amazon.com/dlami/latest/devguide/tutorial-pytorch.html
7. https://aws.amazon.com/pytorch/
8. http://kevinzakka.github.io/2017/08/13/aws-pytorch/
9. https://medium.com/@julsimon/training-with-pytorch-on-amazon-sagemaker-58fca8c69987
10. https://kubernetes.io/
11. https://en.wikipedia.org/wiki/Serialization