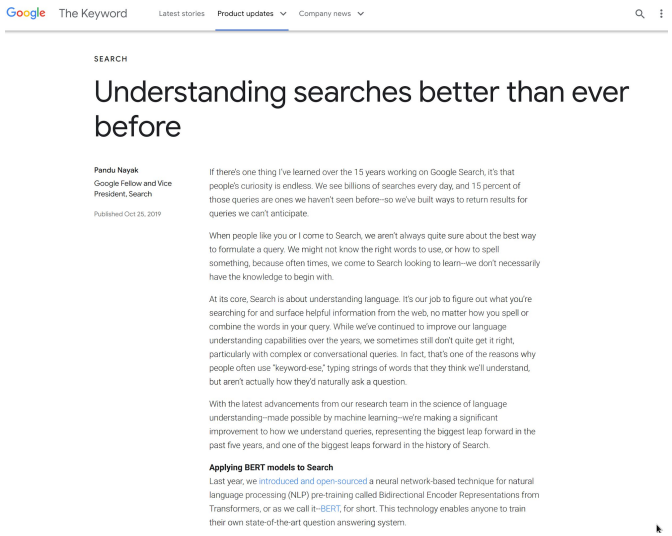
Myrtle.ai

Achieving Low-latency Speech Synthesis at Scale

Sam Davis

Machine Learning Powers Products

Myrtle.ai



Google The Keyword Latest stories Product updates Company news

SEARCH

Understanding searches better than ever before

Pandu Nayak
Google Fellow and Vice President, Search

Published Oct 25, 2019

If there's one thing I've learned over the 15 years working on Google Search, it's that people's curiosity is endless. We see billions of searches every day, and 15 percent of those queries are ones we haven't seen before--so we've built ways to return results for queries we can't anticipate.

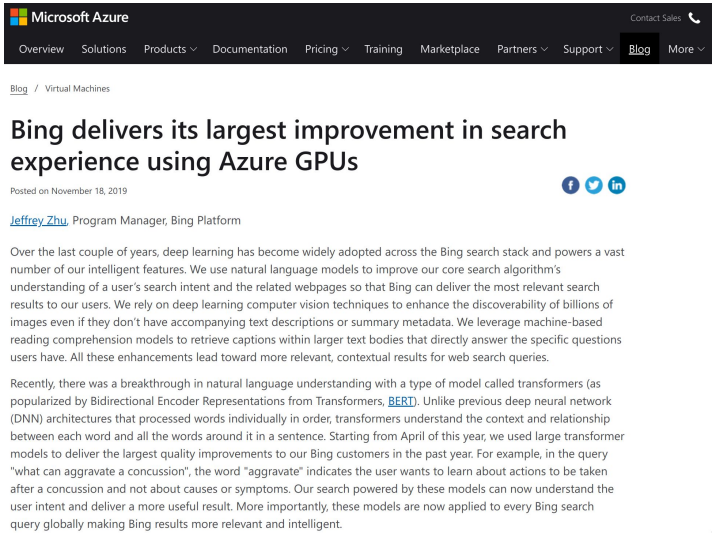
When people like you or I come to Search, we aren't always quite sure about the best way to formulate a query. We might not know the right words to use, or how to spell something, because often times, we come to Search looking to learn--we don't necessarily have the knowledge to begin with.

At its core, Search is about understanding language. It's our job to figure out what you're searching for and surface helpful information from the web, no matter how you spell or combine the words in your query. While we've continued to improve our language understanding capabilities over the years, we sometimes still don't quite get it right, particularly with complex or conversational queries. In fact, that's one of the reasons why people often use "keyword-ese," typing strings of words that they think we'll understand, but aren't actually how they'd naturally ask a question.

With the latest advancements from our research team in the science of language understanding--made possible by machine learning--we're making a significant improvement to how we understand queries, representing the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search.

Applying BERT models to Search

Last year, we introduced and open-sourced a neural network-based technique for natural language processing (NLP) pre-training called Bidirectional Encoder Representations from Transformers, or as we call it--BERT, for short. This technology enables anyone to train their own state-of-the-art question answering system.



Microsoft Azure Contact Sales

Overview Solutions Products Documentation Pricing Training Marketplace Partners Support Blog More

Blog / Virtual Machines

Bing delivers its largest improvement in search experience using Azure GPUs

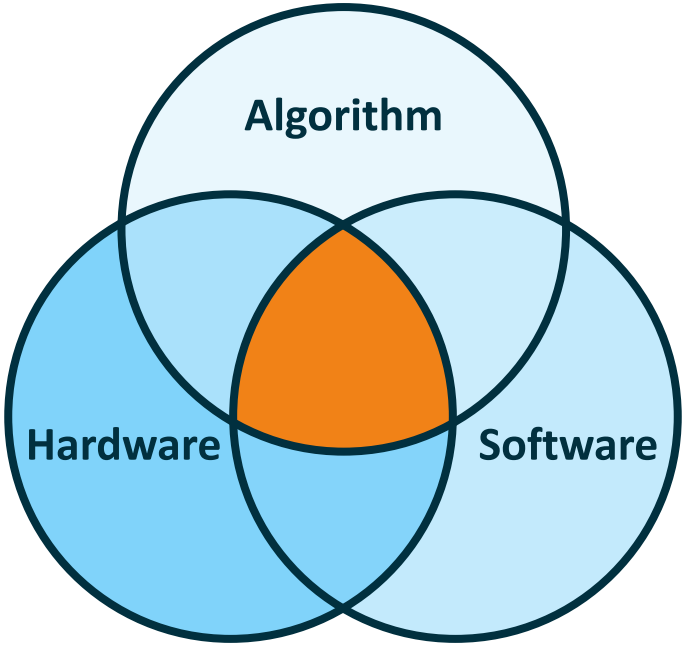
Posted on November 18, 2019

Jeffrey Zhu, Program Manager, Bing Platform

Over the last couple of years, deep learning has become widely adopted across the Bing search stack and powers a vast number of our intelligent features. We use natural language models to improve our core search algorithm's understanding of a user's search intent and the related webpages so that Bing can deliver the most relevant search results to our users. We rely on deep learning computer vision techniques to enhance the discoverability of billions of images even if they don't have accompanying text descriptions or summary metadata. We leverage machine-based reading comprehension models to retrieve captions within larger text bodies that directly answer the specific questions users have. All these enhancements lead toward more relevant, contextual results for web search queries.

Recently, there was a breakthrough in natural language understanding with a type of model called transformers (as popularized by Bidirectional Encoder Representations from Transformers, BERT). Unlike previous deep neural network (DNN) architectures that processed words individually in order, transformers understand the context and relationship between each word and all the words around it in a sentence. Starting from April of this year, we used large transformer models to deliver the largest quality improvements to our Bing customers in the past year. For example, in the query "what can aggravate a concussion", the word "aggravate" indicates the user wants to learn about actions to be taken after a concussion and not about causes or symptoms. Our search powered by these models can now understand the user intent and deliver a more useful result. More importantly, these models are now applied to every Bing search query globally making Bing results more relevant and intelligent.

Powering the Future

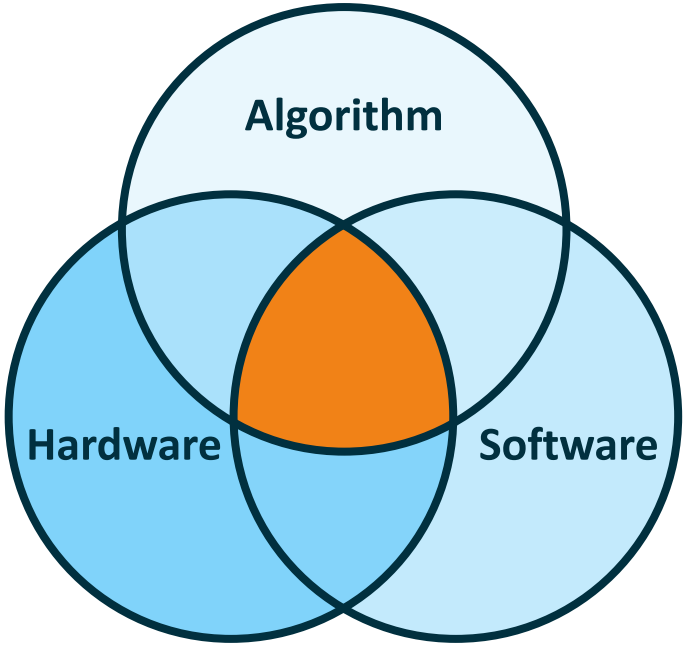


Powering the Future

Myrtle.ai

"The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation."

- Rich Sutton, [The Bitter Lesson]



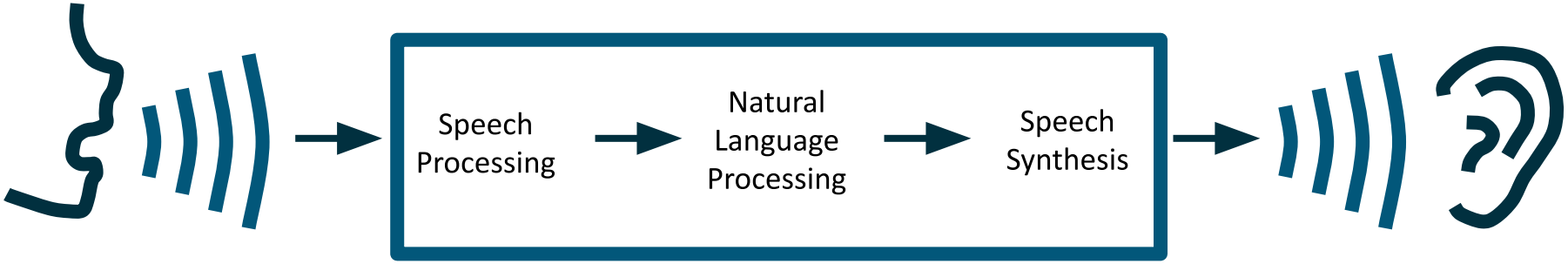
Talk Outline

- Conversational AI Overview
- Speech Synthesis Solution Deep Dive
  - Algorithm
  - Hardware
  - Software
- Solution Demonstration and Performance

Myrtle.ai

Conversational AI

Conversational AI: What?



Conversational AI: Challenges

Myrtle.ai

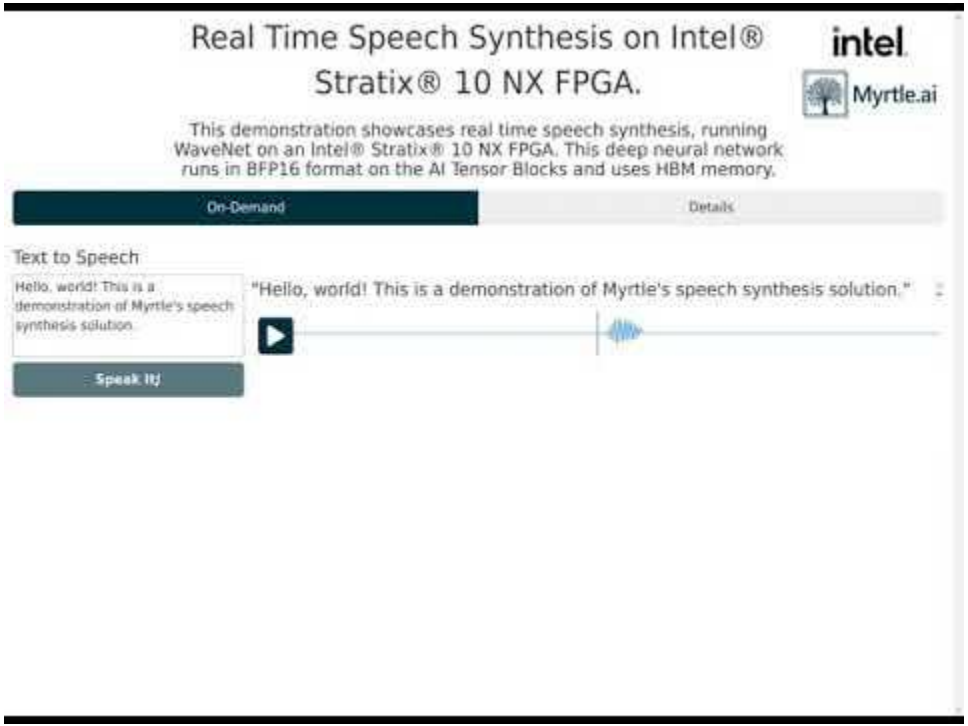
- Users!
  - Large volume of application requests
  - Each request uses multiple models
  - Each model has large throughput requirement
- Quality
  - Consistently return high quality results to satisfy users
  - Drives increase in model size and complexity
- Latency
  - Real-time interaction sets application latency bound to 100’s ms
  - Model tail latencies critical to meet bound
  - Drives decrease in model size and complexity

Myrtle.ai

Solution

Speech Synthesis at Scale

Speech Synthesis Demo



Myrtle.ai

Algorithm

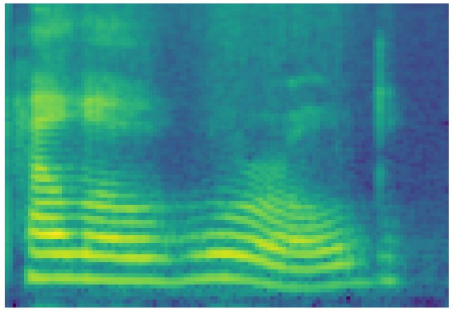
Neural Speech Synthesis

Neural Speech Synthesis Overview

Myrtle.ai

Hello, world!

Tacotron 2



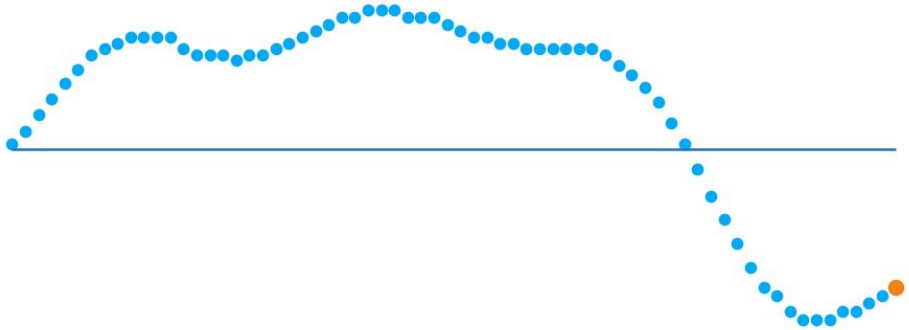
WaveNet

WaveNet: A Generative Model for Raw Audio

Myrtle.ai

WaveNet models:

$p(y_t \mid y_{<t}, f)$



y_t Audio sample at step t. e.g. for 16-bit audio: y_t is in $[0, 2^{16})$

$y_{<t}$ All previous audio samples

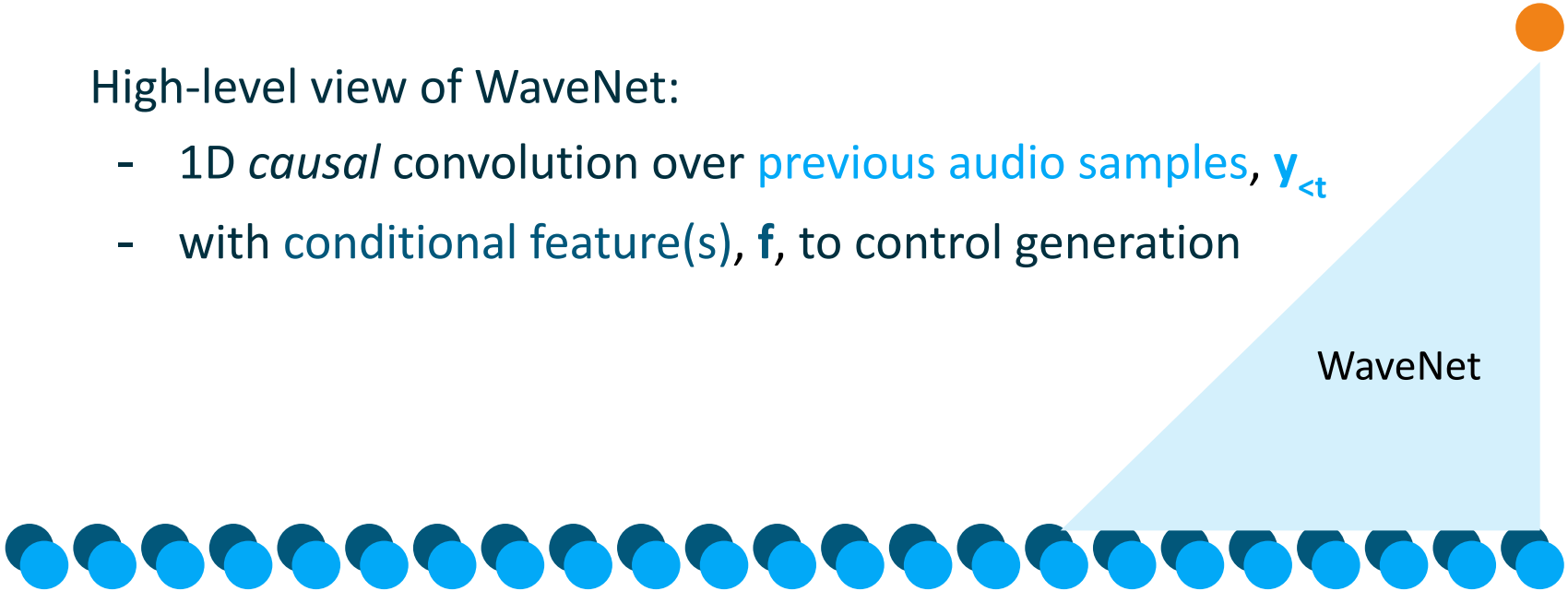
f Feature(s) to condition on. e.g. spectrogram

WaveNet: Model Overview

$p(y_t \mid y_{<t}, \mathbf{f})$

High-level view of WaveNet:

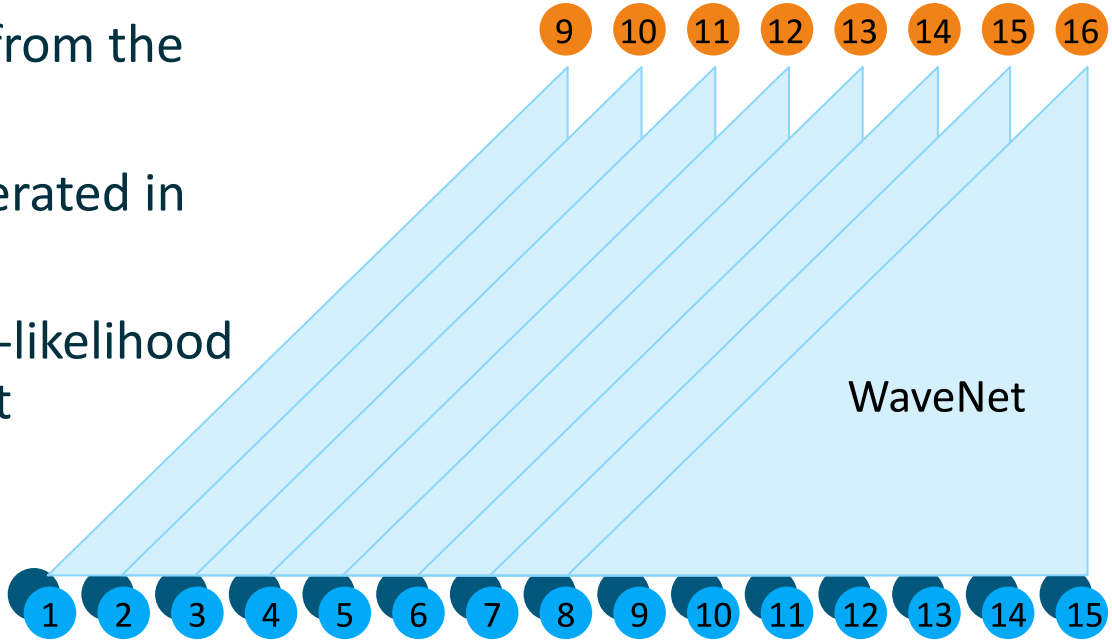
- 1D *causal* convolution over previous audio samples, $y_{<t}$
- with conditional feature(s), **f**, to control generation



WaveNet: Model Training

$p(y_t \mid y_{<t}, f)$

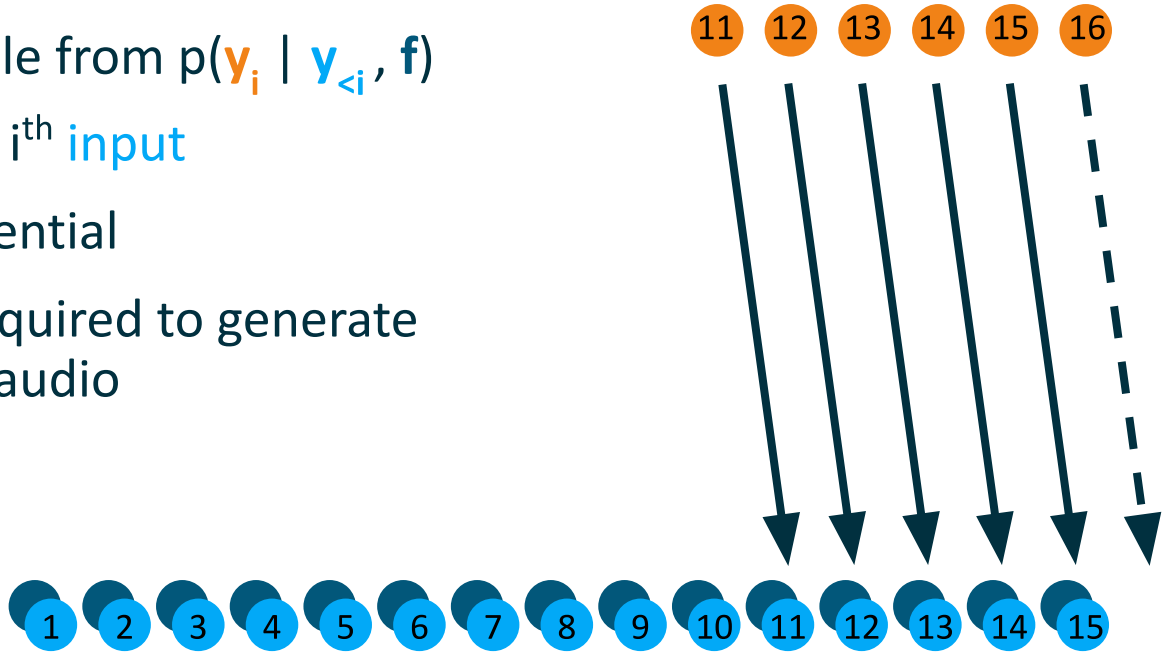
- Input and output are from the same audio clip
- All output can be generated in parallel
- Minimize negative log-likelihood using gradient descent



WaveNet: Model Inference

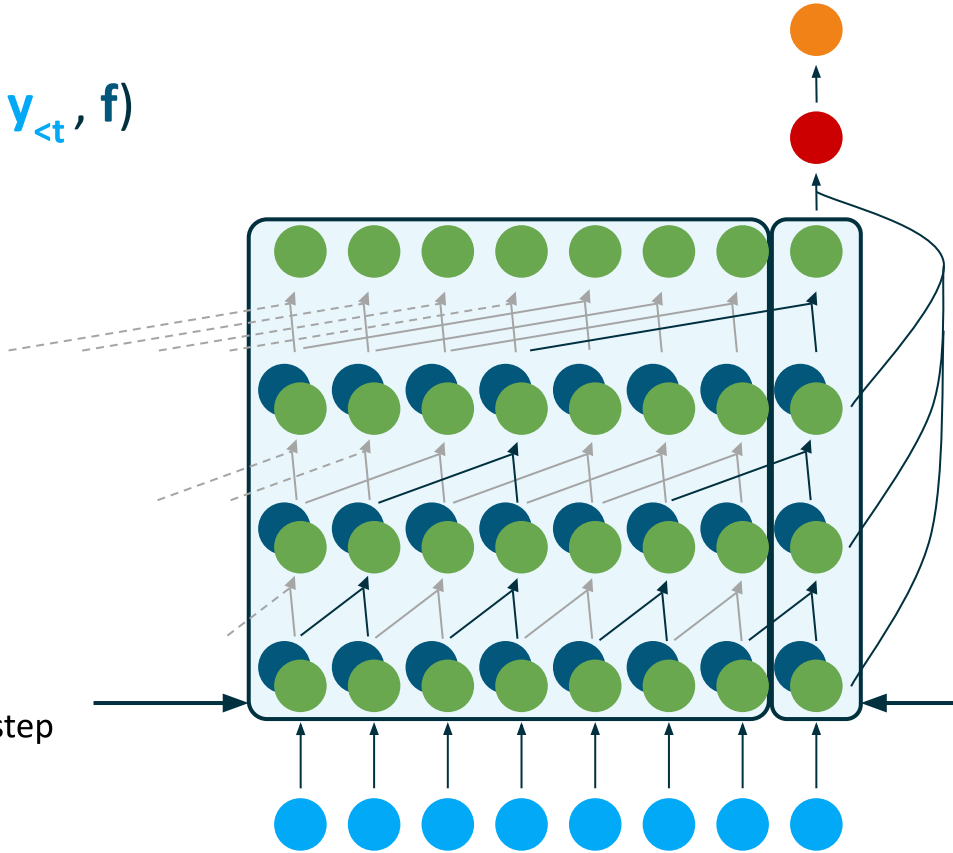
$p(y_t \mid y_{<t}, f)$

- i^{th} output is a sample from $p(y_i \mid y_{<i}, f)$
- i^{th} output becomes i^{th} input
- Generation is sequential
- 16000 iterations required to generate 1 second of 16kHz audio



WaveNet: Model Architecture

$p(y_t \mid y_{<t}, f)$



Read
at current output step

Computed and stored
at current output step

WaveNet: Model Architecture

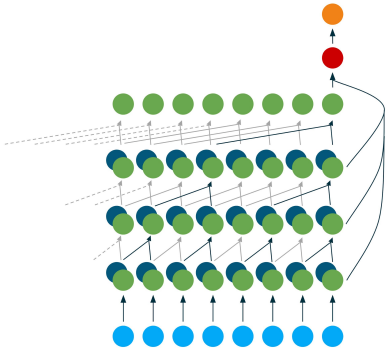
Myrtle.ai

$$p(y_t \mid y_{<t}, f)$$

$$h_0 = \text{Embed}(y_{<t})$$

$$z_k = \text{GatedActivationUnit}(h_{k-1}, f)$$
$$h_k = h_{k-1} + \text{Linear}(z_k)$$

$$o = \Sigma\ \text{Linear}(z_k)$$



$$\text{GatedActivationUnit}(h_k, f) = \tanh(\text{DilatedConv1D}(h_k) + f) \odot \sigma(\text{DilatedConv1D}(h_k) + f)$$

$$p(y_t \mid y_{<t}, f) = (\text{Softmax} \circ \text{Linear} \circ \text{ReLU} \circ \text{Linear} \circ \text{ReLU})(o)$$

WaveNet: Key Characteristics

- Autoregressive: sequential generation

- Model backbone: deep stack of dilated convolutions

Quality Metric: Mean Opinion Score

Myrtle.ai

How natural (i.e. human-sounding) is this recording?

RATING	LABEL	DESCRIPTION
1	Bad	Completely unnatural speech
2	Poor	Mostly unnatural speech
3	Fair	Equally natural and unnatural speech
4	Good	Mostly natural speech
5	Excellent	Completely natural speech

Quality Dials?

Model
Depth

Layer
Dimensions

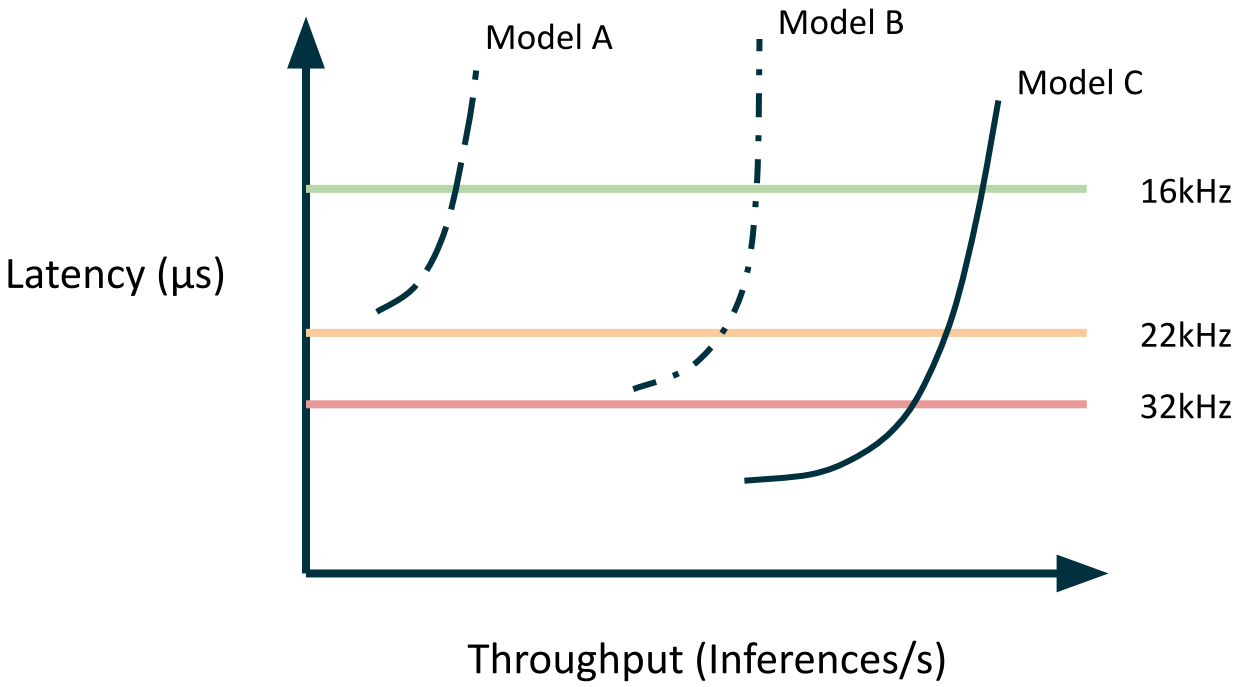
Audio
Frequency

Myrtle.ai

Hardware

Intel® Stratix® 10 NX FPGA

Performance Metric: Latency-bound Throughput



Hard latency constraints for the single forward pass of a model in order to generate X kHz audio in real-time

WaveNet on Existing Accelerators?

$$z_k = \text{GatedActivationUnit}(h_{k-1}, f)$$
$$h_k = h_{k-1} + \text{Linear}(z_k)$$

Intel® Stratix® 10 NX FPGA

Myrtle.ai

- FPGA Architecture
  - Adapt to rapidly changing model architectures and components
- HBM2
  - Meet neural network parameter and activation memory bandwidth requirements
- AI Tensor Blocks
  - Meet neural network compute requirements (OPS)

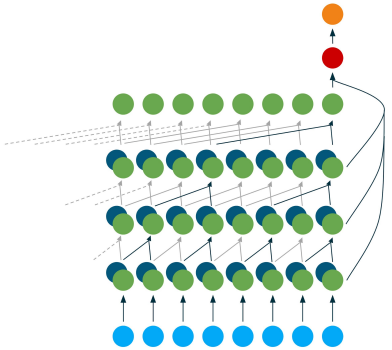
FPGA Architecture: Efficient Sampling

Myrtle.ai

$\mathbf{l} = (\text{Linear} \circ \text{ReLU} \circ \text{Linear} \circ \text{ReLU})(\mathbf{o})$

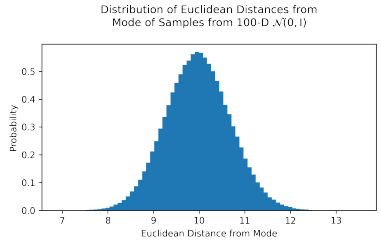
$p(y_t \mid y_{<t}, \mathbf{f}) = \text{Softmax}(\mathbf{l})$

$y_t = \text{sample}(\ p(y_t \mid y_{<t}, \mathbf{f})\ )$



- $\text{Softmax}(\mathbf{l})$ and $\text{sample}(\ p(y_t \mid y_{<t}, \mathbf{f})\ )$ is expensive (logic/area, cycles)

- Alternative: Always “sample” mode of $p(y_t \mid y_{<t}, \mathbf{f})$?
  - Cheap: select index of largest value in $\mathbf{l}$
  - Problem: Quality metric, MOS, is poor
  - Why? Typicality!



FPGA Architecture: Efficient Sampling

Myrtle.ai

$$\mathbf{l} = (\text{Linear} \circ \text{ReLU} \circ \text{Linear} \circ \text{ReLU})(\mathbf{o})$$

$$p(y_t \mid y_{<t}, \mathbf{f}) = \text{Softmax}(\mathbf{l})$$

$$y_t = \text{sample}(\, p(y_t \mid y_{<t}, \mathbf{f}) \,)$$

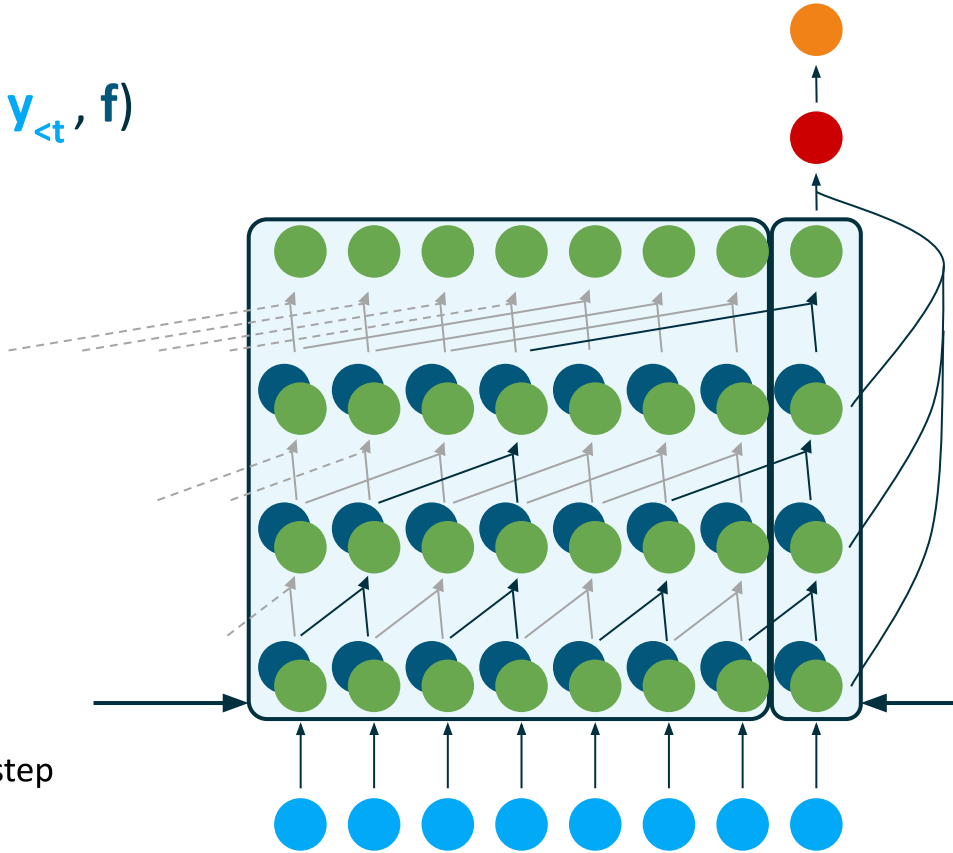
- Must `sample`, but how? Gumbel-max trick!

$$y_t = \text{sample}(\, p(y_t \mid y_{<t}, \mathbf{f}) \,)$$
$$= \text{argmax}(\, \mathbf{l} + \mathbf{g} \,)$$

$$\mathbf{g} \sim \text{Gumbel}(0, 1)$$

HBM2

$p(y_t \mid y_{<t}, f)$

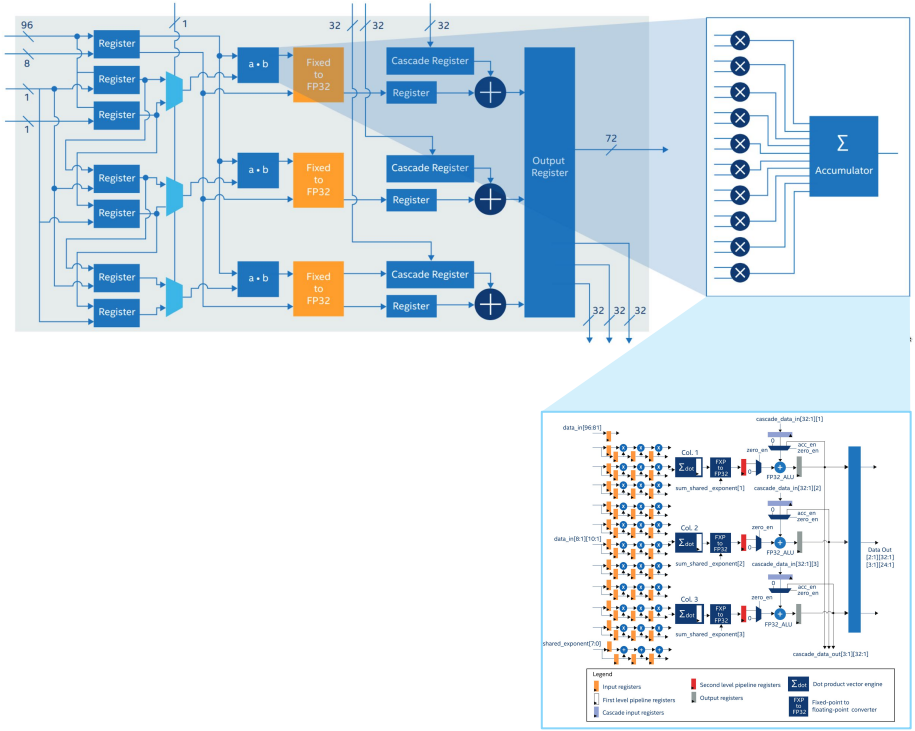


Read
from HBM
at current output step

Computed and stored
to HBM
at current output step

AI Tensor Blocks

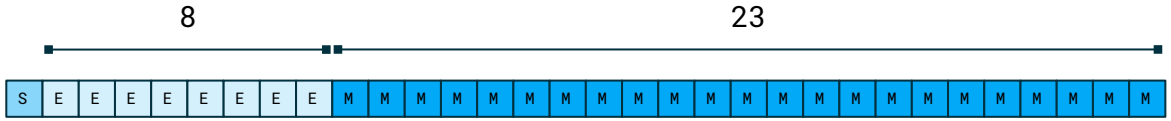
PRECISION	PERFORMANCE	EFFICIENCY
INT4	286 TOPS	2 TOPS/W
INT8	143 TOPS	1 TOPS/W
Block FP12	286 TFLOPS	2 TFLOPS/W
Block FP 16	143 TFLOPS	1 TFLOPS/W
@600 MHz Max Frequency		



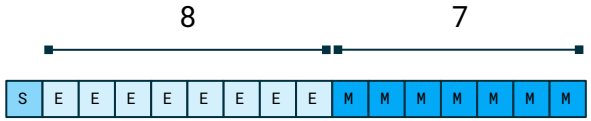
AI Tensor Blocks: Block Floating Point

Myrtle.ai

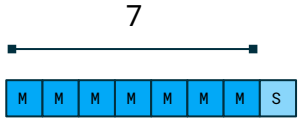
Single-precision Floating Point (FP32)



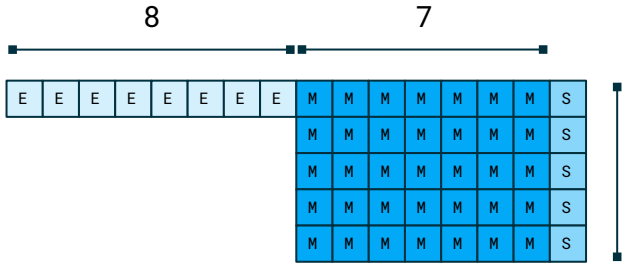
Brain Floating Point (bfloat16)



8-bit Integer (INT8)



Block Floating Point (BFP16)
or
“Microsoft Floating Point” (MSFP16)



Myrtle.ai

Software

Myrtle.ai MAU Accelerator

WaveNet: Performance?

Layer	Type	# Parameters		GOP/second audio	
		Per-layer	Total	Per-layer	Total
Pre-processing Layers					
Embedding	Embedding		30,720		-
Repeated Layers					
Dilation	Dilated Conv1d	57,840	925,440	1.84	29.49
Conditional	Conv1d	19,440	311,040	0.61	9.83
Residual	Conv1d	14,520	217,800	0.46	6.91
Skip	Conv1d	29,040	464,640	0.92	14.75
Post-processing Layers					
Out	Conv1d		61,440		1.96
End	Conv1d		65,536		2.09
Total			2,076,616		65.03

FPGA Accessibility

Myrtle.ai

- Tooling and ecosystem limits usability and accessibility of FPGAs for ML and DL

- Deploying non-standard models requires deep understanding of full stack (ML -> FPGA/HW)

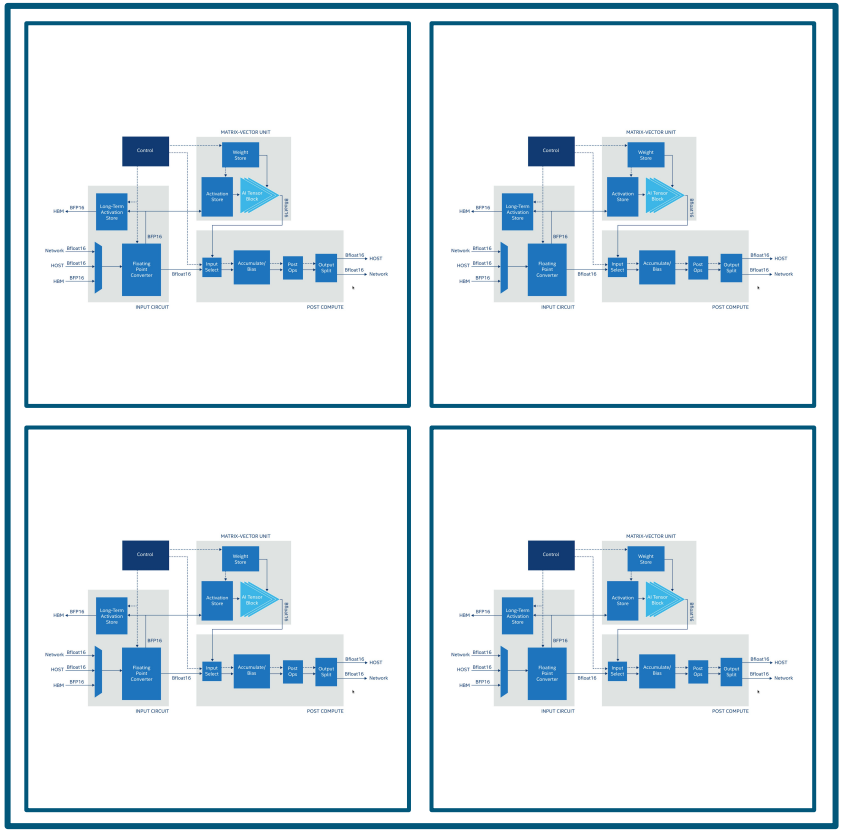
- Driving FPGA adoption for ML and DL requires improved platform, ecosystem, and tooling.

Myrtle.ai MAU Accelerator

- Myrtle.ai MAU Accelerator technologies and proven design techniques:
  - Deliver the right range of abstractions to all users across the stack
  - Enable Myrtle.ai to engage with partners to achieve 10x latency-bounded throughput on a wide range of workloads

Myrtle.ai MAU Accelerator: Architecture

- Configurable number of MAU Cores
- MAU Cores reconfigurable to make optimal use of target platform
- Abstraction reduces complexity of programming device and provides abstraction for ML engineers



Quantization - Quality Evaluation?

Myrtle.ai

Last Layer Quality Impact

$l = (\text{Linear2} \circ \text{ReLU} \circ \text{Linear1} \circ \text{ReLU})(o)$

$p(y_t \mid y_{<t}, f) = \text{Softmax}(l)$

QAT	Linear2	Linear1	WaveNet MOS
No	FP32	FP32	3.930 ± 0.028
No	MSFP16	MSFP16	3.711 ± 0.029
Yes	MSFP16	MSFP16	3.823 ± 0.028
No	FP32	MSFP16	3.902 ± 0.027

Post-training Quantization

Category	Model	Float32	MSFP16
CNNs	Resnet-50	1.000 (75.26)	1.000
	Resnet-101	1.000 (76.21)	1.000
	Resnet-152	1.000 (76.58)	1.000
	Inception-v3	1.000 (77.98)	1.000
	Inception-v4	1.000 (80.18)	1.000
	MobileNet-V1	1.000 (70.90)	0.998
	VGG16	1.000 (70.93)	1.000
	VGG19	1.000 (71.02)	1.000
	EfficientNet-S	1.000 (77.61)	1.000
	EfficientNet-M	1.000 (78.98)	1.000
	EfficientNet-L	1.000 (80.47)	1.000
RNNs	Production-DR	1.000 (76.10)	1.000
	Production-DS	1.000 (73.10)	1.000
Transformers	BERT-MRPC	1.000 (88.39)	1.000
	BERT-SQuAD1.1	1.000 (88.45)	1.000
	BERT-SQuADv2	1.000 (77.23)	1.000

Part of Table 2 from: Darvish Rouhani, Bita, et al. "Pushing the Limits of Narrow Precision Inferencing at Cloud Scale with Microsoft Floating Point." *Advances in Neural Information Processing Systems* 33 (2020)

Myrtle.ai

Solution

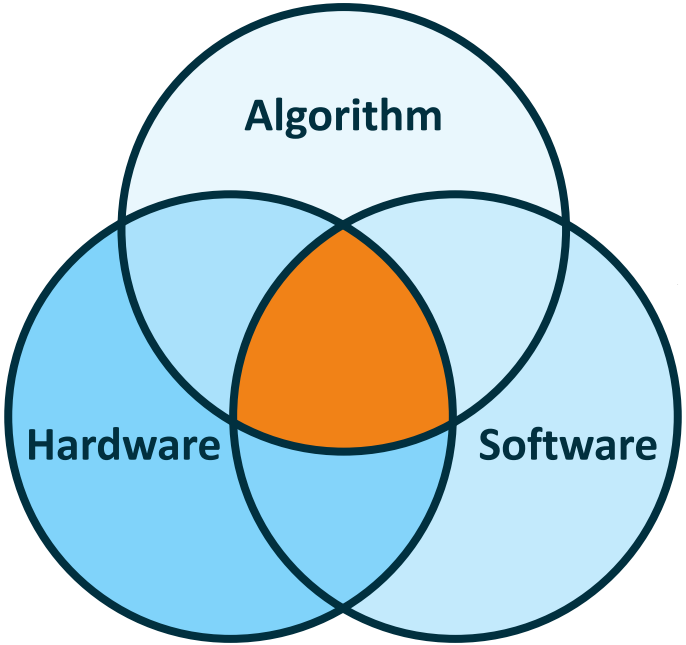
Speech Synthesis at Scale

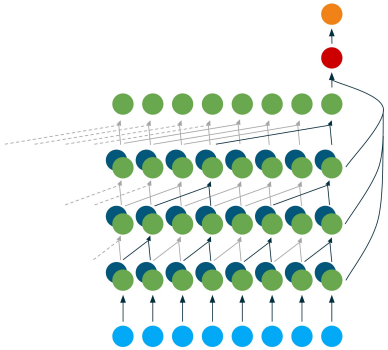
using WaveNet

on Intel® Stratix® 10 NX FPGA

enabled by Myrtle.ai MAU Accelerator

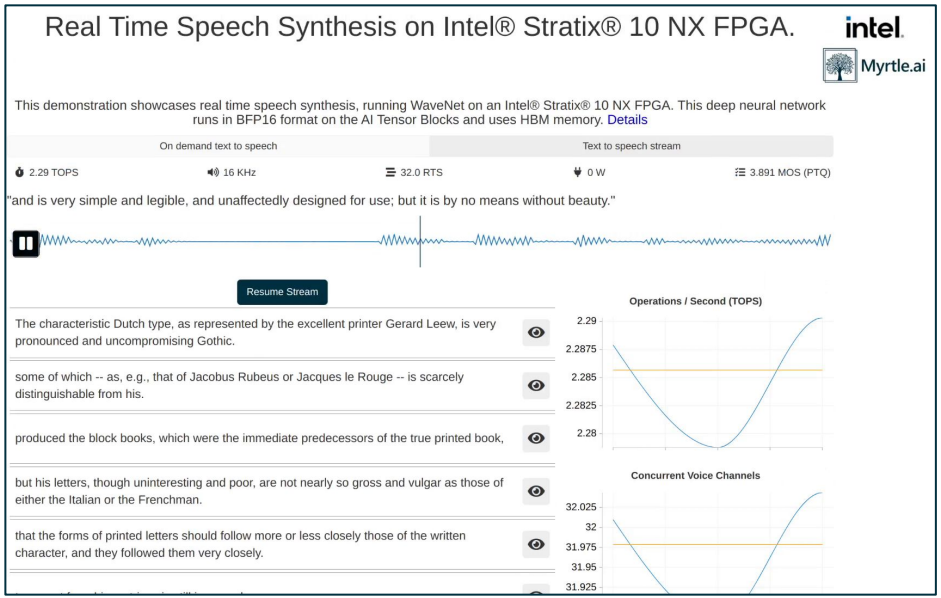
Powering the Future





MAU

Solution Performance



AUDIO FREQUENCY	CONCURRENT VOICE CHANNELS	
	MYRTLE.AI WAVENET	NV-WAVENET
16 kHz	256	32
24 kHz	160	16
32 kHz	128	8

More Information

Myrtle.ai

- Solution content:
  - Intel Blog
  - Whitepaper
  - Demo Video

- Interactive solution website

Myrtle.ai

Thank You

www.myrtle.ai


Sam Davis


+44 1223 967248


sam@myrtle.ai