From research to production
Enabling real world applications with Conversational AI
Adam Henryk Grzywaczewski & Zenodia Charpy
About Me

Zenodia Charpy

- Senior Solution Architect @ NVIDIA - Auto & Healthcare focus on Deep Learning.

- My past experience:
  - Azure : Data Scientist & Solution Architect
  - Telia : Data Scientist
About Me

Adam Grzywaczewski

- Senior Deep Learning Data Scientist @ NVIDIA - Supporting delivery of AI / Deep Learning solutions
- Specialising in Deep Learning at scale.
- My past experience:
  - Capgemini: https://goo.gl/MzgGbq
  - Jaguar Land Rover Research: https://goo.gl/ar7LuU
Alex: Find us a Mexican restaurant
Jarvis: The nearest Mexican Restaurant is Luna Kitchen
Conversational AI

Conversational AI Skills

**ASR**
- Batched mode
- Streaming mode

**TTS**
- Batched mode
- Streaming mode

**NLU**
- Intent Classification
- Slot Filling
- NER
- Punctuation
- Classify Text/Token
- Transform Text

**CV**
- Emotion Classify
- Gaze Estimate
- Facial Detect
- Facial Landmark Detect
- Head Pose
- Body Pose (2D)
- User Attributes Retrieval

Triton Inference Server
Tensor RT | NVIDIA Docker | CUDA-X AI

NVIDIA GPUs (Datacenter, Embedded)
The Workflow

Research, Optimization, Deployment

Jarvis
An integrated & Scalable solution for serving Conversational AI models

NeMo / TLT

NeMo: Open source applied research toolkit for researching, developing and training conversational AI models

TLT: Transfer Learning Toolkit is a 0-coding tool for training supported Conversational AI models

TensorRT + Triton


Triton: Open-source inference serving software.
NeMo

Open-Source Toolkit for Development of Conversational AI Models

Automatic Speech Recognition
- Spoken word to text transcription

Natural Language Processing
- Understanding tasks
- Named Entity Recognition
- Question Answering
- Dialog Management
- Machine Translation

Text to Speech
- Text to spoken language
What is NeMo
Encapsulation of Best Practice
Broad adoption

Welcome Contribution and Enable Collaboration
Enabling R&D

Build for Flexibility and Ease-of-Use

User Applications / Scripts
- Voice Recognition
- Natural Language
- Speech Synthesis

Pretrained Models

NeMo Collection Libraries
- nemo_asr
- nemo_nlp
- nemo_tts

PyTorch Lightning & Facebook Hydra
PyTorch accelerated by CUDA-X

github.com/NVIDIA/NeMo
Neural Modules

Lego-Like Building Blocks Enabling Fast Experimentation

- Neural Modules, building blocks of conversational AI
- Typed inputs and outputs
- Easy management of experiments (Hydra)
- Integration with PyTorch Lightning
- Highly scalable and performant (FP16, Distributed)
- Extensive collection of pretrained models
• Ensure compatibility between modules
• Catch semantic, rank and dimensionality mismatches
• Simplifying the debug process

Highlight 1

Strong Typed Tensors

Valid neural type connection
Invalid neural type connection
Not interchangeable

inpt = NeuralType(('B', 'D', 'T'), SpectrogramType())
Out1 = NeuralType(('B', 'D', 'T'), MelSpectrogramType())
Out2 = NeuralType(('B', 'D', 'T'), MFCCSpectrogramType())

Tutorial notebooks available at https://github.com/NVIDIA/NeMo
Hydra: Simplify Complex Model Development

Flexible approach for developers to configure and customize the model

Single-stop solution for editing the end-to-end neural network

Configurable with both YAML and Hydra CLI commands

# specify the name of the model you want to use
name: &name "QuartzNet15x5"

# manage training data parameters
train_ds:
  manifest_filepath: ???
  sample_rate: 16000
  ...

# manage validation data parameters
validation_ds:
  manifest_filepath: ???
  sample_rate: 16000
  ...

QuartzNet Model Customization with .YAML file
Up to 4.5x Faster Training on Single GPU, Scale to Multiple GPUs Easily

Tight integration with PyTorch Lightning Trainer to easily invoke training actions.

Scale to multi-GPU and multi-node to speed-up training while retaining the accuracy

Speed-up training up to 4.5X on a single GPU with mixed-precision versus FP32 precision

Ease to use parameters to enable Multi-GPU/node training and mixed-precision

```
trainer = pl.Trainer(**cfg.trainer)
asr_model = EncDecCTCModel(cfg=cfg.model, trainer=trainer)
trainer.fit(asr_model)
```

Training NeMo model with PyTorch Lightning Trainer API

Training at Scale - Multi GPU and Multi Node Training

Reduce total training time
Distribute workload onto multiple compute instances with a single parameter change
Maintain very high accuracy (Word Error Rate)
Model Zoo

Key Area of Focus

Speech Processing
- Automatic Speech Recognition
- Speech Classification
- Speaker Recognition
- Speaker Diarization

Natural Language Processing
- Punctuation & Capitalization
- Token Classification (NER)
- Joint Intent and Slot Classification
- Text Classification
- Question Answering
- Dialogue State Tracking
- Information Retrieval
- Machine Translation
- Language Modelling (other tasks)

Text to speech
- Two stage pipelines
- End to end pipelines
Model Zoo

ASR

Models
- Jasper
- QuartzNet
- Citrinet
- Conformer

Languages
- English
- Mandarin
- German
- Polish
- Italian
- Russian
- Spanish
- Catalan

https://docs.nvidia.com/deeplearning/nemo/
Speech Command Recognition
The task of classifying an input audio pattern into a discrete set of classes.

Audio Sentiment Classification
Extends the conventional text-based sentiment analysis to depend on the acoustic features extracted from speech.

Speaker Identification
Who is speaking?

Speaker Verification
Is the speaker who they claim to be?

Voice Activity Detection (VAD)
The task of predicting which parts of input audio contain speech versus background noise.

Many other
....
Model Zoo

Two Stage Pipeline

- Mel-Spectrogram Generation
- Audio Generation (Vocoder)

End to End Pipeline

Audio Generation (TextToWaveform)
Model Zoo

TTS

Two Stage Pipeline

Mel-Spectrogram Generation
- Tacotron2
- GlowTTS
- FastSpeech2
- FastPitch
- TalkNet

Audio Generation (Vocoder)
- WaveGlow
- SqueezeWave
- UniGlow
- MelGAN
- HiFiGAN

End to End Pipeline

Audio Generation (TextToWaveform)
- Wavenet
- DeepVoice 3
- 2 Stages in End-to-End
  - FastPitch_HifiGan_E2E
  - FastSpeech2_HifiGan_E2E
What’s Next

Zenodia Charpy

- Model Graduation
- Optimisation and deployment
GRADUATING THE MODEL
Graduating the model

Three Aspects of The Problem

Ensuring code & data compliance

High Performance

Strive for SOTA
Ensuring code & data compliance

Crediting | Citing | License Compliance

- Double check datasets’ legal compliance
- Appropriate crediting & citing in NeMo code

Model graduation

Ensuring code & data compliance

High Performance

Strive for SOTA
Striving towards State Of The Art

Benchmark | Iterative Improvement

- Authoritative test dataset for benchmarking
- Iterate until SOTA is reached

Ensuring code & data compliance

Model graduation

High Performance

Strive for SOTA
Ensuring high performance

Code Quality | Model Size | Throughput

- Code quality matters
- Shrink model size & post-quantization for throughput

High Performance

Ensuring code & data compliance

Model graduation

Strive for SOTA

OPTIMIZATION
NVIDIA TensorRT

From Every Framework, Optimized for Each Target Platform
Why should we optimize (with TensorRT)

Table 1: Comparison of PyTorch and TensorRT TTS inference latencies on 1x NVIDIA T4 GPU

What are we optimizing

Layer & Tensor Fusion

Precision Calibration

Kernel Auto-Tuning

Trained Neural Network

Dynamic Tensor Memory

Multi-Stream Execution

Optimized Inference Engine
Deploy & Serve with Triton Inference Server

Open-Source Software for Scalable Inference Serving

- Dynamic Batching (Real time, Batch, Stream)
- Per Model Scheduler Queues
- Flexible Model Loading (All, Selective)
- Multiple GPU & CPU Backends
  - TensorFlow
  - PyTorch
  - ONNX
  - Custom
- Standard HTTP/gRPC
- Utilization, Throughput, Latency Metrics

AI Application

Query → Result

GPU → CPU

Model Store

Metrics

Kubernetes, Prometheus
NVIDIA Top MLPERF Data Center Benchmarks

GPU is Faster Than CPU

MLPerf v1.0 Inference Closed; Per-accelerator performance derived from the best MLPerf results for respective submissions using reported accelerator count in

Data Center Offline and Server: 3D U-Net 99%: 1.0-19, 1.0-53, 1.0-54, 1.0-56, 1.0-30 ResNet-50: 1.0-17, 1.0-53, 1.0-41, 1.0-35, 1.0-54, 1.0-56, 1.0-30, RNN-T: 1.0-20, 1.0-54, 1.0-56, 1.0-30 SSD-Large: 1.0-17, 1.0-53, 1.0-35, 1.0-54, 1.0-56, 1.0-30 DLRM 99%: 1.0-18, 1.0-54, 1.0-56, 1.0-30, BERT 99%: 1.0-32, 1.0-54, 1.0-56, 1.0-30.

MLPerf name and logo are trademarks. See www.mlcommons.org for more information.
Deploying large NLP models

Triton Inference Server Hosting 4 Copies of GPT-3 18B on DGX A100

NVLinks to connect different parts of GPT-3
Deploying large NLP models

TensorRT + Triton Inference Server Deliver the Highest Throughput

GPT-3 18B parameters. Inference cases per 1 x DGX A100 40GB

- 8xA100 40GB: 721X higher throughput
- 2xA100 40GB: 274X higher throughput

CPU: 1X throughput

Inference throughput comparisons (requests / per second)
GPT-3 MEGATRON-LM 18B parameters. Seq_length=1024.

*CPU case: Dual AMD Rome 7742, 128 cores total, 2.25 GHz (base), 3.4 GHz (max boost), FP32. Container: nvcr.io/nvidia/pytorch:21.03-py3 OnnxRuntime-CPU version. Code was not properly optimized for this processor, so with better optimization, the difference in results between GPU and CPU may differ multiple times.

Put It All Together = Multimodal Conversational AI

Develop, Train and Fine-Tune AI Models using Transfer Learning Toolkit & NeMo Then Deploy in Jarvis

Data Sources

- Customer Data
- Pretrained Models

Model Training Fine Tuning

- NeMo Or Transfer Learning Toolkit

Model Validation

Model Export

Jarvis API server

- gRPC
- Jarvis Inference Server
- Triton Inference Server

Jarvis AI Services At Scale

- gRPC
- Jarvis API server
- Triton Inference Server
- GPU
- GPU
What’s Next
Adam Grzywaczewski

- Packaging
- Final remarks
NVIDIA Jarvis use cases
A Flexible & Extendible Framework for Conversational AI

In-car experience (@Youtube)

Chatbot with A2F
(JHH Keynote @Youtube)

Misty: making of 3D AI assistant (@Youtube)

Virtual assistant with multi-domain conversation (recorded)

Call center transcription and annotation (recorded)

Virtual assistant with digital avatar (recorded)
NVIDIA Transfer Learning Toolkit

Bring Your Own Data (BYOD)

- Increase accuracy by fine-tuning on proprietary data
- Zero Coding approach reduces barrier to entry for enterprises
- Use Tensor Cores to achieve highest training performance
- Integrated with Jarvis to deploy models as real-time services
Try Jarvis and NeMo today

NeMo
github.com/NVIDIA/NeMo

Jarvis
developer.nvidia.com/nvidia-jarvis-getting-started

TLT
developer.nvidia.com/transfer-learning-toolkit
NVIDIA DEEP LEARNING INSTITUTE

Hands-on training in deep learning, accelerated computing, and accelerated data science for developers, data scientists and researchers

Introductory courses on AI for IT professionals

Take self-paced training online, view instructor-led training catalog and upcoming public workshop schedule, and learn about university resources at www.nvidia.com/dli

For a consultation contact us: nvdli@nvidia.com
Try Jarvis and NeMo today

NeMo
github.com/NVIDIA/NeMo

Jarvis
developer.nvidia.com/nvidia-jarvis-getting-started

TLT
developer.nvidia.com/transfer-learning-toolkit