OctoML

Secure and efficient deep learning everywhere

Octomizer Outline

Who we are (recap)

Deployment pain

The vision

The Octomizer: TVM for everyone

OctoML

Simple, secure, and efficient
deployment of ML models in
the edge and the cloud

Drive TVM adoption
Core infrastructure
and improvements

Expand the set of users who can
deploy ML models:
Services, automation, and
integrations

Apache TVM ecosystem

OctoML

Founding Team - The Octonauts

Luis Ceze
Co-founder, CEO
PhD in Computer Architecture and Compilers
Professor at UW-CSE
Venture Partner, Madrona Ventures
Previously: IBM Research, consulting for Microsoft, Apple, Qualcomm

Jason Knight
Co-founder, CPO
PhD in Computational Biology and Machine Learning
Previously: HLI, Nervana, Intel

Tianqi Chen
Co-founder, CTO
PhD in Machine Learning
Professor at CMU-CS

Thierry Moreau
Co-founder, Architect
PhD in Computer Architecture

Jared Roesch
Co-founder, Architect
(soon) PhD in Programming Languages

40+ years of combined experience in computer systems design and machine learning

Deployment Pain/Complexity

- Model ingestion
- Performance estimation and comparison
  - Cartesian product of models, frameworks, and hardware
- Optimization
  - O0, O1, O2
  - Target settings: march, mtune, mcpu
  - Size reductions
  - Quantization, pruning, distillation
- Custom operators (scheduling, cross hardware support)
- Lack of portability / varying coverage across frameworks
- Model integration
  - Output portability
  - Packaging (Android APK, iOS ipa, Python wheel, Maven artifact, etc)

Deep learning deployment should be easy.
For *everyone.*

TVM is core to making that happen.

… but it's only the first (important!) step

The Machine Learning Lifecycle

Data collection, curation, annotation

Model training

Model development

OctoML

Model optimization
- Quantization
- Custom kernels
- Framework modifications
- Hardware vendor partnerships

Deployment
- Packaging
- Binary size
- Integration
- Build chain setup

Cloud inference

Edge/embedded inference

Octomizer: deep learning optimization as a service

TensorFlow, Pytorch, ONNX
serialized models

API and web UI

Octomizer

Support for efficient
and secure execution

Microsoft Azure

Amazon EC2

Google Cloud Platform

Optimize over multiple clouds for training and inference at scale.

Better latency, lower OP ex.

Optimize for edge deployment.

Longer battery life, smaller form factor, lower part cost, etc.

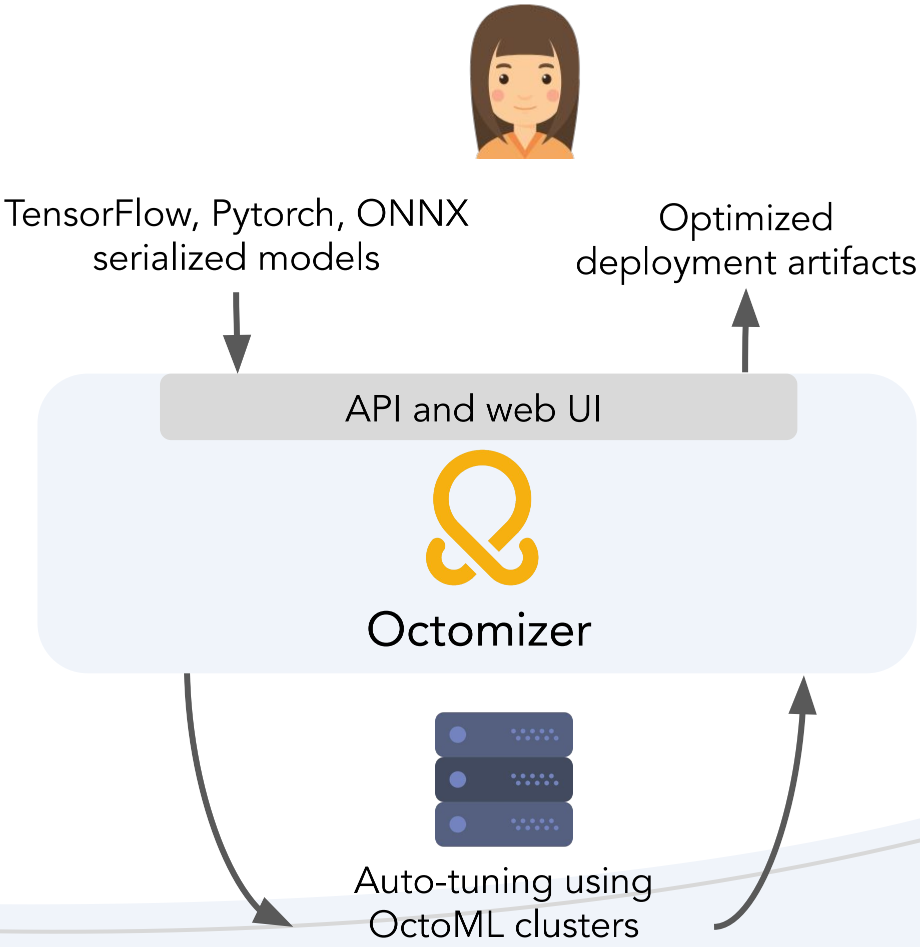
Demo (frontend and optimization)

- Simple, easy to use Python API
  - `pip install octomizer`
  - `export OCTOML_ACCESS_TOKEN=...`

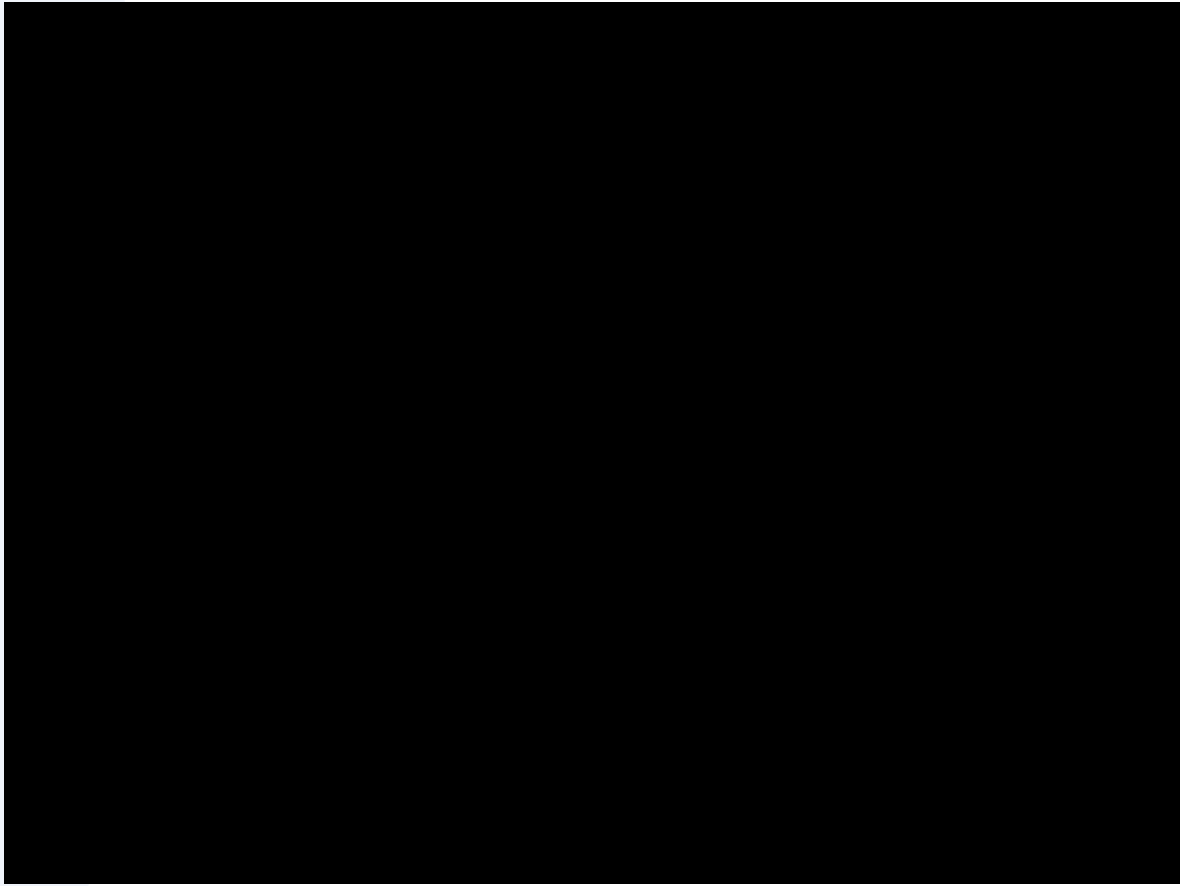
```
import octomizer
model = octomizer.upload(model, params, 'resnet-18')
job = model.start_job('autotvm', {  # also 'onnxrt' etc...
   'hardware': 'gcp/<instance_type>',
   'TVM_NUM_THREADS': 1,
   'tvm_hash': '...'
})
while job.get_status().status != 'COMPLETE':
   sleep(1)
model.download_pkg("base_model", 'python')  # Package with default schedules
model.download_pkg("optimized_model", 'python', job)
```

Octomizer optimization

- Code generation of operator library
  - Auto-tuning per hardware target, operator, and operator parameters
- Hardware targets supported:
  - GCP cloud instances
  - ARM A class CPU/GPU
  - ARM M class microcontrollers
- On the roadmap:
  - AWS and Azure cloud instances
  - Quantization
  - Hardware-aware architecture search
  - Compression/distillation



Demo (visualization)



Octomizer under the hood

- Entire stack designed for easy, cross-cloud and private cloud/on-prem deployment
- Consists of:
  - Kubernetes
  - Kustomize for declarative deployments
  - Rust + Actix-web for robust, safe and simple deployments
  - Only external service dependency is an object store
  - Support for TVM RPC Trackers for external device management/execution
- OctoML hosted Octomizer today supports
  - GCP cloud instances
  - ARM A class CPU/GPU
  - ARM M class microcontrollers
  - More to come...

ML Workloads and Requirements

Upcoming Hardware
(accelerator, SOC,
HW IP blocks, ...)

Stay tuned...

OctoML

Existing HW

- CPU
- GPU
- FPGA
- uControllers

Focus today

Efficient and secure execution
(and perf/power estimation)

Next steps

Stay tuned through twitter (@octoml) or email.

Reach out if you have use cases to share: jknight@octoml.ai

Looking for private beta partners.

We are hiring see octoml.ai for more details!