Interactive
Long-Form
Conversational
Speech &
Language
Application
Roadmap

1. Fundamental Challenges
2. Case Study: Closed Loop Feedback in Speech Recognition Training
3. Case Study: Latency and Randomness in Speech Synthesis Inference
FUNDAMENTAL CHALLENGES
<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimize</strong></td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
<td>Scale ⚭</td>
</tr>
<tr>
<td><strong>Manage</strong></td>
<td>Jitter &amp; Aperiodicity 🌊</td>
<td>Nonlinearity 🦋</td>
<td>Interactivity 🧵</td>
</tr>
</tbody>
</table>
LATENCY
<table>
<thead>
<tr>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimize</strong></td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
</tr>
<tr>
<td><strong>Manage</strong></td>
<td>Jitter &amp; Aperiodicity 🎨</td>
<td>Nonlinearity 🦋</td>
</tr>
</tbody>
</table>
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.

ADC Sample Capture  
Encoding Delay  
Packetization Delay  
Output Queue  
Access Up Delay  
Backbone Delay  
Access Down Delay  
Application Input Queue  
Jitter Buffer  
Decoding Delay  
Device Playout Delay
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.
INTERACTIVITY
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
<td>Scale 👇</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🌊</td>
<td>Nonlinearity 🦋</td>
<td>Interactivity</td>
</tr>
</tbody>
</table>
Simple function
The Tent Map

\[ F(x) = \mu \min(x, 1 - x) \]
The Tent Map

\[ F(x) = \mu \min( x, 1 - x ) \]
The Tent Map

\[
\frac{\mu}{2}
\]

\[
\frac{1}{2}
\]
The Tent Map ($\mu < 1$)
The Tent Map ($\mu > 1$)
The Tent Map ($\mu >> 1$)
The Tent Map
Bifurcation Diagram
More Complex Function
Feature a Neural Network

ASR → Dialogue System → Speech Synthesis → The Internet → PSTN → LTE Tower Networks

Phones & Internet Again

Microphone → Sound Waves → Vocal Tract → Brain → Cochlea → Sound Waves

iPhone
Physical Effects

ASR → Dialogue System → Speech Synthesis → The Internet → PSTN → LTE Tower Networks

Phones & Internet Again → Microphone → Sound Waves → Vocal Tract → Brain → Cochlea → Sound Waves

iPhone
Noisy & Stochastic

ASR → Dialogue System → Speech Synthesis → The Internet → PSTN → LTE Tower Networks

Phones & Internet Again

Microphone → Sound Waves → Vocal Tract → Brain → Cochlea → Sound Waves

iPhone
Nonlinearity and Distortion

ASR → Dialogue System → Speech Synthesis → The Internet → PSTN → LTE Tower Networks

Phones & Internet Again

Microphone → Sound Waves → Vocal Tract → Brain → Cochlea → Sound Waves

iPhone
Information Loss

ASR → Dialogue System → Speech Synthesis → The Internet → PSTN → LTE Tower Networks

Phones & Internet Again

Microphone → Sound Waves → Vocal Tract → Brain → Cochlea → Sound Waves

iPhone
Consumes Latency Budget

- ASR
- Dialogue System
- Speech Synthesis
- The Internet
- PSTN
- LTE Tower Networks
- Phones & Internet Again
- Microphone
- Sound Waves
- Vocal Tract
- Brain
- Cochlea
- Sound Waves
- iPhone
<table>
<thead>
<tr>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency ☟</td>
<td>Stochasticity ☶</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scale 🕒</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🧵</td>
<td>Nonlinearity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interactivity 🎉</td>
</tr>
</tbody>
</table>
I. \( F(x + y) = F(x) + F(y) \)
San Jose is a city in California.
Machine learning as a statistical computing technique.
F(x+y)

0:00:01  That needs to be easy to California.
Linearity

II. $F(ax) = aF(x)$
0:00:00  San Jose is a city in California.
F(\alpha x)
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
<td>Scale ⬇️</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🕳️</td>
<td>Nonlinearity 🦋</td>
<td>Interactivity 🐸</td>
</tr>
</tbody>
</table>

Time: ⌛️
Predictability: 🎲
Complexity: ⬇️
Physical Signals and Noise → \( \Sigma \) → Nonlinearity → Stochasticity

Interactivity & Feedback Loops
Physical Signals and Noise → \( \Sigma \) → Nonlinearity → Stochasticity

Interactivity & Feedback Loops → Nonlinearity
JITTER & APERIODICITY
New England Load - Super Bowls 49 and 50

New England vs. Seattle, 2/1/2015

Denver vs. Carolina, 2/7/2016

Start of Game

Half Time

End of Game
Typical US Call Center Diurnal (Weekday)

Start of the work day (eastern time zone)

Lunch break
Typical US Call Center Diurnal (First of month)

Start of the work day (eastern time zone)

Lunch break
SCALE
\( f(V) \)
API Web Server

\[ f(V) \]
CASE STUDY:
CLOSED LOOP FEEDBACK
IN SPEECH RECOGNITION
TRAINING
<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
<td>Scale ⚖️</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🌊</td>
<td>Nonlinearity 🦋</td>
<td>Interactivity 🎉</td>
</tr>
</tbody>
</table>
Speech Recognition

“Welcome to Gridspace”
Speech Recognition

< signal processing >

< Language>

< pronunciation model>

< acoustic model>
Speech Recognition - History

The Switchboard corpus is a collection of recorded telephone conversations widely used to train and test speech-recognition systems.

Sources: Microsoft; research papers
Data Boost

Data(Hrs) vs. Year

Year

Data(Hrs)
Computation Boost

GPU Performance (FP32, single precision floating point)

(from: Grigory Sapunov, Intento blogs)
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.

Model Boost

(from: Ming Sun et el., IS 2017 paper)

(from: Kyu Han et el., IS 2018 paper)
Speech recognition in the Wild - Still Challenging

“When my oldest child was a baby...”

“It didn't send me any kind of verification code. It did pop up a message saying that the account was locked”

<table>
<thead>
<tr>
<th>Switchboard</th>
<th>DATA</th>
<th>Gridspace Call Center Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal vocab, normal emotion</td>
<td>Speech Type</td>
<td>domain specific vocab, emotional modulated speech</td>
</tr>
<tr>
<td>7.5%</td>
<td>Word Error Rate</td>
<td>18% (without optimization)</td>
</tr>
</tbody>
</table>
Speech recognition in the Wild - Still Challenging

Each domain has each specific language.

- command query
- financial
- media
- customer support
Speech recognition in the Wild - Still Challenging

There is lots of **variabilities** in speech

- accent
- noise
- emotion modulated speech
- mis-pronunciation
Speech recognition in the Wild - Still Challenging

It’s hard to learn all beforehand!
Speech recognition in the Wild - in Practice
Speech recognition in the Wild - in Practice

Train a model

Evaluate model

Train model

Deploy to production

Deploy the model

Fetch

Generate example data

Clean

Prepare
## Use of Unsupervised Data

<table>
<thead>
<tr>
<th></th>
<th>SWBD</th>
<th>Librispeech</th>
<th>GS-SupData</th>
<th>GS-CS2018</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hours</strong></td>
<td>0.3k</td>
<td>1k</td>
<td>30k</td>
<td>17M (2000 years)</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Supervised</td>
<td>Supervised</td>
<td>Supervised</td>
<td>Unsupervised</td>
</tr>
</tbody>
</table>
Use of Unsupervised Data

Semi-supervised training (SST)

- a good way to use unsupervised data for supervised tasks
- It has to deal with uncertainties
- We can update AM and LM iteratively
Use of Unsupervised Data

Data selection for SST

- 100% use (no filtering) of Unsupervised data would cause model’s degradation on accuracy
- knowledge based selection helps
  - confidence score, length, topic, speaker info
Use of Unsupervised Data

Unlabeled Audio

ASR on K8 Clusters

Machine Transcripts

Data Selection / Weighting / Auto Correction

Training Data with Transcription

ASR update with Evaluation

AM & LM Training on GPU Clusters
Continuous Learner

- accent learning
- noise learning
- language/grammar learning

... and become better learner over time
## Continuous Learner

### Throughput

<table>
<thead>
<tr>
<th></th>
<th>P-100</th>
<th>V-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST training / 1 sec / 1 gpu</td>
<td>450 sec</td>
<td>580 sec</td>
</tr>
<tr>
<td>SST training / 24 hours / 1 gpu</td>
<td>10800 hours</td>
<td>13920 hours</td>
</tr>
</tbody>
</table>

* Training for Acoustic Model(Resent TDNN), 150 frames per example, 64 example per minibatch
Continuous Learner

Ideal Error Graph?

(Word Error Rate)

Call Volume (days of audio)

Time
## Continuous Learner

### Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Unmatched Domain</th>
<th>Supervised Training</th>
<th>Semi-Sup Training</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word Error Rate</strong></td>
<td>14.29%</td>
<td>9.52%</td>
<td>7.83%</td>
</tr>
<tr>
<td><strong>System Building Hours</strong></td>
<td></td>
<td>3 months</td>
<td>&lt; 1 DAY</td>
</tr>
</tbody>
</table>
CASE STUDY:
STOCHASTICITY AND LATENCY IN SPEECH SYNTHESIS INFERENCE
Does this bot understand me?

01101000011110
01010110101010
10101010000111
1000101001111
Dialog System Evaluators

- Content
- Naturalness
- Cadence
Influenced by TTS

- Content
- Naturalness
- Cadence
Modern Neural TTS

“I’m speech that came from a big neural network”
WE HARNESS THE POWER OF THE MACHINE TO ENHANCE CUSTOMER RELATIONSHIPS.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency 🕒</td>
<td>Stochasticity 🎲</td>
<td>Scale ⚖</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🌊</td>
<td>Nonlinearity</td>
<td>Interactivity</td>
</tr>
</tbody>
</table>
CNN/RNN

RNN
RNN
RNN
RNN

"g" "r" "a" "c" "e"
Influenced by TTS

Content

Naturalness

Cadence
<table>
<thead>
<tr>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>Latency 🕗</td>
<td>Stochasticity 🎲</td>
</tr>
<tr>
<td>Manage</td>
<td>Jitter &amp; Aperiodicity 🌊</td>
<td>Nonlinearity □</td>
</tr>
</tbody>
</table>
Can you transfer some money please?
Hello?
Please...

How can I help you?
.....

Hello
Where does the latency come from?
WaveNet Vocoding

Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

WaveNet

Training

Inference
Vocoder Evolution

WaveNet 2016 → Parallel WaveNet 2017 → WaveGlow 2018
WaveGlow

Training

Inference
from tensorflow.python.compiler.tensorrt.trt_convert import TrtGraphConverter

converter = TrtGraphConverter(
    input_saved_model_dir='my_saved_model',
    precision_mode=FP16
)

converter.convert()

converter.save('trt_saved_model')
## Tesla V100 Latency (batch size=1)

<table>
<thead>
<tr>
<th>Precision</th>
<th>Latency /ms</th>
<th>Samples per second /Hz</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32</td>
<td>277</td>
<td>520</td>
<td>1x</td>
</tr>
<tr>
<td>FP16 (TRT)</td>
<td>196</td>
<td>735</td>
<td>1.4x</td>
</tr>
</tbody>
</table>
Can we go faster?
INT8 Calibration

$2^{31} - 1$

$-2^{31} + 1$

128

-128
from tensorflow.python.compiler.tensorrt.trt_convert import TrtGraphConverter

converter = TrtGraphConverter(
    input_saved_model_dir='my_saved_model',
    precision_mode='INT8'
)
converter.convert()
converter.calibrate(
    fetch_names=['output:0'],
    num_runs=1000,
    input_map_fn=get_examples
)
converter.save('trt_saved_model')
## Tesla V100 Latency (batch size=1)

<table>
<thead>
<tr>
<th>Precision</th>
<th>Latency /ms</th>
<th>Samples per second /Hz</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32</td>
<td>277</td>
<td>520</td>
<td>1x</td>
</tr>
<tr>
<td>FP16 (TRT)</td>
<td>196</td>
<td>735</td>
<td>1.4x</td>
</tr>
<tr>
<td>INT8 (TRT)</td>
<td>164</td>
<td>878</td>
<td>1.7x</td>
</tr>
</tbody>
</table>
Deployment

Google Container Engine

Tensorflow Serving
<table>
<thead>
<tr>
<th>Time</th>
<th>Predictability</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize</td>
<td>✓ Latency 🕒</td>
<td>✓ Stochasticity 🎲</td>
</tr>
<tr>
<td>Manage</td>
<td>✓ Jitter &amp; Aperiodicity 🌊</td>
<td>✓ Nonlinearity 🦋</td>
</tr>
</tbody>
</table>

Enables...
THE FOLLOWING IS A REAL INTERACTION
Access the demo video:

[Video on Vimeo Here](#)
Want to demo Grace at GTC?
Do you operate a call center?
Do you need speech processing or automation?

gtc@gridspace.com

Do you want to work at Gridspace?

hiring@gridspace.com

Thank you everyone!