**PyTorch Consulting**

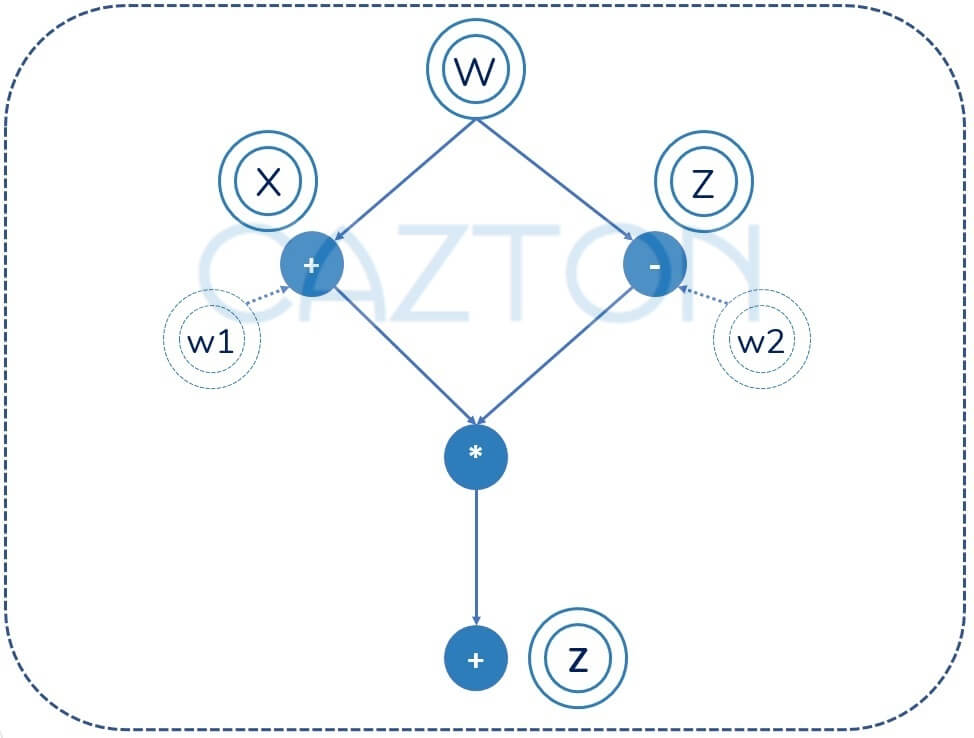
* **Global AI Keynotes: CEO of Cazton** will be delivering keynotes for Global AI Developer Days in Fall 2022. The keynote will compare the AI offerings of AWS, Azure and GCP. The demo will include multiple programming languages, including Node.js, Python and the latest .NET (C#) framework.
* PyTorch is an open source machine learning framework created by Facebook.
* **Stable Diffusion**, a text-to-image model, that helps create stunning art within seconds based on plain text prompts, has been created using PyTorchShape.
* At Cazton, we help Fortune 500, large, mid-size and startup companies with artificial intelligence and **PyTorch development, deployment, consulting, recruiting services and hands-on training services.**

**Introduction**

PyTorch is one of the most popular deep learning frameworks and is based on the library Torch. Computer vision and natural language processing are two major use cases. When it was launched, it was easier to use with Graphics Processing Units (GPUs) and that explains why researchers preferred it over competitive offerings like TensorFlow. API calls in PyTorch are executed when they are called. This is also known as eager execution. TensorFlow originally added the calls to a graph for later execution. This is known as lazy or deferred execution. However, lately it has made its support for eager execution a lot better.

PyTorch allows easy switching between eager and graph mode. This can be done using TorchScript. It’s production ready, easy to scale, and enables distributed training and performance optimization. The community has built a rich ecosystem of machine learning libraries on it. The community is so active that right after Open AI launched its services, Stable Diffusion was created in no-time and built on top of PyTorch using Generative Pre-trained Transformer 3 (or GPT-3). GPT-3 is an autoregressive language model that uses deep learning to produce human-like text.

Tensors and graphs are the core components of PyTorch. Tensors are like a multidimensional array that can run on GPUs to accelerate computing and are used to store and manipulate the inputs, outputs and parameters of a model. Graphs are data structures consisting of nodes and edges. PyTorch records all the tensors and executed operations in a directed acyclic graph (or DAG).



It’s based on dynamic computation, which means the graph is built and rebuilt at runtime. The code that performs the computation for forward pass also creates the data structure needed for back propagation. It was the first deep learning framework that was based on this principle but also matched the performance as well as capabilities of static graph frameworks. This is why it’s a great fit for a wide array of deep learning paradigms including convolutional networks and recurrent neural networks.

**Major use cases**

PyTorch is popular among researchers (data scientists) as well as software developers. It’s an easy-to-use API, easy to debug and comes with a wide range of built-in Python extensions. It's compatible with scikit-learn and Captum (for model operability) and has GPU support. GPUs are composed of hundreds of cores that handle thousands of threads simultaneously.

PyTorch is mainly used for Natural Language Processing (NLP) and computer vision. Models that treat language as a flat sequence of words use recurrent neural network model. However, the consensus is shifting towards language being a hierarchical tree of phrases. Recursive neural networks are deep learning models that can make this structure work even though traditionally they have been hard to implement and inefficient to run. However, PyTorch makes complex NLP models a lot easier to create and more efficient to run.

PyTorch is used for computer vision, including image classification, object detection and segmentation. Images consist of pixels. This collection of pixels can be represented using tensors, which makes PyTorch a great fit for computer vision related problems. In computer vision, we can start with a simple MNIST dataset consisting of handwritten digits. This could be represented with 28x28 pixels. This problem can easily be solved by a NumPy array. However, for more complex data sets it makes sense to use tensors. PyTorch tensors are better than NumPy arrays because tensors support parallel operations on GPU. For example, if we were to add the three color code (Red, Green and Blue), we would need an array of 3xWxH and, if this is a video of 100 frames, we may use the tensor of size 100x3x28x28.

**Performance**

Python is one of the most popular programming languages. However, performance isn’t its best suit and programmers find it to be quite slow. PyTorch uses Python’s strength which is ease of use. It also takes advantage of the high performance benefits of low-level languages like C and C++. All the heavy lifting in PyTorch is implemented in C/C++. This is made possible by using C/C++ as an extension to Python.

Why top Fortune 500 companies use Cazton for TensorFlow

[Know More](https://cazton.com/consulting/artificial-intelligence/tensorflow)

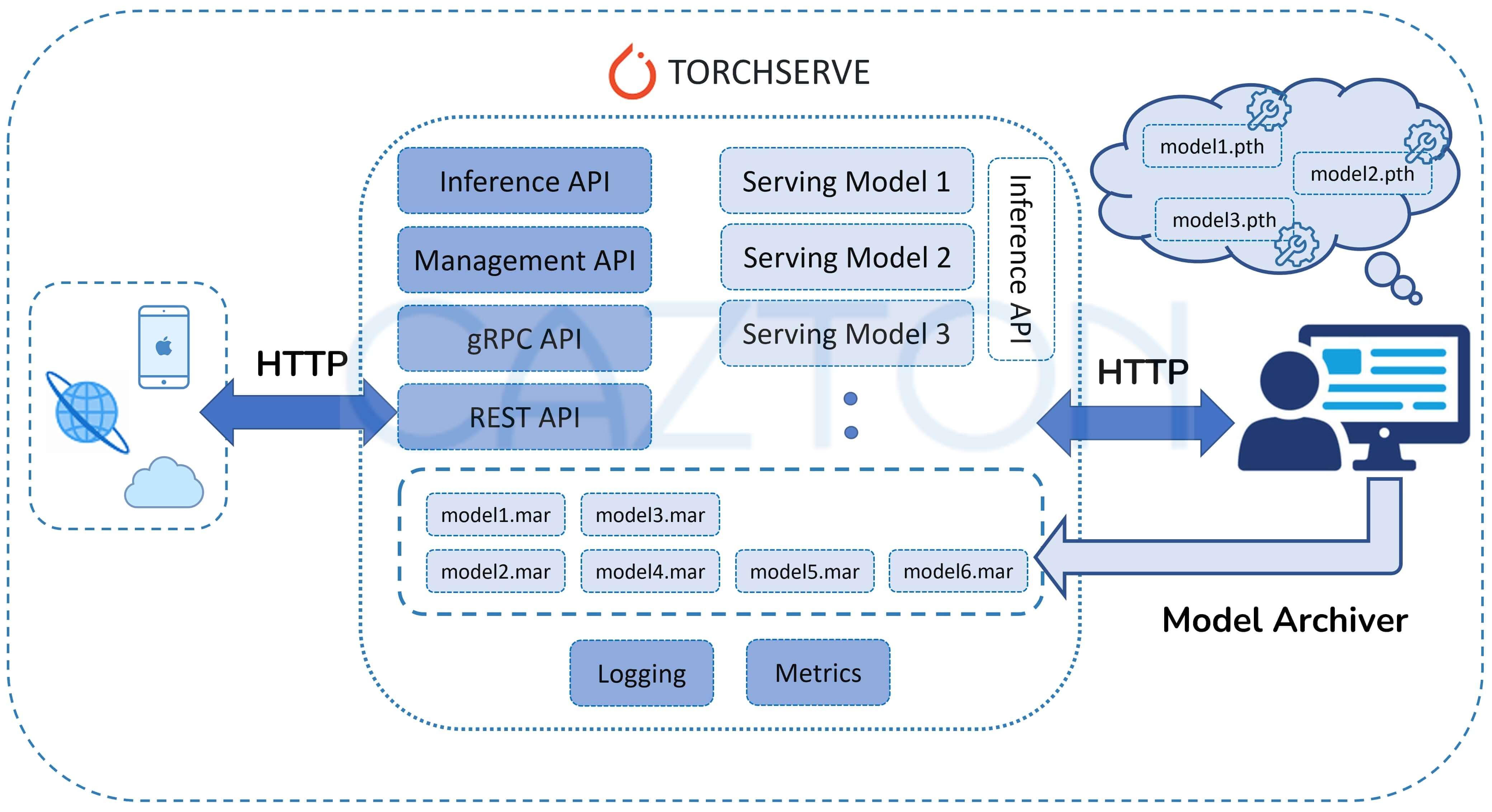
One of the most common use cases is working with NumPy, which is an extremely popular library and offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. While PyTorch enables conversion from NumPy to its own internal representation (a PyTorch tensor), and vice versa, it doesn’t copy the data over. It only uses a pointer to the raw data. Effectively, it’s just a reference that allow the memory to be shared.

**TorchServe**

With TorchServe, you can deploy PyTorch models in either eager or graph mode using TorchScript, serve multiple models simultaneously, version production models for A/B testing, load and unload models dynamically, and monitor detailed logs and customizable metrics.

As mentioned above, Python’s main strength is ease of use. However, in production environment, speed and efficiency are critically important, so it makes sense to use a standalone C++ program in production instead of Python. This can be done using TorchScript. Its main purpose is to create serializable and optimizable models from PyTorch code. The best part is that any TorchScript program can be saved from a Python process and loaded in a process without any Python dependency.

This means that the TorchScript program can be run independently as a standalone C++ program. In short, you are transitioning the Python model to a C++ program with no Python dependency. This enables researchers and developers have the best of the both worlds: speed up development using Python with it diverse ecosystem and high performance output in production using C++. This could be done in a few easy steps: convert the PyTorch model to TorchScript via tracing or annotation, serialize the script module and then load and execute script module in C++.



**PyTorch in Cloud**

PyTorch is supported on all major cloud platforms. It provides frictionless development, GPU training and ability to run models in a highly scalable cloud environment through prebuilt images. Major cloud support for PyTorch has been discussed below:

* **Microsoft Azure:** Microsoft fully supports PyTorch on Azure. It’s also a huge contributor to PyTorch community with contributions in PyTorch profiler, ONNX runtime on PyTorch, PyTorch on Windows and has joined PyTorch foundation. It supports the following PyTorch related offerings on Azure: PyTorch on Azure, Azure Machine Learning and Azure functions.
* **AWS:** Amazon fully supports PyTorch on AWS. It has contributed towards TorchServe that supports ML environments including Amazon SageMaker, Kubernetes, Amazon Elastic Kubernetes Service (or EKS) and Amazon Elastic Compute Cloud (EC2). The AWS Neuron SDK is integration with PyTorch and Amazon S3 plugin for PyTorch is open source and intended to be used from PyTorch to stream data from S3.
* **GCP:** Google supports a PyTorch instance which is a VM instance. GCP Deep Learning Containers are Docker images that use popular frameworks like PyTorch and are performance optimized, compatibility tested, and ready to deploy.

**Why 85% of AI project in 2022 will fail (Gartner)?**

It’s because the entire process is quite complex and requires expertise in many different facets including information retrieval, data engineering and data science. At the bare minimum, the process consists of the following steps:

* **Data Collection**
* **Storage and data flow**
* **ETL (Extract, Transform and Load)**
* **Clean up and anomaly detection**
* **Representation**
* **Aggregation and training**
* **Evaluation**
* **Optimization**

[Why top Fortune 500 companies use Cazton for Machine Learning and AI?](https://cazton.com/consulting/artificial-intelligence/artificial-intelligence-consulting)

[Know More](https://cazton.com/consulting/artificial-intelligence/artificial-intelligence-consulting)

**How Cazton can help you with PyTorch**

Our team of experts is extremely fortunate to work with top companies all over the world. We have an added advantage of creating best practices after witnessing what works and what doesn’t work in the industry. We can help you with the full-development life cycle of your products from initial consulting, development, testing, automation, deployment and scale in an on-premises, multi-cloud or hybrid environment.

* **Technology Stack:** Help create the best machine learning stack using top technologies, frameworks and libraries that suite the talent pool of your organization. This includes PyTorch, TensorFlow, Keras, Apache Spark, Scikit-learn and/or others.
* **ML Models:** Develop models, optimize them for production, deploy and scale them.
* **Best Practices:** Introduce best practices into the DNA of your team by delivering top quality PyTorch models and then training your team.
* **Models on Web and Mobile:**Incorporating PyTorch models in your existing enterprise solutions.
* **Development, DevOps and automation:**Develop enterprise apps or augment existing apps with real time ML models. This includes Web apps, iOS, Android, Windows, Electron.js app.

**Cazton Success Stories**

Our team has great expertise in building multi-million-dollar enterprise web applications (with billions in revenue per year) for our **Fortune 500** clients. Our team includes **Microsoft awarded Most Valuable Professionals, Azure Insiders, Docker Insiders, ASP.NET Insiders, Web API Advisors, Cosmos DB Insiders** as well as experts in other Microsoft as well as open-source technologies. Our team has mentored professionals worldwide by speaking at top conferences in multiple continents. With the frequent changes in operating systems, platforms, frameworks and browsers; technical folks have a lot to keep up with these days. They can’t specialize and be experts in every technology they work on. With an expert-led team like ours, we bring the **benefit of our network to you**. We save our clients a serious amount of effort on tasks they shouldn’t be doing by providing them a streamlined strategy and updating it as and when we get more information. We work with Fortune 500, large, mid-size and startup companies and **we learn what works and what doesn’t work**. This incremental learning is passed on to our clients to make sure their projects successful.

Why top Fortune 500 companies use Cazton for Big Data

[Know More](https://cazton.com/consulting/big-data-development)

Cazton is composed of technical professionals with expertise gained all over the world and in all fields of the tech industry and we put this expertise to work for you. We serve all industries, including banking, finance, legal services, life sciences & healthcare, technology, media, and the public sector. Check out some of our services:

* [Artificial Intelligence](https://cazton.com/consulting/artificial-intelligence)
* [Big Data](https://cazton.com/consulting/big-data-development)
* [Web Development](https://cazton.com/consulting/web-development)
* [Mobile Development](https://cazton.com/consulting/mobile-development)
* [Desktop Development](https://cazton.com/consulting/desktop-development)
* [API Development](https://cazton.com/consulting/api-development)
* [Database Development](https://cazton.com/consulting/database-development)
* [Cloud](https://cazton.com/consulting/cloud)
* [DevOps](https://cazton.com/consulting/devops)
* [Enterprise Search](https://cazton.com/consulting/enterprise-search)
* [Blockchain](https://cazton.com/consulting/blockchain-technologies)
* [Enterprise Architecture](https://cazton.com/consulting/enterprise)

Cazton has expanded into a global company, servicing clients not only across the United States, but in **Oslo, Norway; Stockholm, Sweden; London, England; Berlin, Germany; Frankfurt, Germany; Paris, France; Amsterdam, Netherlands; Brussels, Belgium; Rome, Italy; Sydney, Melbourne, Australia; Quebec City, Toronto Vancouver, Montreal, Ottawa, Calgary, Edmonton, Victoria, and Winnipeg** as well. In the United States, we provide our consulting and training services across various cities like **Austin, Dallas, Houston, New York, New Jersey, Irvine, Los Angeles, Denver, Boulder, Charlotte, Atlanta, Orlando, Miami, San Antonio, San Diego, San Francisco, San Jose, Stamford and others**. [Contact us](https://cazton.com/contact-us) today to learn more about what our experts can do for you.