# Part 1: ASR - Automatic Speech Recognition

Building Conversational AI Applications

# About Me

Instructor Name – Instructor Contact (e.g. LinkedIn / email)

- About instructor 1
- About instructor 2
- About instructor 3

# Course Timeline

| | | | |
|---|---|---|---|
| 8:50 | — | 9:00 | Welcome |
| 9:00 | — | 10:00 | Lecture 1 |
| 10:00 | — | 11:00 | Lab 1 |
| 11:00 | — | 11:15 | Break |
| 11:15 | — | 12:15 | Lecture 2 |
| 12:15 | — | 13:15 | Lab 2 |
| 13:15 | — | 14:00 | break |
| 14:00 | — | 15:00 | Lecture 3 |
| 15:00 | — | 16:00 | Lab 3 |
| 16:00 | — | 17:00 | Assessment support |

# Course Agenda

## Part 1: ASR - Automatic Speech Recognition

Lecture: Challenges of conversational AI applications,  building blocks, ASR (automatic speech recognition) focus

Lab: Investigate ASR pipeline with NeMo and Riva

## Part 2: TTS – Text To Speech and Conversational AI Customization

Lecture: How can we take advantage of conversational AI platforms to build custom applications?

Lab: Investigate the TTS pipeline with NeMo and Riva; then build a full custom conversational AI pipeline.

## Part 3: Production Deployment

Lecture: Discussion of production deployment considerations including an overview of NVIDIA Triton Inference Server and TensorRT.

Lab: Hands-on deployment of an example application using Helm and Kubernetes; includes profiling the application


# ASR (Part 1)

Lecture

- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

Lab

- Investigate ASR Pipeline with NeMo and Riva


Shannon's

# Automatic Speech Recognition Is Everywhere!

Hundreds of billions of minutes of speech generated daily

## Call Center

500M Calls Daily


## Consumer Apps

1.8B Daily Minutes

## Online Meetings

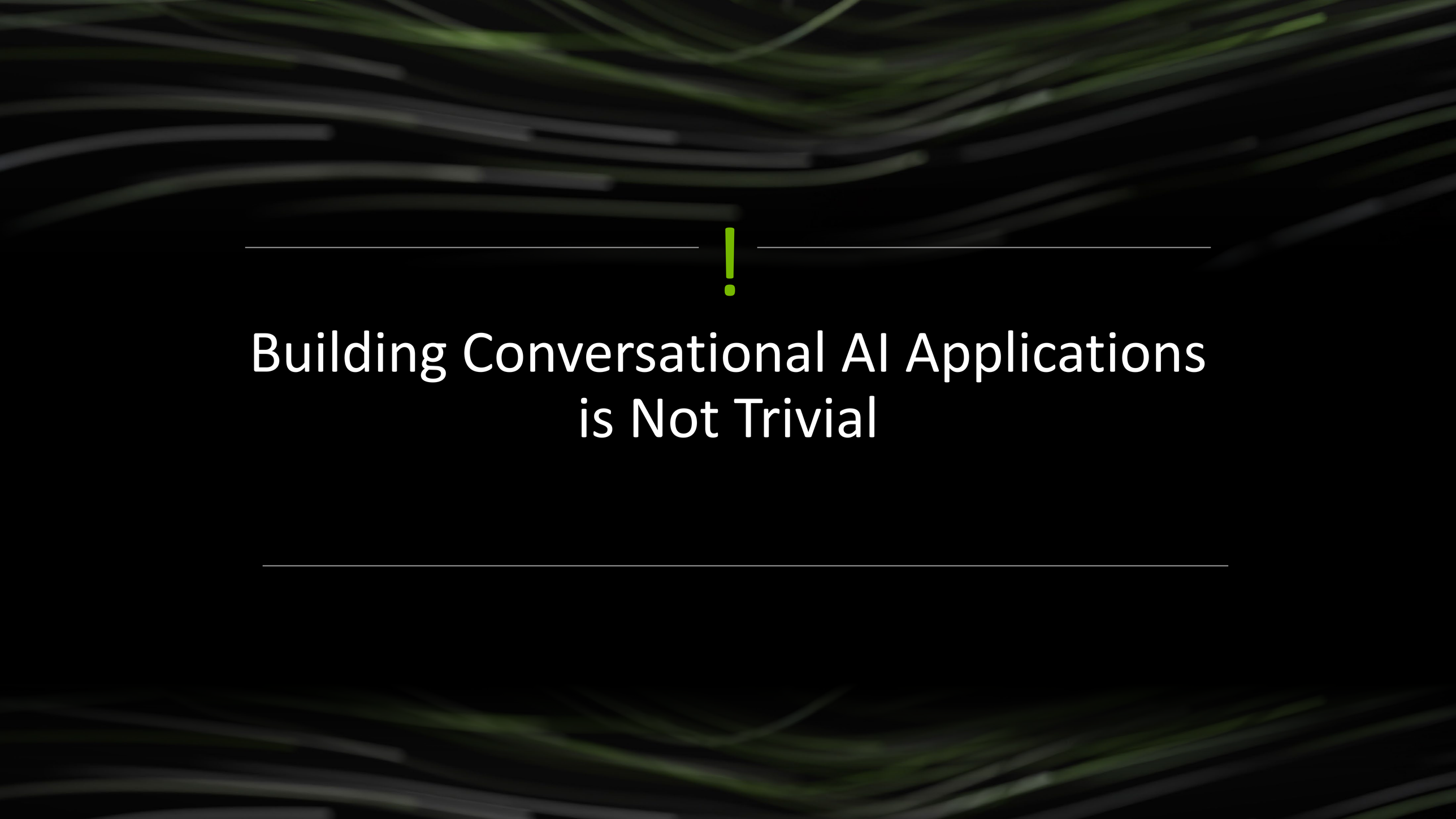
200M Daily

Telecom

Finance

Healthcare

Retail


!
Building Conversational AI Applications
is Not Trivial

# Challenges

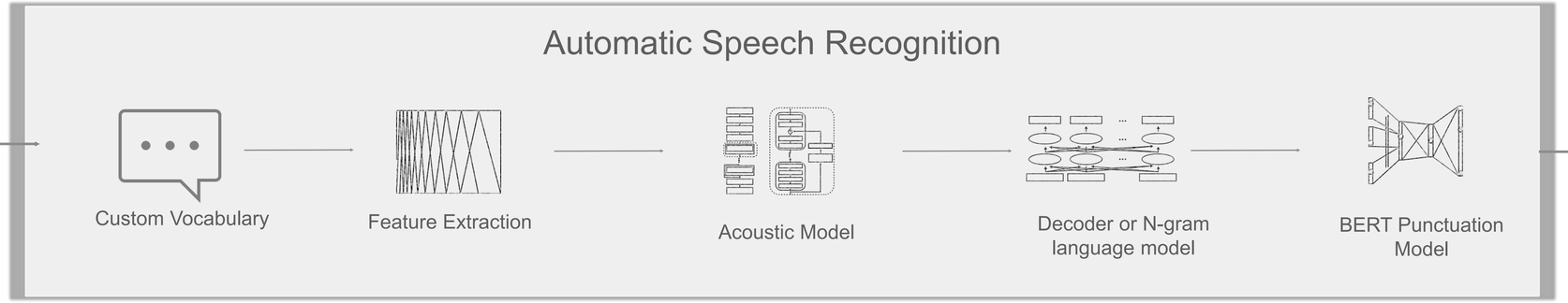
New models , domain-specific accuracy, complex pipelines, real-time responses

| HIGH ACCURACY | REAL-TIME RESULTS | SCALABLE DEPLOYMENT | CUSTOMIZATION | DATA SOVEREIGNTY |
|---|---|---|---|---|
| • Lack of access to state-of-the-art speech algorithms and models | • Require low-latency for natural interactions | • Limited support for deployment platforms<br>• Limited support for scaling to multiple client requests | • Limited model, domain, and pipeline customization | • Limited control over the proprietary data used for training and inference |

# Challenges

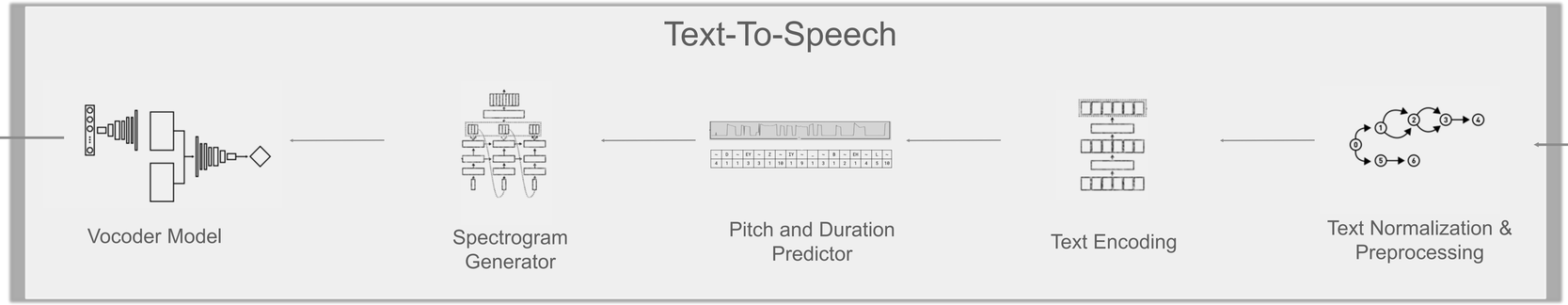
## New models , domain-specific accuracy, complex pipelines, real-time responses

*“What is the weather like tomorrow in London?”*




NLU

Dialog Manager



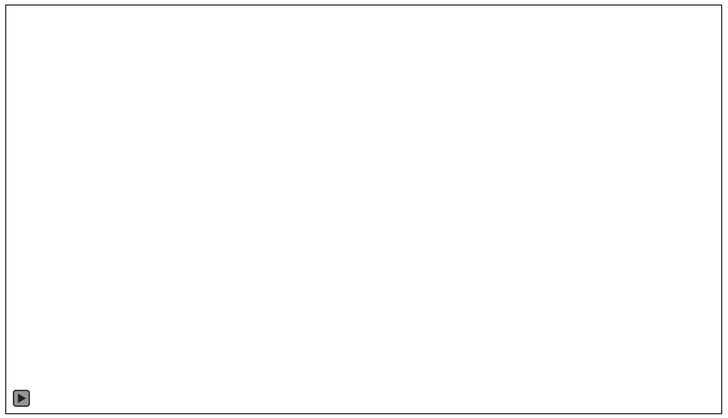

*“Expect some clear skies in London tomorrow.”*

## Real-time applications need to deliver latency <300 ms

# Challenges

New models , domain-specific accuracy, complex pipelines, real-time responses

Intent Classification
Slot Filling
Named Entity Recognition
...



*How many CUDA cores is that?*

Automatic Speech Recognition

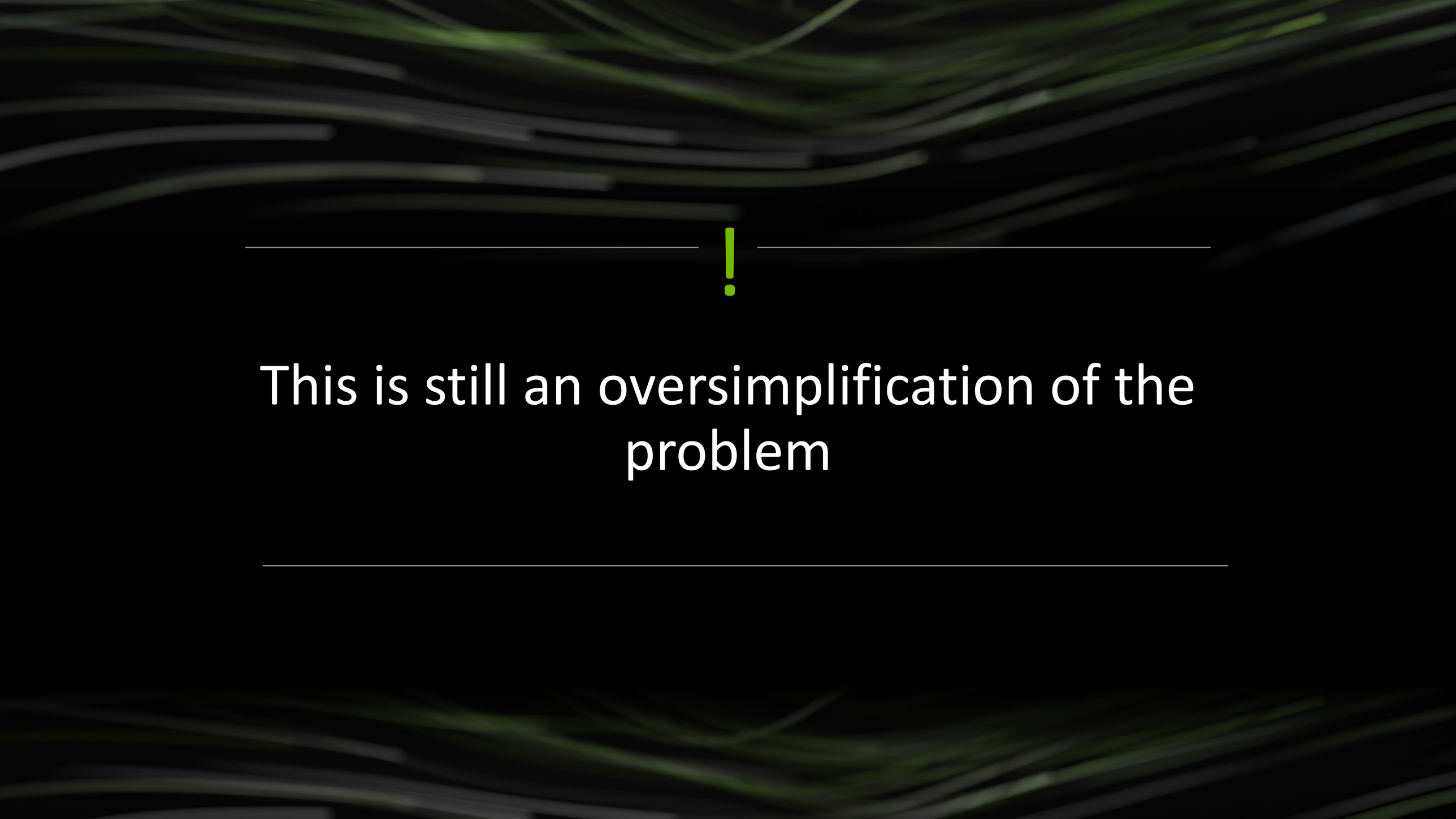
Natural Language Processing

Text to Speech

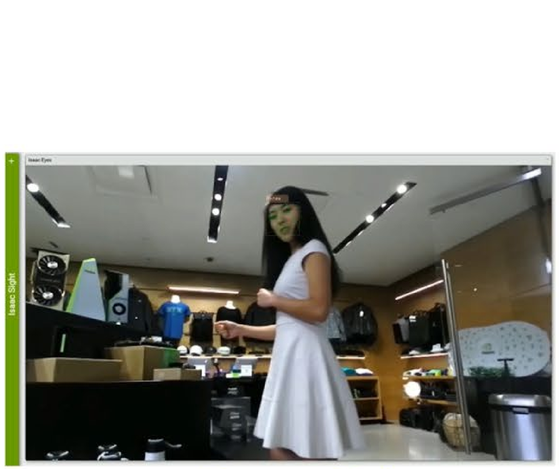
*Jetson Nano is 128 CUDA Cores*

Computer Vision

Digital Avatar


!
This is still an oversimplification of the problem

# What About Additional Logic?



*How many Cuda cores is that?*

Automatic Speech Recognition

Intent Classification
Slot filling
Named Entity Recognition

Natural Language Processing

Text to Speech

*Jetson Nano is 128 Cuda Cores*

Digital Avatar

Computer Vision

*"The quality of the product is amazing. I liked it so much."*

Input Text

*Tokenization*
*Lemmatization*
*Stemming*

Data Preprocessing

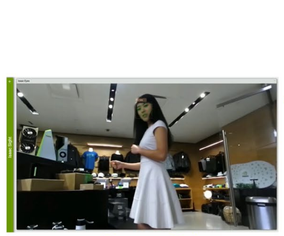
BERT Pre-trained Model

Additional-Layers
(Sentiment Analysis)

*Positive: 98%*
*Negative: 2%*

Output

# Not Something Unique to NLP



How many Cuda cores is that?

Automatic Speech Recognition

Intent Classification
Slot filling
Named Entity Recognition
...

Natural Language Processing

Text to Speech

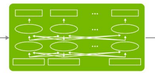
Jetson Nano is 128 Cuda Cores

Digital Avatar

Computer Vision

"The quality of the product is amazing. I liked it so much."

Tokenization
Lemmatization
Stemming

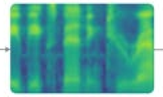


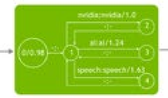
Positive: 98%
Negative: 2%

Input Text | Data Preprocessing | BERT Pre-trained Model | Additional-Layers (Sentiment Analysis) | Output

"Hello"

Audio Feature Extractor and Preprocessor





"Hello"

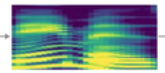
Audio Input | Spectrogram | Neural Acoustic Model | Prob(t, c) | Decoder With Language Model | Output Transcript

Text Analysis | Linguistic Analysis

"Pay $200"

"Pay two hundred dollars"

(Pay, Verb), (Two, Noun), (Hundred, Noun), (Dollars, Noun)



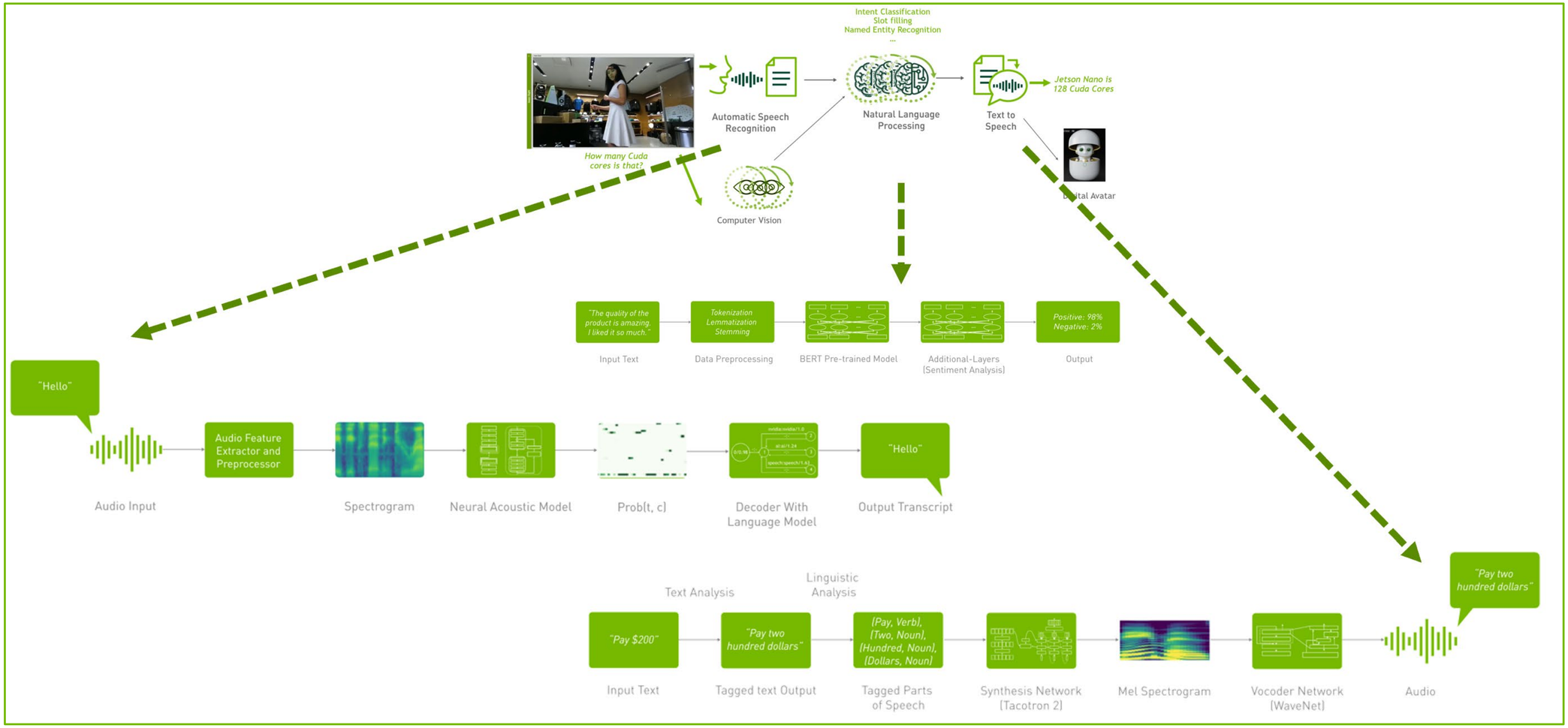
"Pay two hundred dollars"

Input Text | Tagged text Output | Tagged Parts of Speech | Synthesis Network (Tacotron 2) | Mel Spectrogram | Vocoder Network (WaveNet) | Audio

NVIDIA

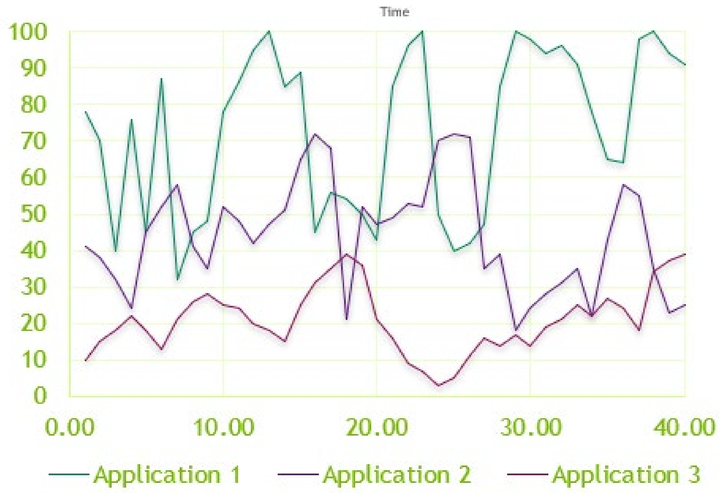
# This Was Just a Single Application

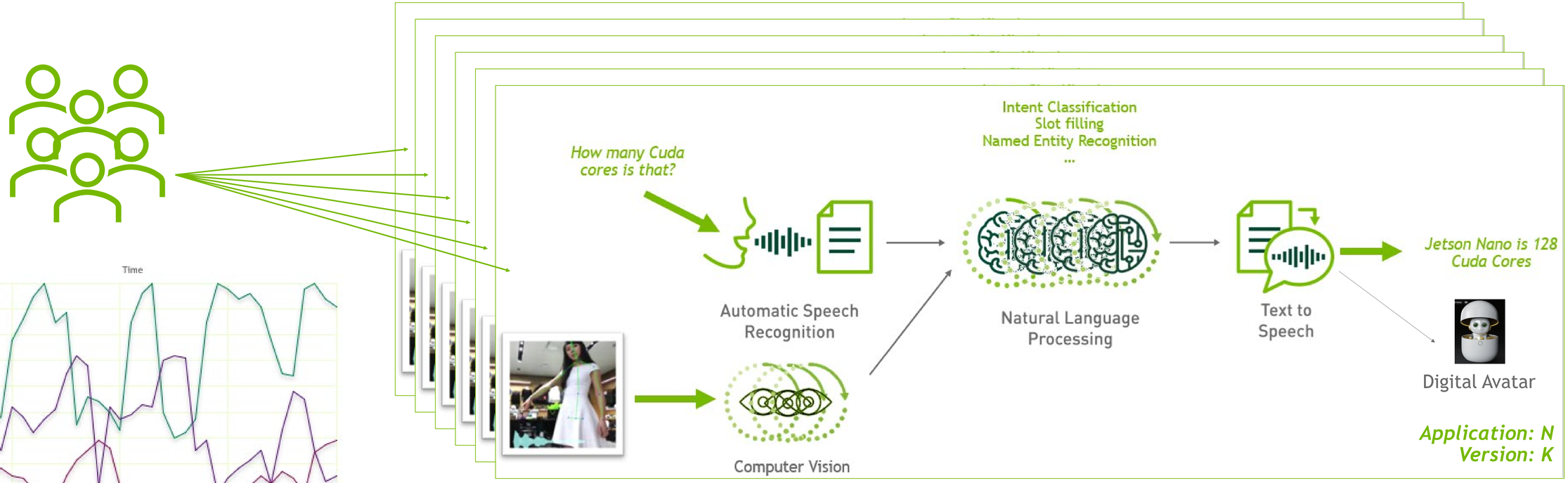
## Application 1



# Challenges

Latency, throughput, utilization, capital investment, ongoing costs




# ASR (Part 1)

## Lecture

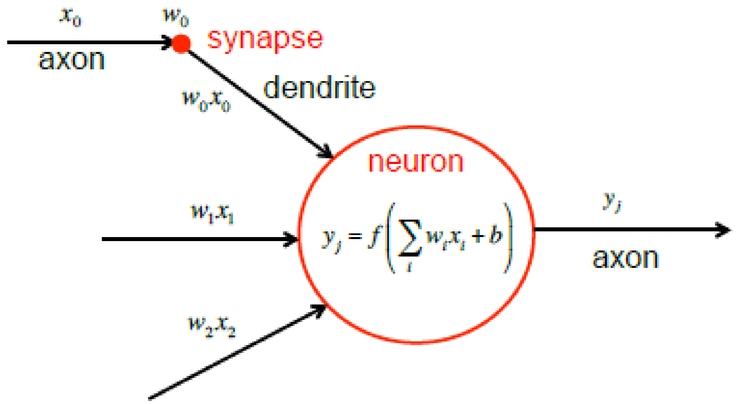
- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

## Lab

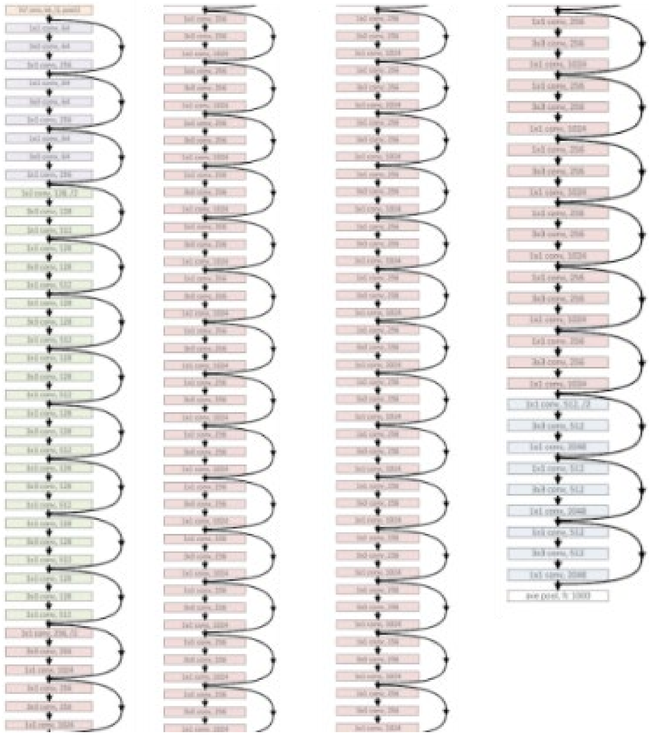
- Investigate ASR Pipeline with NeMo and Riva

# Neural Networks are Not New

They are surprisingly simple as an algorithm



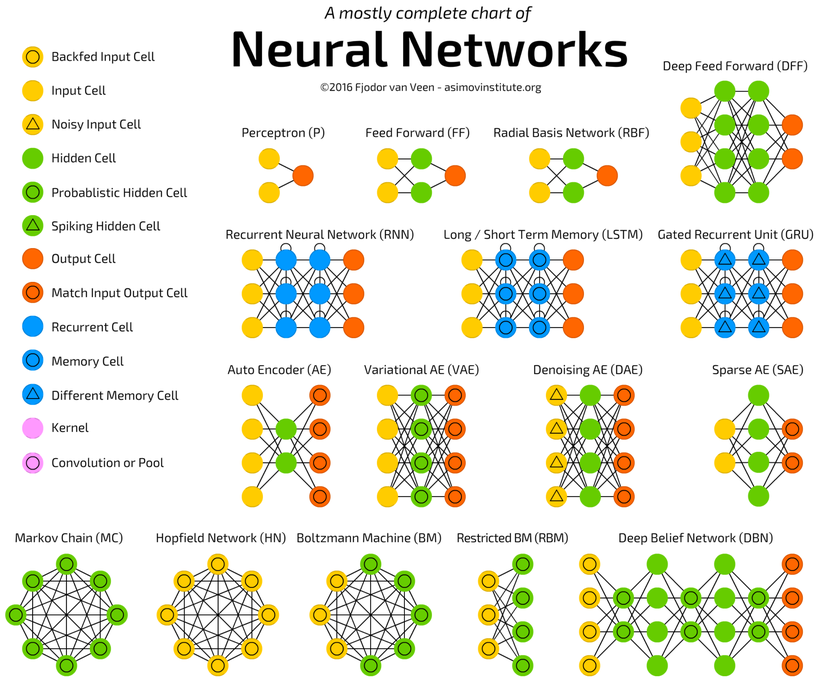

Neural Network - Neuron

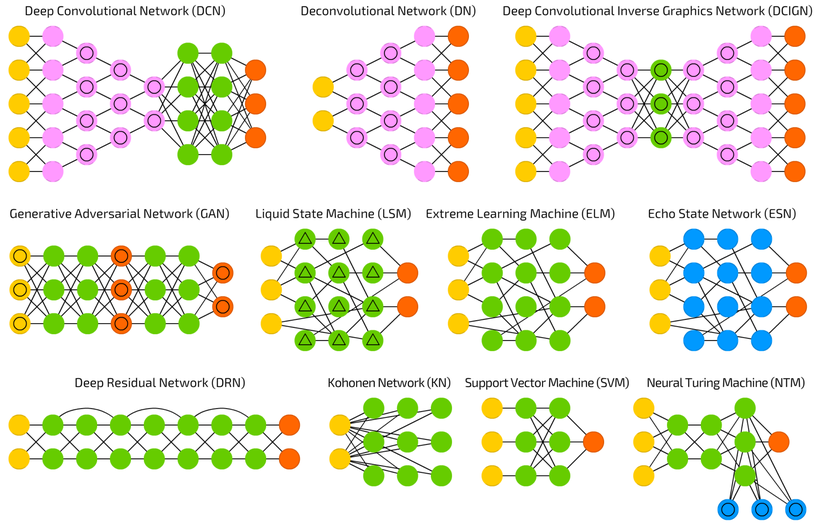


Deep Neural Network

# Neural Networks

## Convolutional and recurrent neural networks

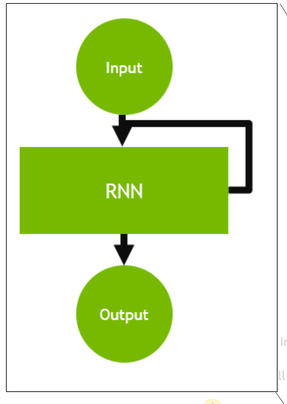


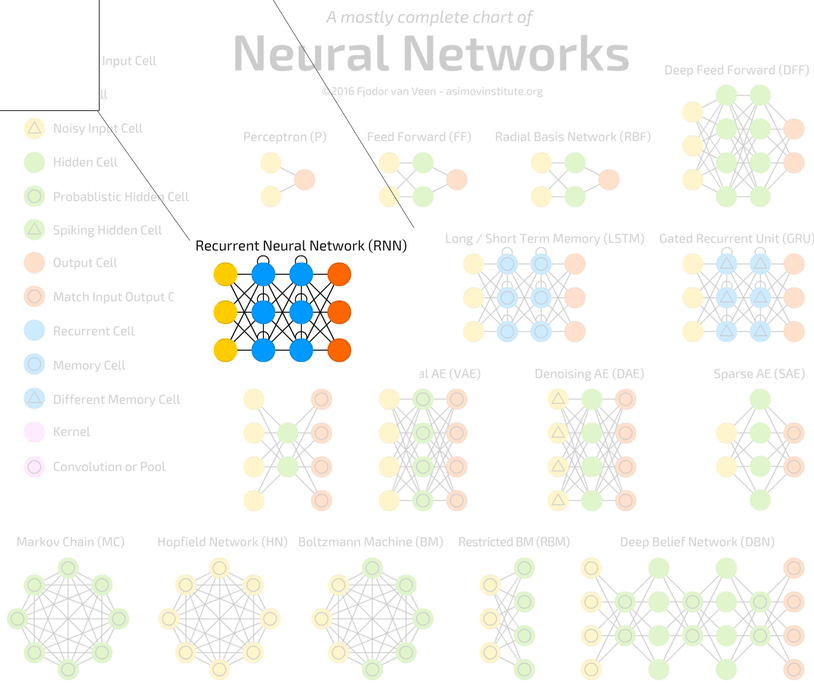



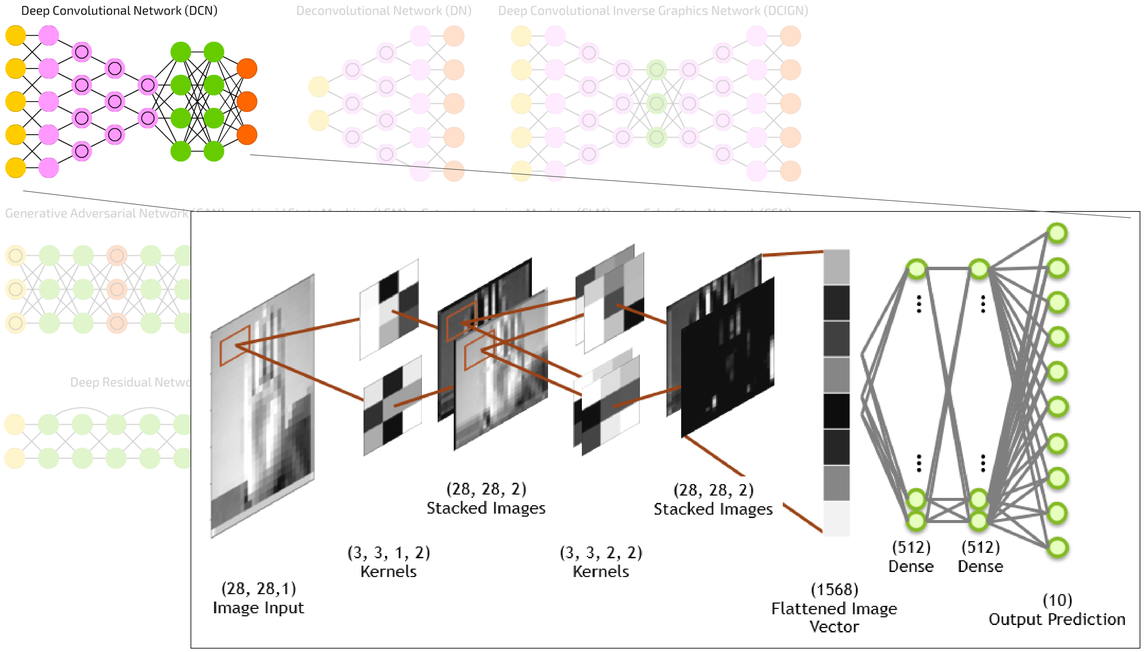

Credit: https://www.asimovinstitute.org/the-neural-network-zoo-2016/

# Neural Networks

## Convolutional and recurrent neural networks



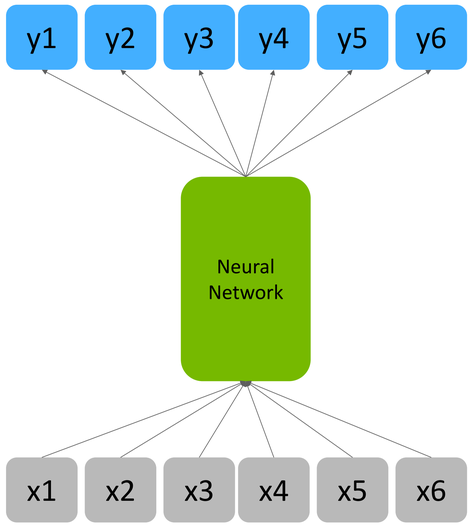




Credit: https://www.asimovinstitute.org/the-neural-network-zoo-2016/

# Neural Networks

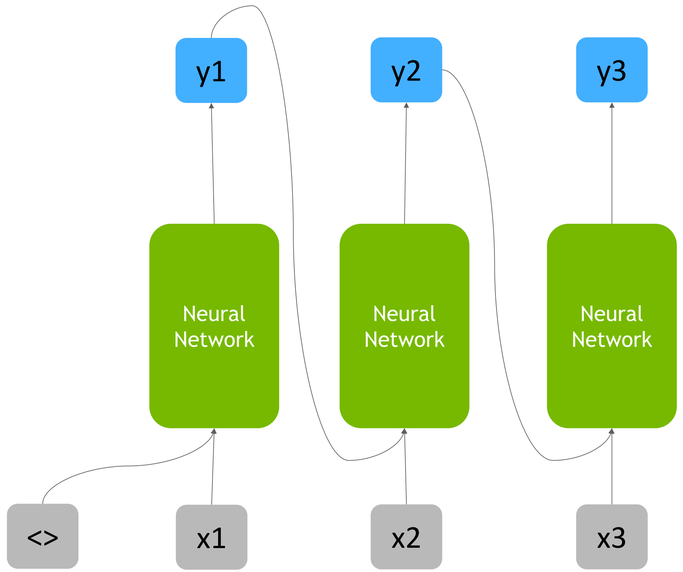
## Autoregressive and non-autoregressive

Non Autoregressive Models




- Predictions of a Non-autoregressive model at the current time step **does not depend** on the predictions of the previous time-steps
- Fast at inference time
- Called also parallel

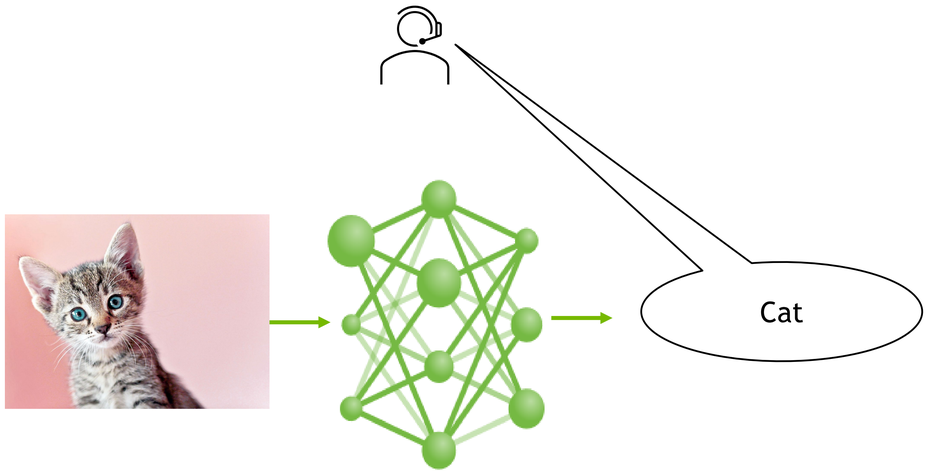
Autoregressive Models



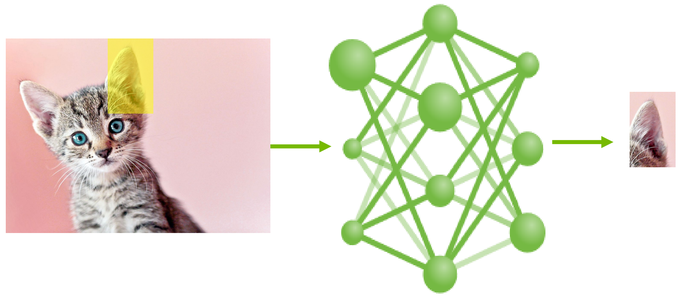

- Predictions of an autoregressive model at the current time step **depends on** the predictions of the previous time-steps
- Usually higher accuracy compared to non-autoregressive models due to temporal dependencies
- Slow at inference time

# Neural Networks

## Supervised, Self-Supervised




Supervised Training

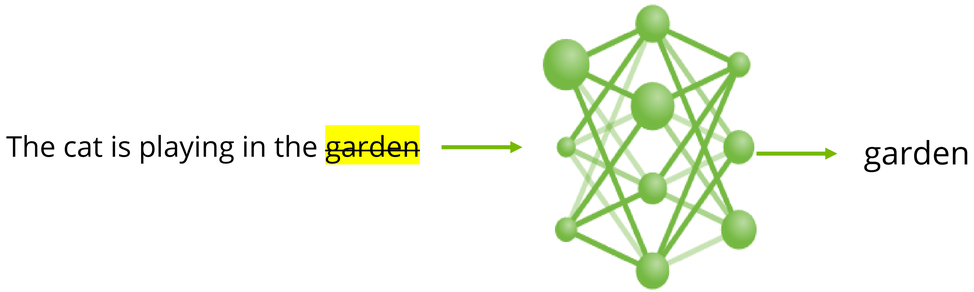


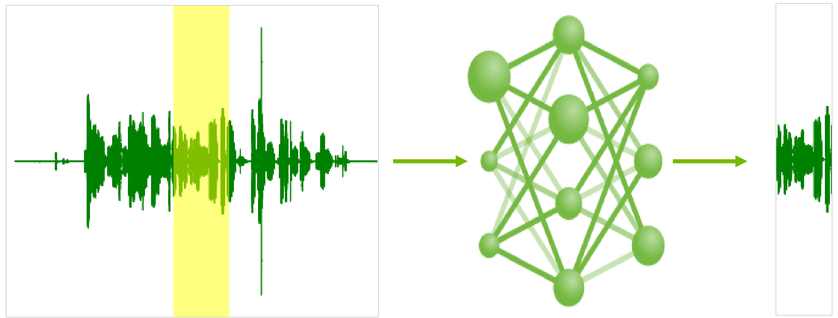
Self-Supervised Training

# Neural Networks

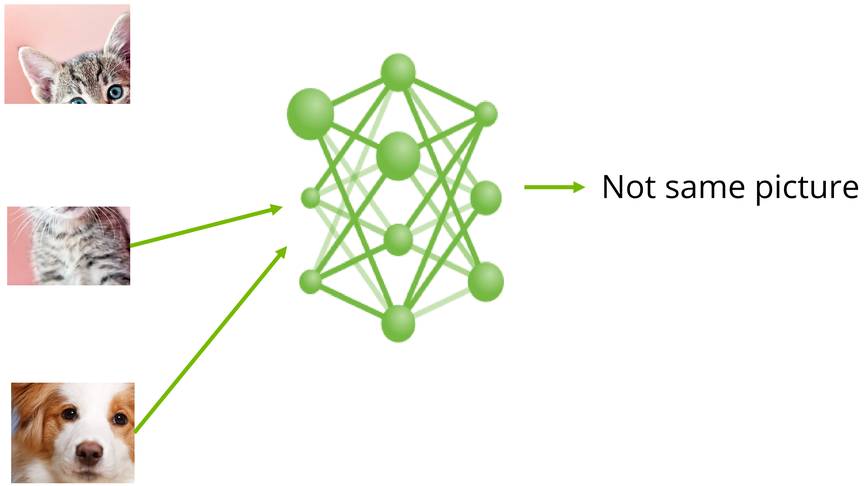
## Self-Supervised

Masking Loss



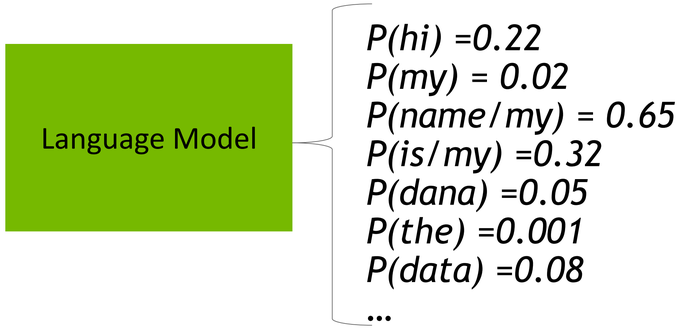



Contrastive Loss



# Neural Language Models

## Language Model




*P(hi my name is dana) ?*

Probability of a sentence is decomposed into conditional probabilities of each word given its previous context:

*P(hi my name is dana) = p(hi) * p(my/hi) * p(name/hi my) * p(is/hi my name) * p(dana/hi my name is)*


# ASR (Part 1)

## Lecture

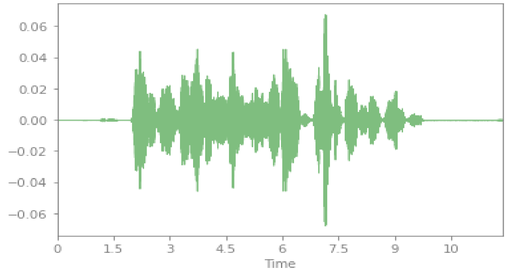
- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

## Lab

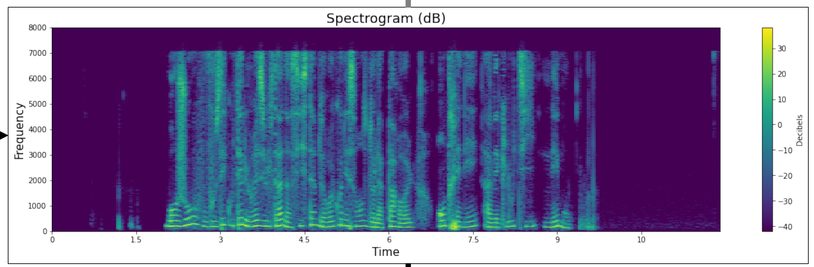
- Investigate ASR Pipeline with NeMo and Riva

# Audio Data

## Features Representation

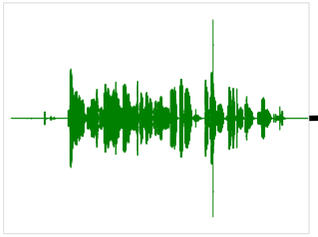



Inverse Discrete Fourier Transform

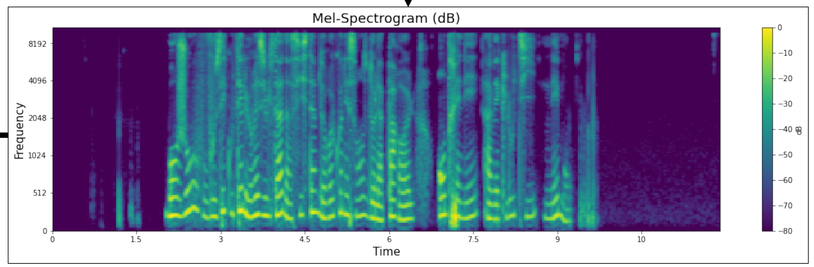



Discrete Fourier Transform

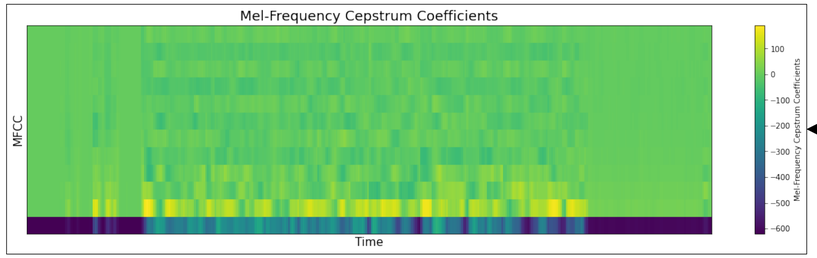
Windowing 20-25ms, stride 10ms



Log Filter Bank




Discrete Cosine Transform



# Text Data

## char/graphemes/words → phonemes/tokens

| Text | Definition | Examples |
|---|---|---|
| Char | Alphabet Characters | a, b, c, d, …n,… |
| Graphemes | Small unit of a written language | n, nn, kn, gn, pn |
| Words | Words | Hello, world, .. |

| Text Representation | Definition | Examples | Standard | Comments |
|---|---|---|---|---|
| Phonemes | Small unit of spoken sound | /n/ | Grapheme to phonemes<br><br>International Phonetic Alphabet (IPA):<br>'H', 'E', 'L', 'L', 'O', ' ', 'ˈ', 'w', 'ɝ', 'l', 'd'<br><br>ARPABET:<br>HH, AH0, L, OW1, ,W, ER1, L, D | Non Phonetic Languages:<br>• Multiple graphemes for a phoneme (eg. "f" "ph")<br>• Different pronunciations for same spelling (e.g. "I read book" in present and past) |
| Tokens | Subwords | _use_ful | Tokenizers: Byte-Pair Encoding, WordPiece or SentencPiece<br><br>bert-base-uncased:<br>'i', 'work', 'for', 'n', '##vid', '##ia'<br>1045, 2147, 2005, 1050, 17258, 2401<br>xlnet-base-cased:<br>'_', 'i', '_work', '_for', '_', 'n', 'vid', 'ia'<br>17, 150, 154, 28, 17, 180, 7791, 780 | • ML Algorithms deal with indexes instead of tokens<br>• Tokenizers uses a fixed size of vocabulary<br>• Out of Vocabulary OOV:<br>  • Chars are part of the vocabulary<br>  • Special token UNK |


# ASR (Part 1)

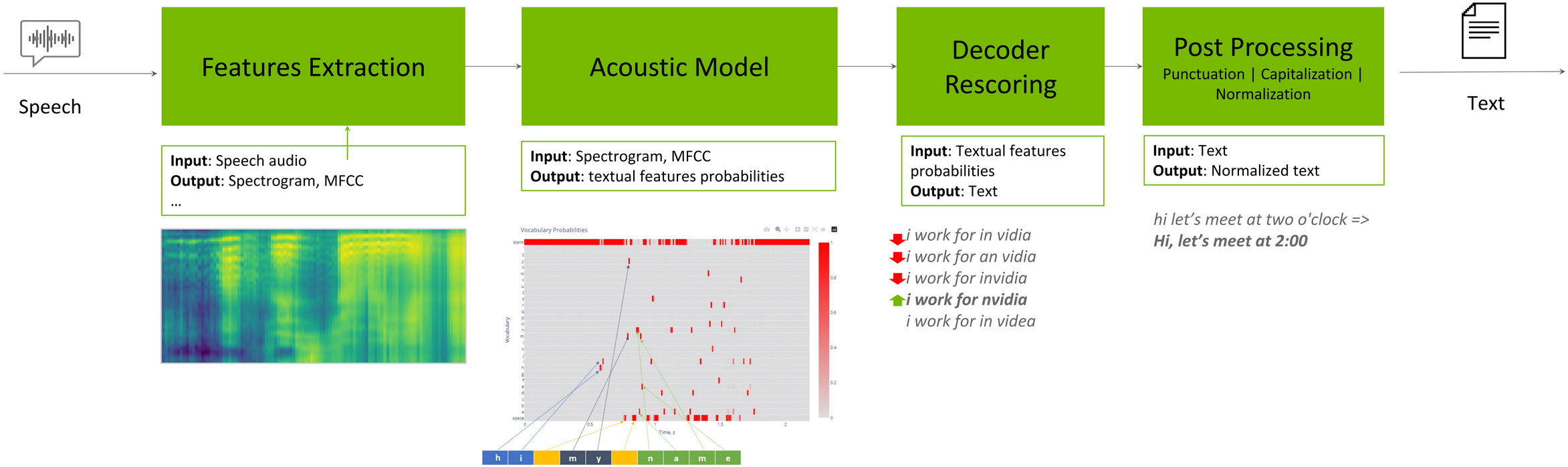
## Lecture

- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

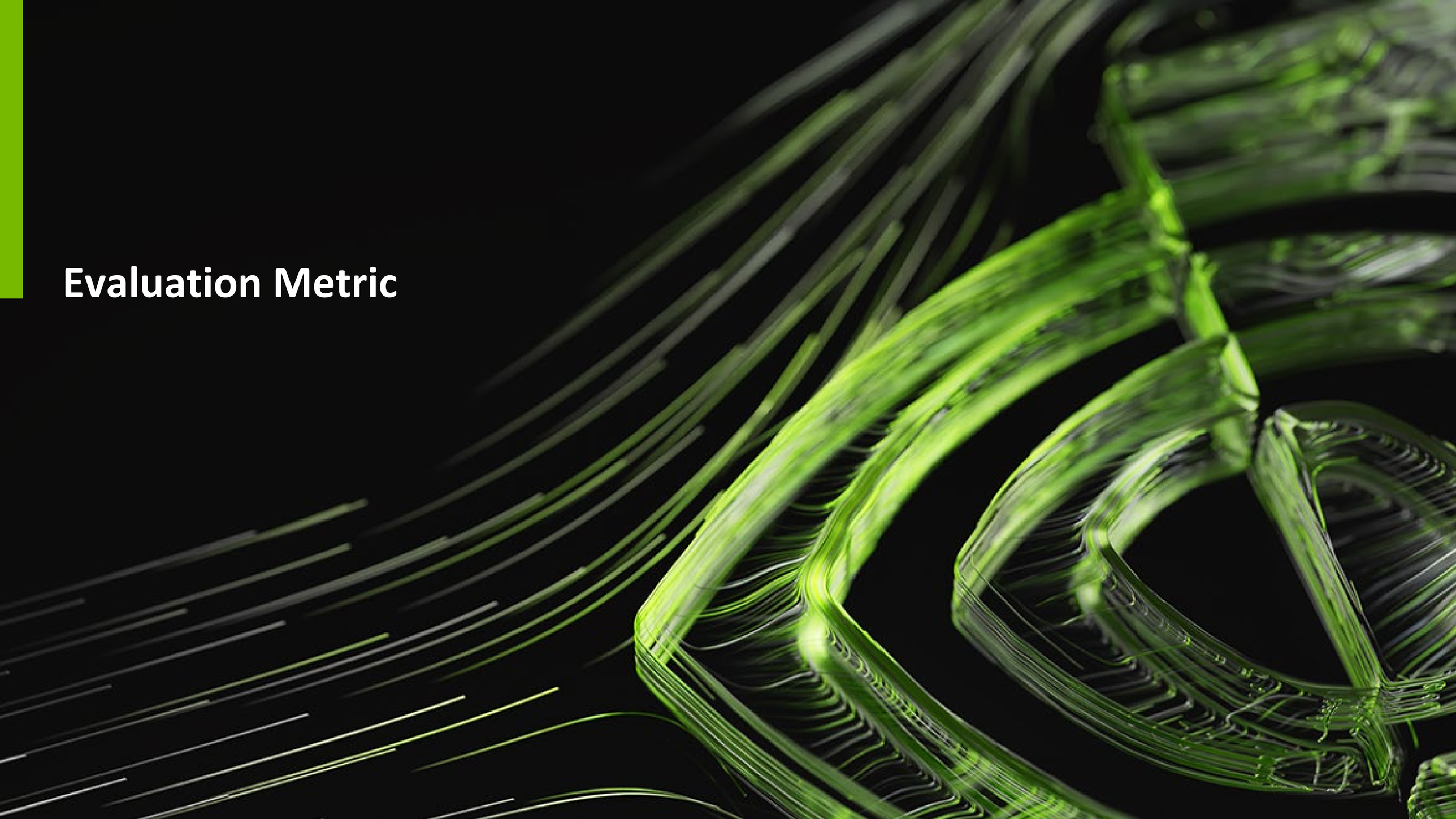
## Lab

- Investigate ASR Pipeline with NeMo and Riva

# ASR Pipeline



# Evaluation Metric



# Measuring Success of ASR

## WER | CER

$$Word\ Error\ Rate\ (WER) = \frac{S + D + I}{N}$$

- *S: Number of words substituted*
- *I: Number of words inserted*
- *D Number of words deleted*
- *N: Number of words in the reference*

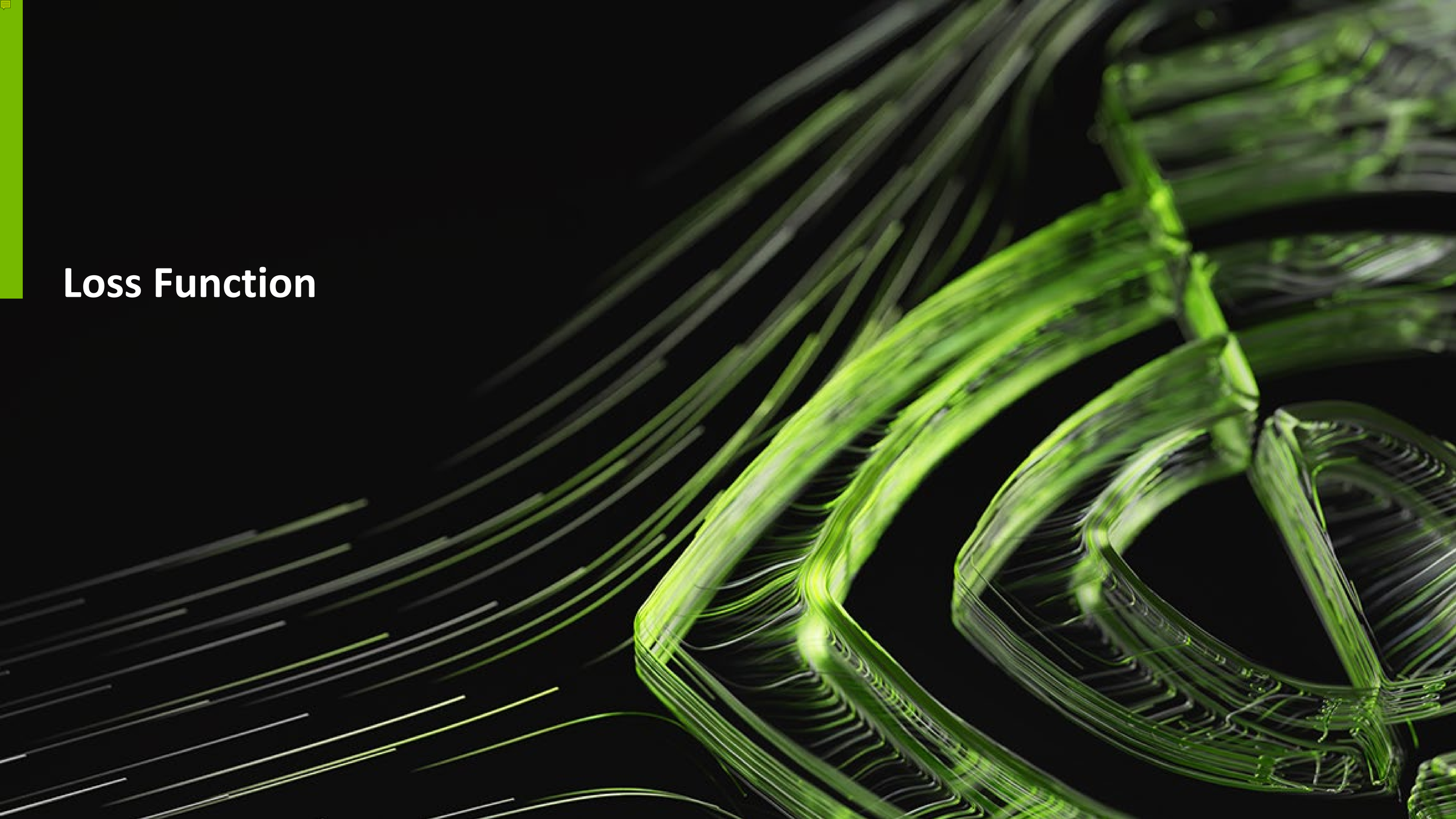
| | |
|---|---|
| Reference | *the cat sat on the mat* |
| Transcription | *the cats at on mat yeah* |
| WER=0.66 | D=1, S=2, I=1, N=6 |

$$Character\ Error\ Rate\ (CER) = \frac{S + D + I}{N}$$

- *S: Number of chars substituted*
- *I: Number of chars inserted*
- *D Number of chars deleted*
- *N: Number of chars in the reference*

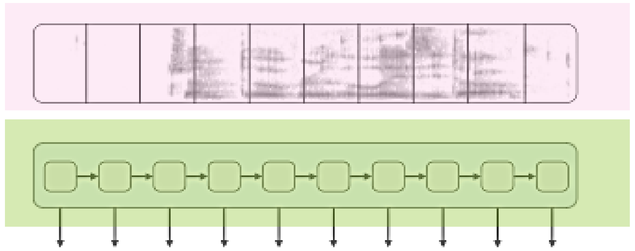
| | |
|---|---|
| Reference | *'t', 'h', 'e', ' ', 'c', 'a', 't', ' ', 's', 'a', 't', ' ', 'o', 'n', ' ', 't', 'h', 'e', ' ', 'm', 'a', 't'* |
| Transcription | *'t', 'h', 'e', ' ', 'c', 'a', 't', ' ', 's', 'a', 't', ' ', 'o', 'n', ' ',  'n', 'a', 't'* |
| CER=0.2272 | D=4, S=1, I=0, N=22 |

# Loss Function



# Acoustic Models

## Connectionist Temporal Classification (CTC)



We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, $\epsilon$} for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

To be precise, the CTC objective for a single $(X, Y)$ pair is:

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

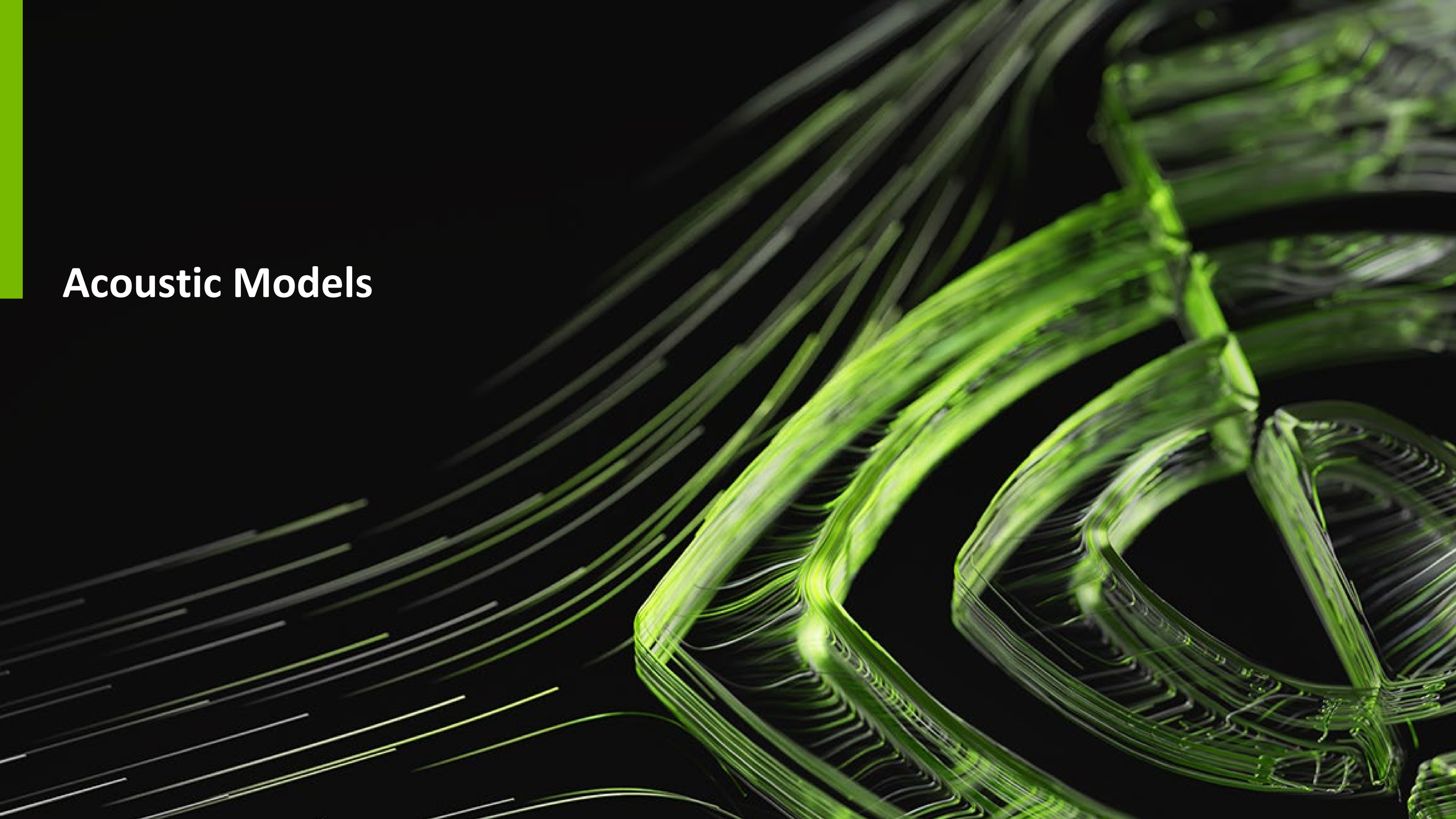
The CTC conditional **probability**

**marginalizes** over the set of valid alignments

computing the **probability** for a single alignment step-by-step.

https://distill.pub/2017/ctc/

# Acoustic Models



# Acoustic Models

## RNN-based approaches: DeepSpeech and DeepSpeech2

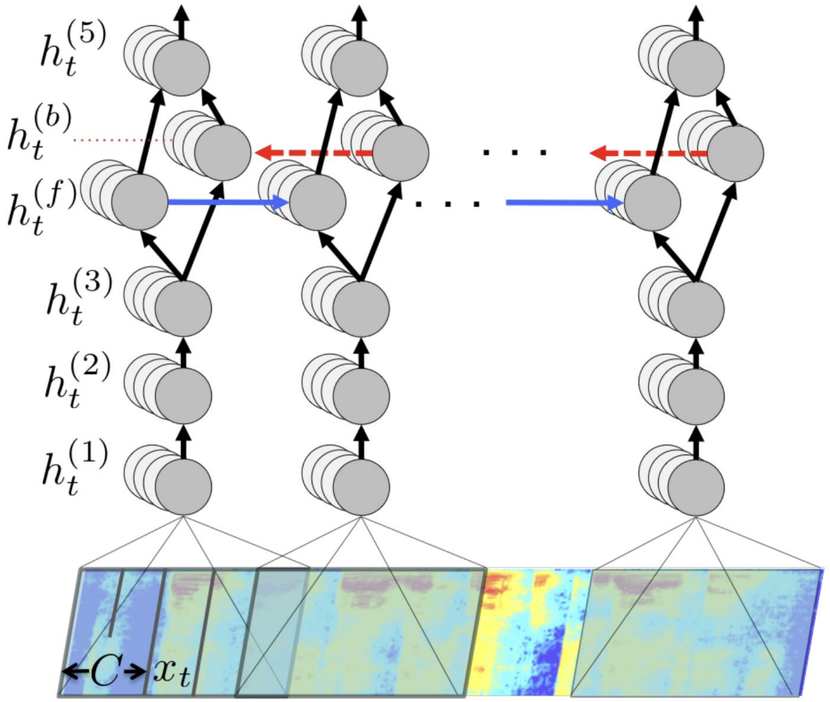



Figure 1: Structure of our RNN model and notation.

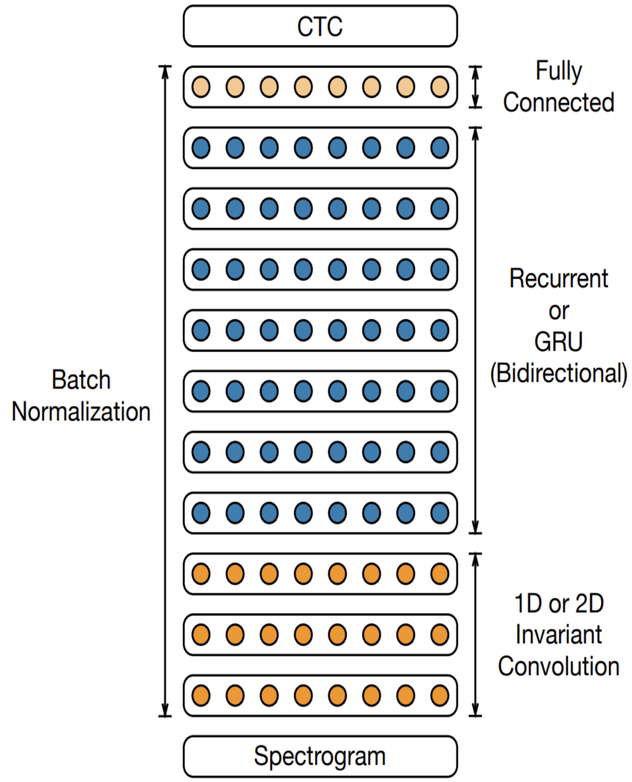



**Figure 1:** Architecture of the DS2 system used to train on both English and Mandarin speech. We explore variants of this architecture by varying the number of convolutional layers from 1 to 3 and the number of recurrent or GRU layers from 1 to 7.

Awni Hannun , Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng .Deep Speech: Scaling up end-to-end speech recognition. 2014

# Acoustic Models

## RNN-based approaches: DeepSpeech and DeepSpeech2

| RNN output | Decoded Transcription |
|---|---|
| what is the weather like in bostin right now | what is the weather like in boston right now |
| prime miniter nerenr modi | prime minister narendra modi |
| arther n tickets for the game | are there any tickets for the game |

Table 1: Examples of transcriptions directly from the RNN (left) with errors that are fixed by addition of a language model (right).

| System | Clean (94) | Noisy (82) | Combined (176) |
|---|---|---|---|
| Apple Dictation | 14.24 | 43.76 | 26.73 |
| Bing Speech | 11.73 | 36.12 | 22.05 |
| Google API | 6.64 | 30.47 | 16.72 |
| wit.ai | 7.94 | 35.06 | 19.41 |
| **Deep Speech** | **6.56** | **19.06** | **11.85** |

Table 4: Results (%WER) for 5 systems evaluated on the original audio. Scores are reported *only* for utterances with predictions given by all systems. The number in parentheses next to each dataset, e.g. Clean (94), is the number of utterances scored.

| GPU | OpenMPI all-reduce | Our all-reduce | Performance Gain |
|---|---|---|---|
| 4 | 55359.1 | 2587.4 | 21.4 |
| 8 | 48881.6 | 2470.9 | 19.8 |
| 16 | 21562.6 | 1393.7 | 15.5 |
| 32 | 8191.8 | 1339.6 | 6.1 |
| 64 | 1395.2 | 611.0 | 2.3 |
| 128 | 1602.1 | 422.6 | 3.8 |

**Table 7:** Comparison of two different all-reduce implementations. All times are in seconds. Performance gain is the ratio of OpenMPI all-reduce time to our all-reduce time.

| Language | Architecture | CPU CTC Time | GPU CTC Time | Speedup |
|---|---|---|---|---|
| English | 5-layer, 3 RNN | 5888.12 | 203.56 | 28.9 |
| Mandarin | 5-layer, 3 RNN | 1688.01 | 135.05 | 12.5 |

**Table 8:** Comparison of time spent in seconds in computing the CTC loss function and gradient in one epoch for two different implementations. Speedup is the ratio of CPU CTC time to GPU CTC time.

Awni Hannun , Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng
.Deep Speech: Scaling up end-to-end speech recognition. 2014

# Acoustic Models

## Simplifying with 1D convolutional layers: Wave2Letter

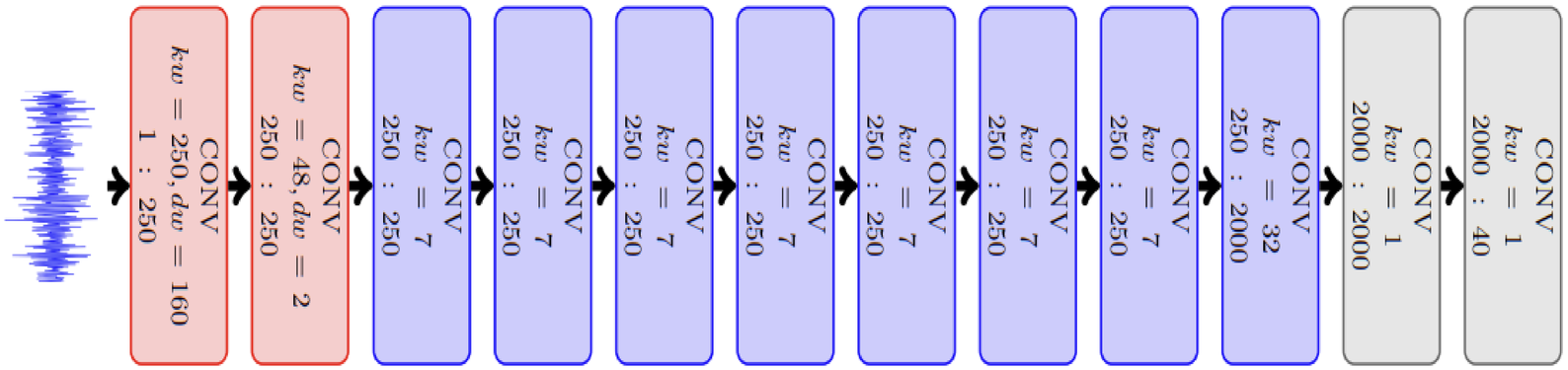


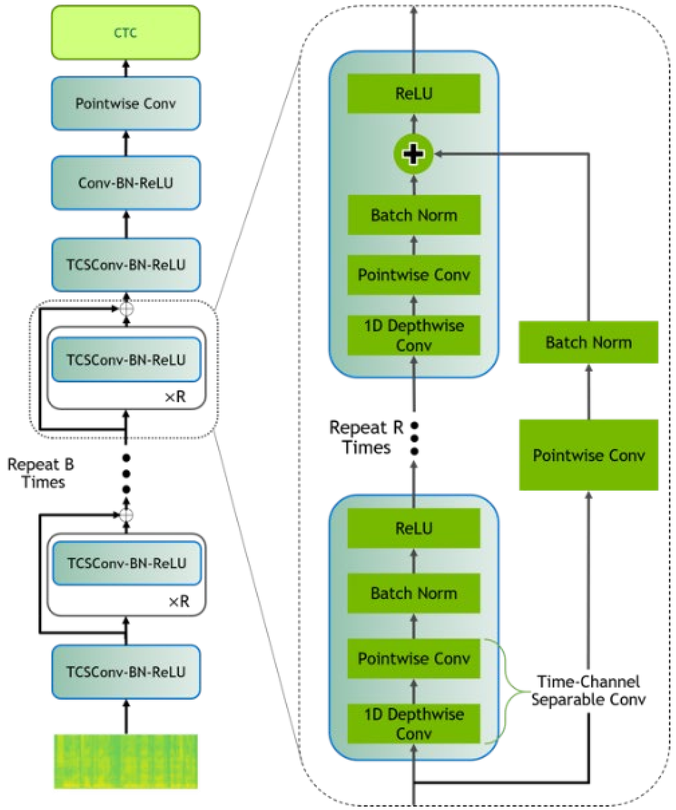

Table 2: LER/WER of the best sets of hyper-parameters for each feature types.

| | MFCC | | PS | | Raw | |
|---|---|---|---|---|---|---|
| | LER | WER | LER | WER | LER | WER |
| dev-clean | 6.9 | | 9.3 | | 10.3 | |
| test-clean | 6.9 | 7.2 | 9.1 | 9.4 | 10.6 | 10.1 |

Ronan Collobert and Christian Puhrsch and Gabriel Synnaeve. Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. 2016
https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition/wave2letter.html

# Acoustic Models

## QuartzNet




## Citrinet

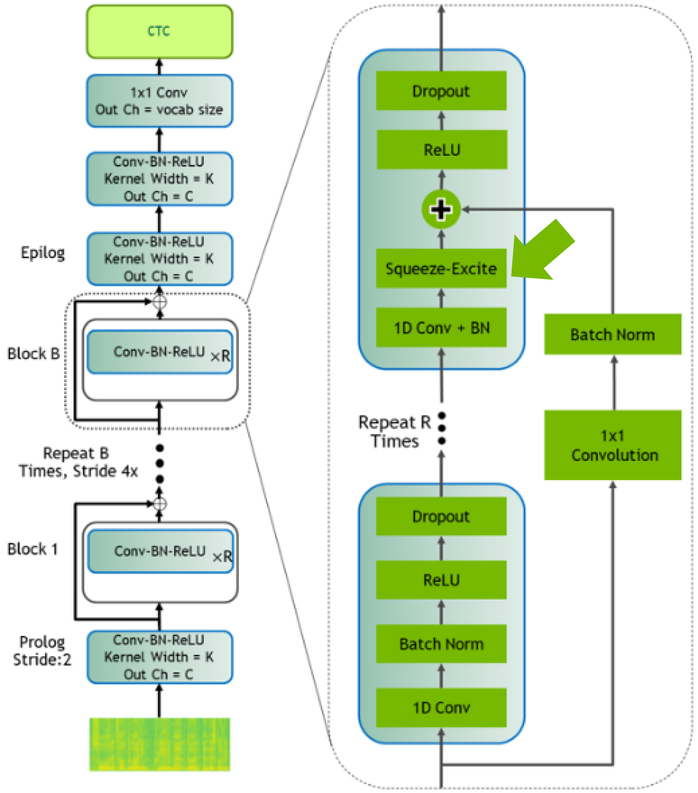



Table 3: *LibriSpeech: Citrinet vs Transducers, WER(%)*

| Model | LM | Test clean | Test other | Params, M |
|---|---|---|---|---|
| ContextNet-L [14] | - | 2.10 | 4.60 | 112.7 |
| | RNN | 1.90 | 4.10 | |
| Conformer-L[15] | - | 2.10 | 4.30 | 118 |
| | RNN | 1.90 | 3.90 | |
| Citrinet-256 | - | 3.78 | 9.6 | 9.8 |
| | 6-gram | 3.65 | 8.06 | |
| | Transf | 2.75 | 6.87 | |
| Citrinet-384 | - | 3.20 | 7.90 | 21.0 |
| | 6-gram | 2.94 | 6.71 | |
| | Transf | 2.52 | 5.95 | |
| Citrinet-512 | - | 3.11 | 7.82 | 36.5 |
| | 6-gram | 2.40 | 6.08 | |
| | Transf | 2.19 | 5.5 | |
| Citrinet-768 | - | 2.57 | 6.35 | 81 |
| | 6-gram | 2.15 | 5.11 | |
| | Transf | 2.04 | 4.79 | |
| Citrinet-1024 | - | 2.52 | 6.22 | 142 |
| | 6-gram | 2.10 | 5.06 | |
| | Transf | 2.00 | 4.69 | |

Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., ... & Zhang, Y. (2020, May). QuartzNet: Deep automatic speech recognition with 1d time-channel separable convolutions. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6124-6128). IEEE.

# Acoustic Models

## Conformer and Conformer-CTC

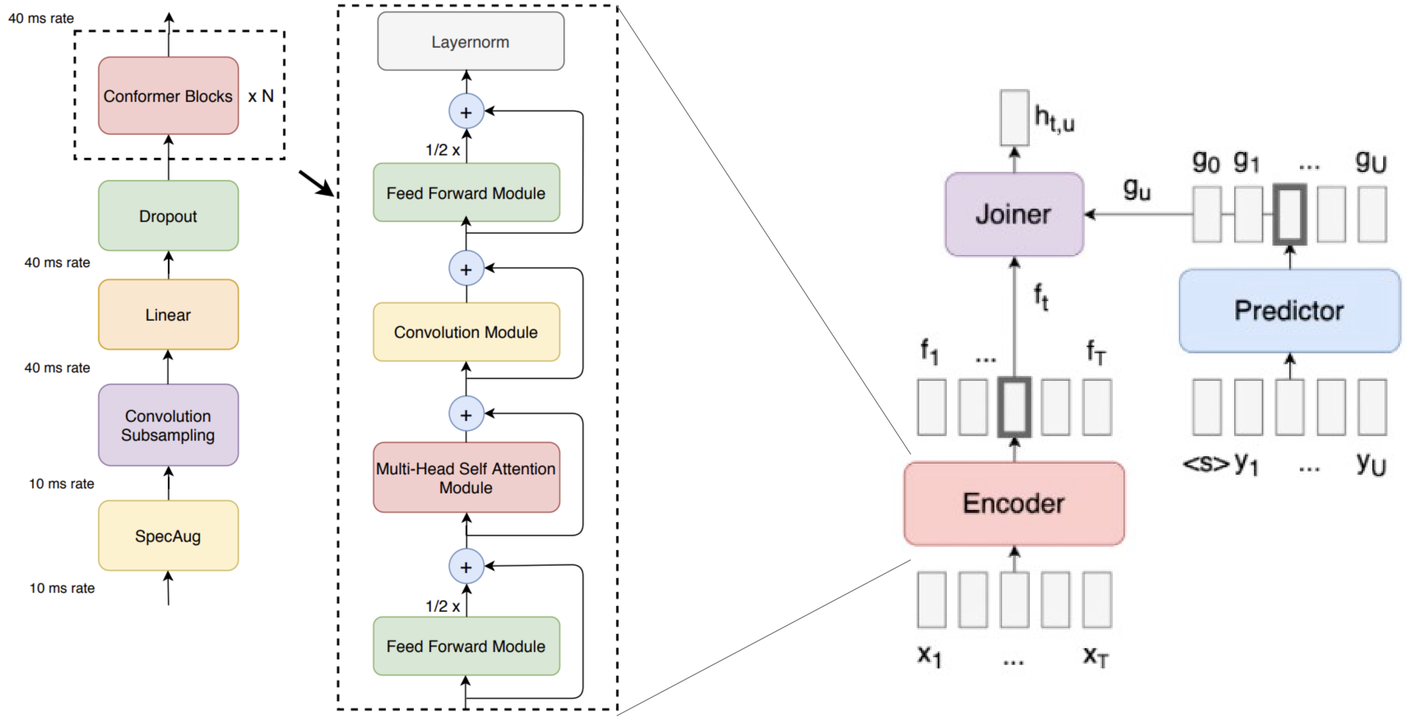



Figure 1: ***Conformer encoder model architecture.*** *Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post layernorm.*

| Method | #Params (M) | WER Without LM | | WER With LM | |
|---|---|---|---|---|---|
| | | testclean | testother | testclean | testother |
| **Hybrid** | | | | | |
| Transformer [33] | - | - | - | 2.26 | 4.85 |
| **CTC** | | | | | |
| QuartzNet [9] | 19 | 3.90 | 11.28 | 2.69 | 7.25 |
| **LAS** | | | | | |
| Transformer [34] | 270 | 2.89 | 6.98 | 2.33 | 5.17 |
| Transformer [19] | - | 2.2 | 5.6 | 2.6 | 5.7 |
| LSTM | 360 | 2.6 | 6.0 | 2.2 | 5.2 |
| **Transducer** | | | | | |
| Transformer [7] | 139 | 2.4 | 5.6 | 2.0 | 4.6 |
| ContextNet(S) [10] | 10.8 | 2.9 | 7.0 | 2.3 | 5.5 |
| ContextNet(M) [10] | 31.4 | 2.4 | 5.4 | **2.0** | 4.5 |
| ContextNet(L) [10] | 112.7 | **2.1** | 4.6 | **1.9** | 4.1 |
| **Conformer (Ours)** | | | | | |
| Conformer(S) | 10.3 | **2.7** | **6.3** | **2.1** | **5.0** |
| Conformer(M) | 30.7 | **2.3** | **5.0** | **2.0** | **4.3** |
| Conformer(L) | 118.8 | **2.1** | **4.3** | **1.9** | **3.9** |

LSTM-layer decoder in all our models.

# Acoustic Models

## Conformer-CTC

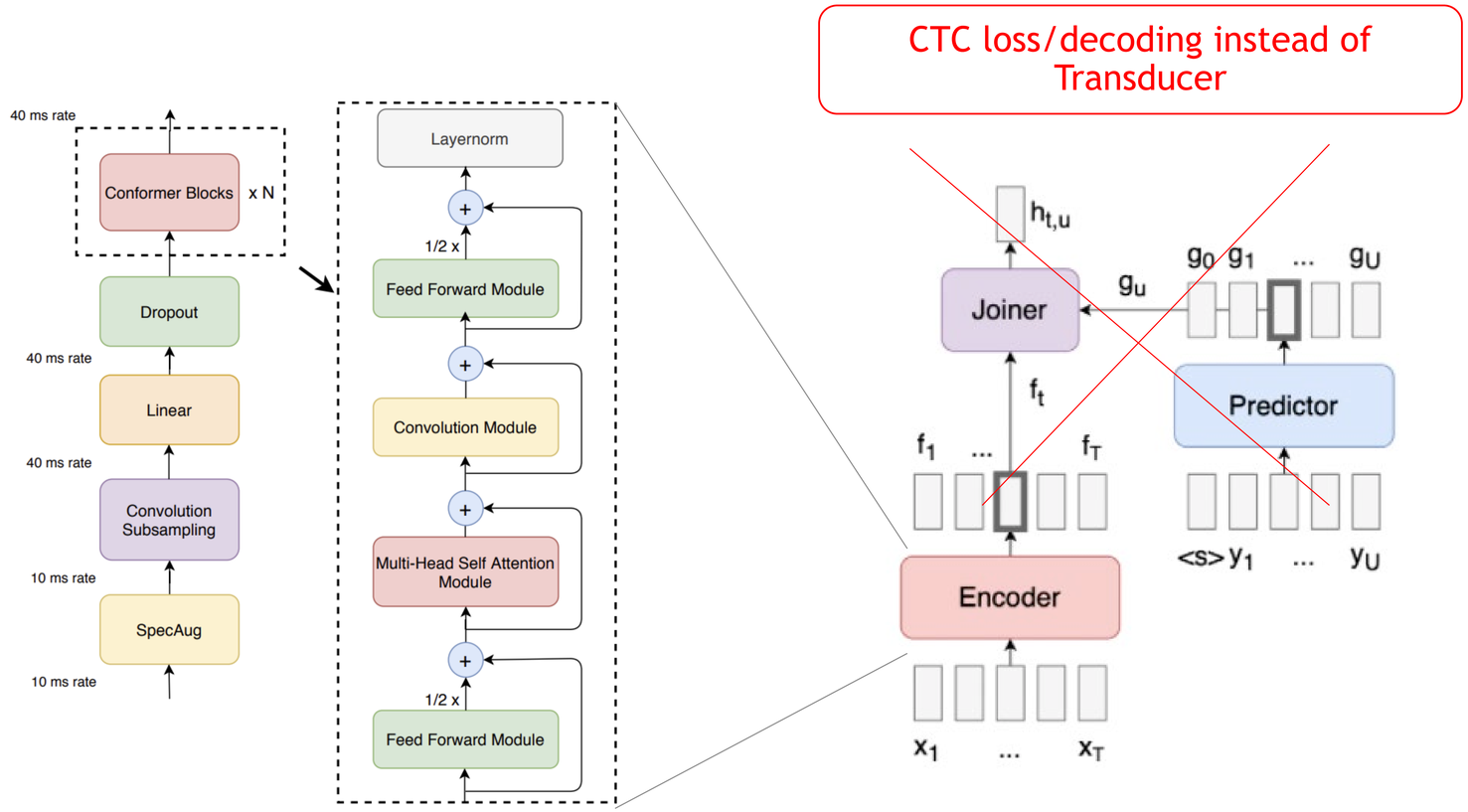
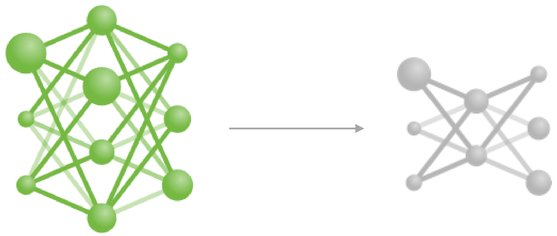



Figure 1: ***Conformer encoder model architecture.*** *Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post layernorm.*

# Acoustic Models

## Transfer learning

*"Transfer learning is a process of transferring learned features from one model to another"*



### Key Benefits

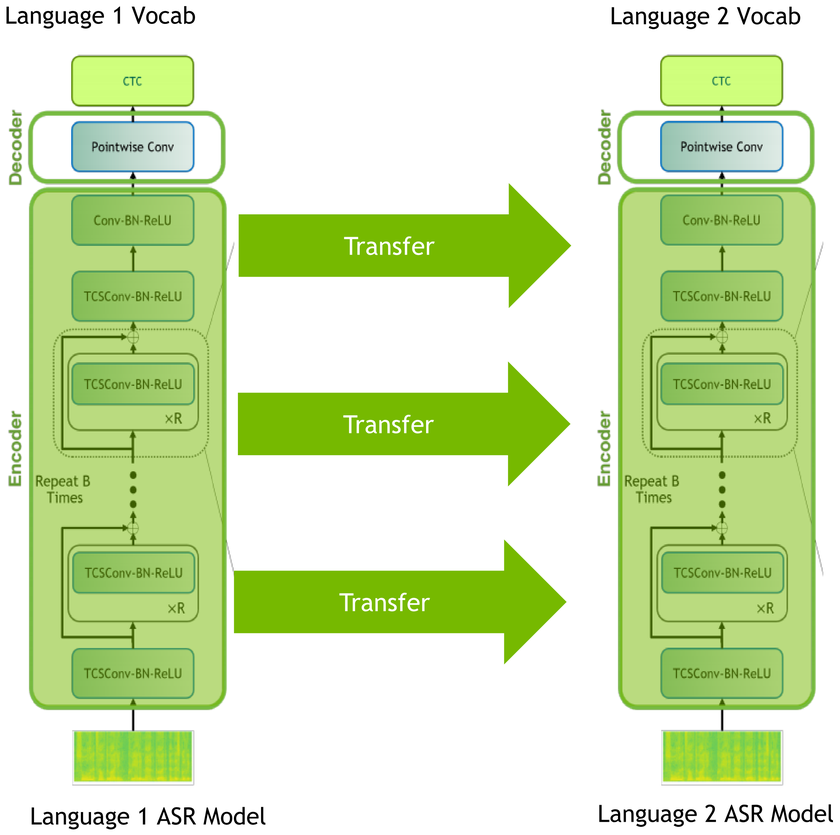
Less Data Required to Train Accurately

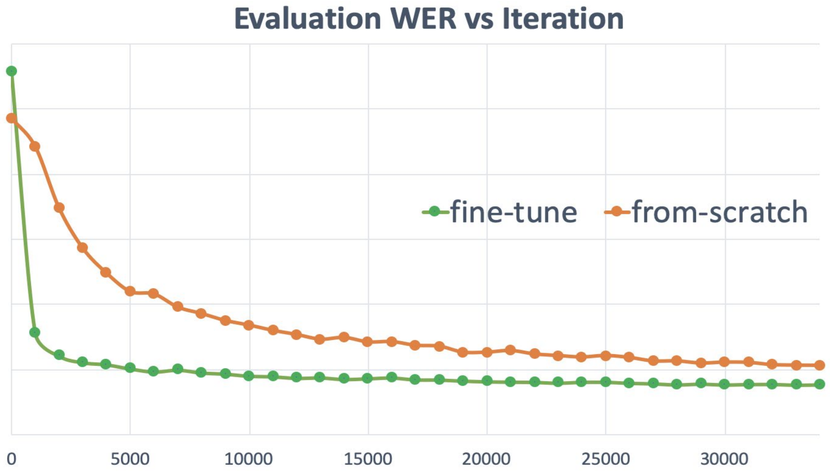

Reduce Training Time and Cost

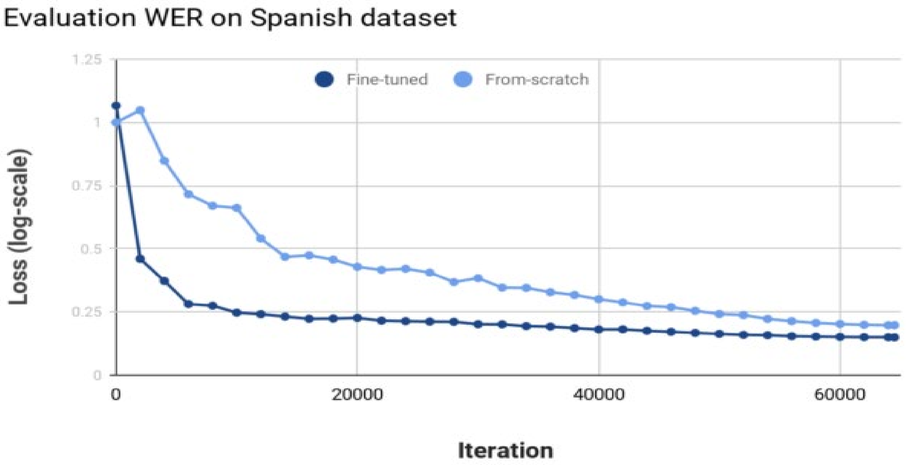
https://blogs.nvidia.com/blog/2019/02/07/what-is-transfer-learning/

# Acoustic Models

## Transfer learning



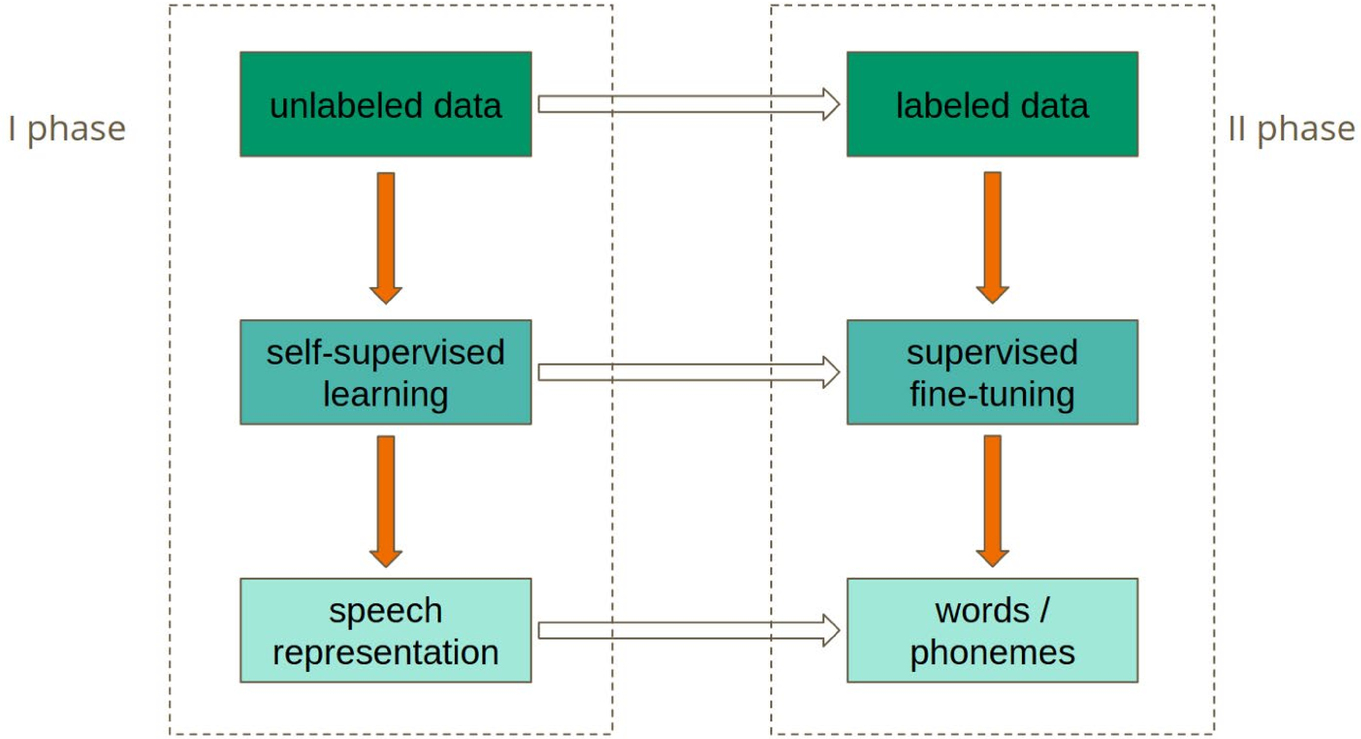




Jocelyn Huang, Oleksii Kuchaiev, Patrick O'Neill, Vitaly Lavrukhin, Jason Li, Adriana Flores, Georg Kucsko, Boris Ginsburg
Cross-Language Transfer Learning, Continuous Learning, and Domain Adaptation for End-to-End Automatic Speech Recognition, 2020

# Acoustic Models

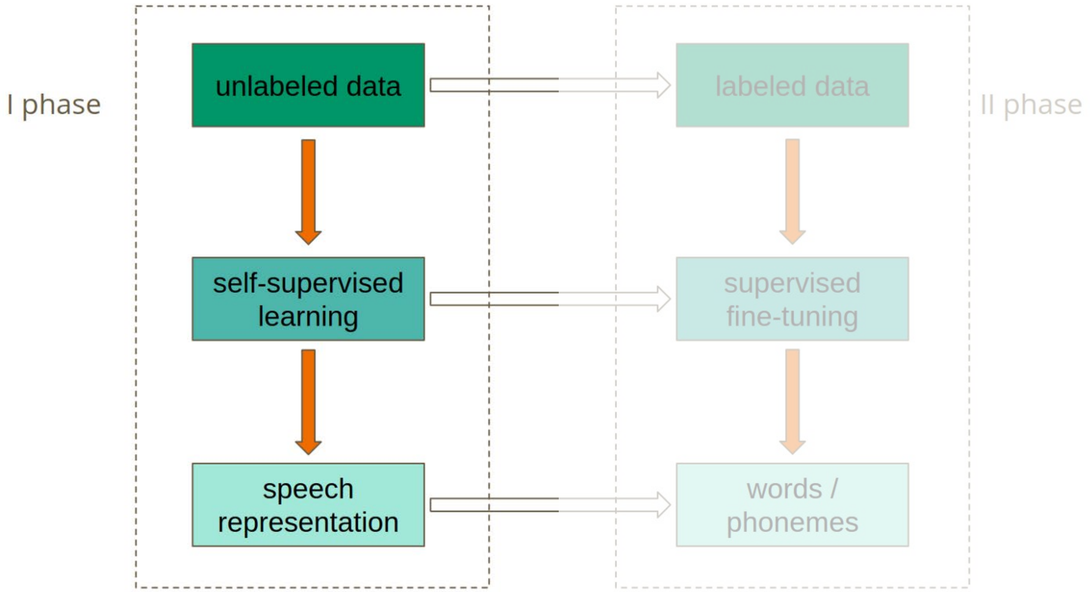
## Wave2vec 2.0

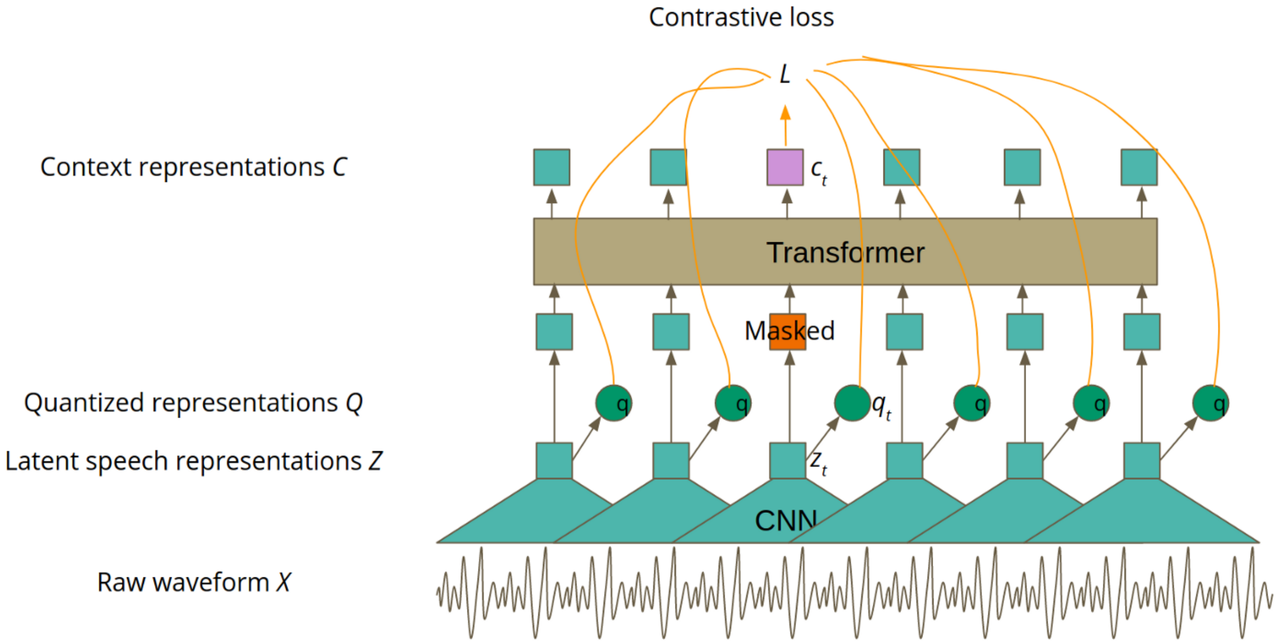



Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2020

# Acoustic Models

## Wave2vec 2.0

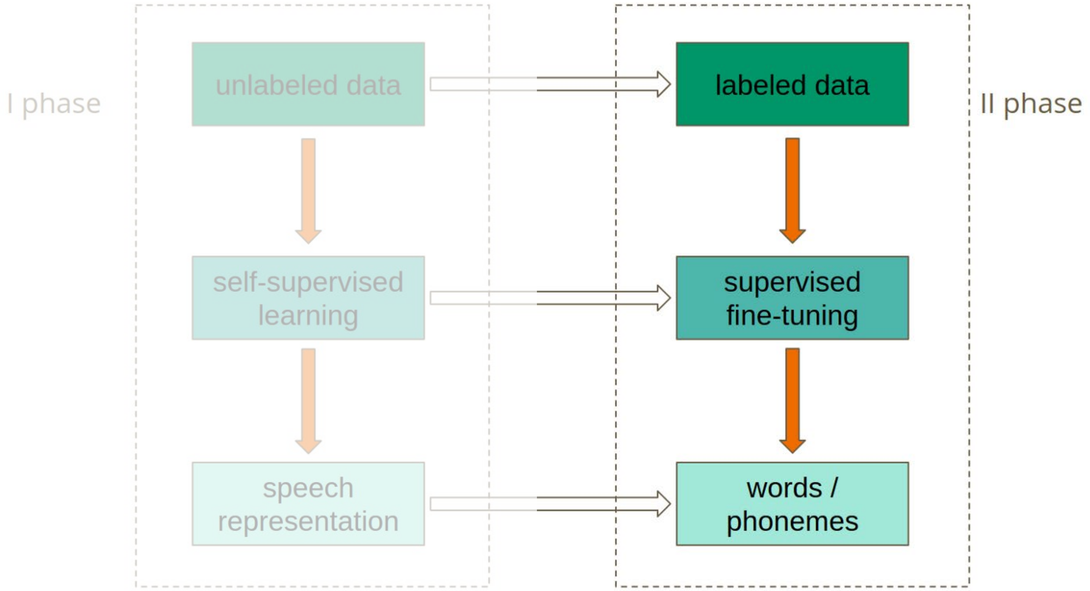


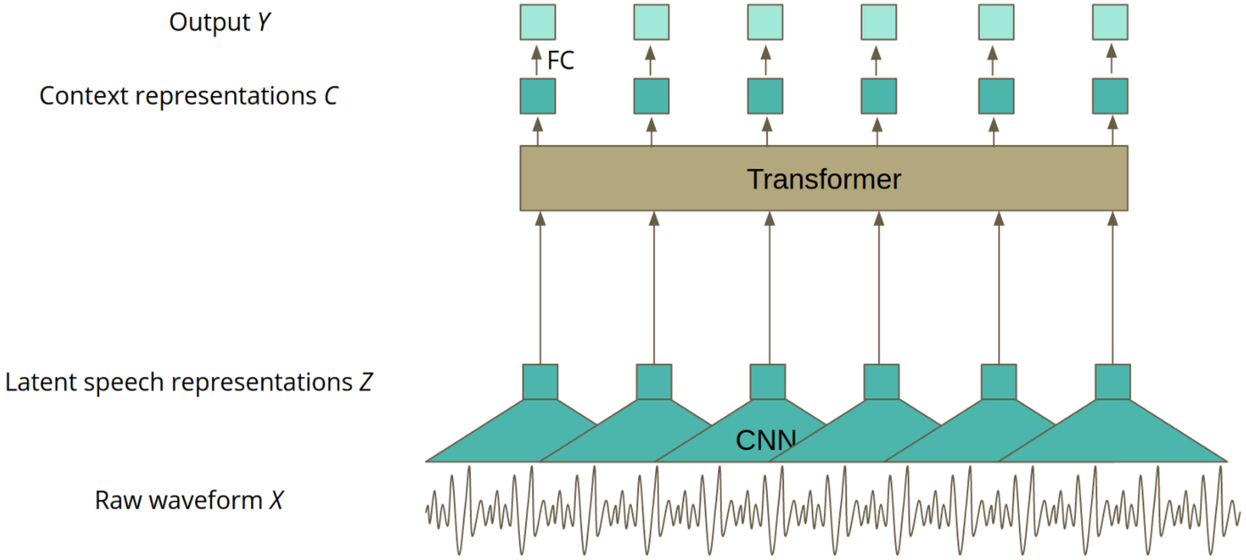




Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2020

# Acoustic Models

## Wave2vec 2.0




Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2020

# Acoustic Models

## Wave2vec 2.0

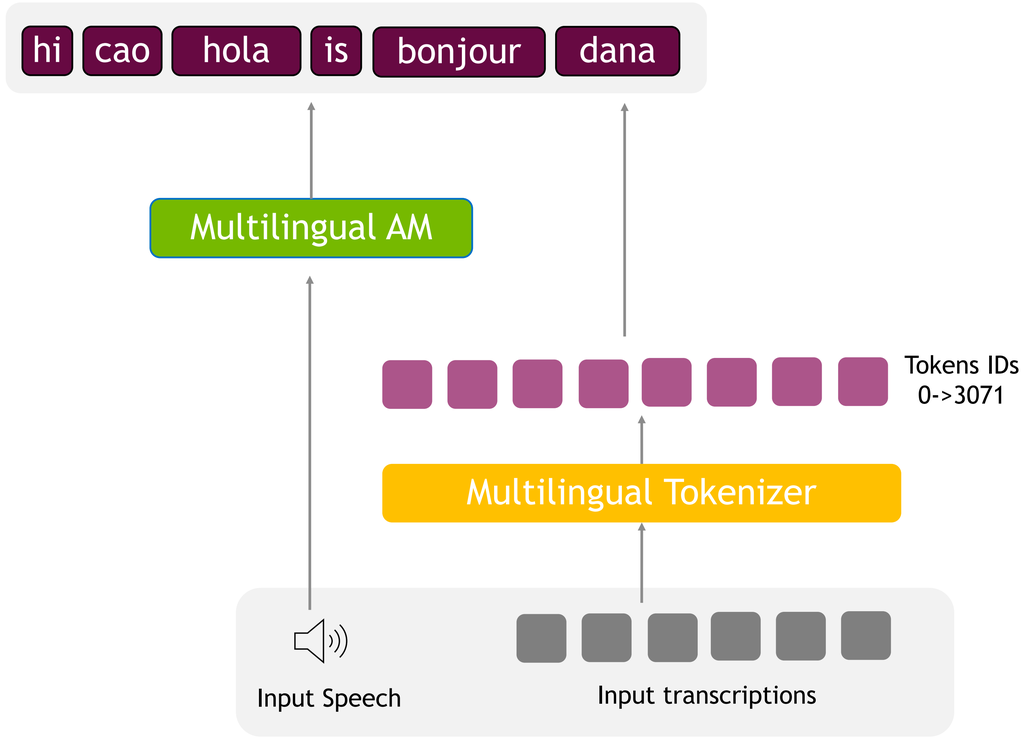
*“This demonstrates the feasibility of speech recognition with limited amounts of labeled data.”*

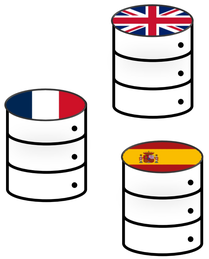
*“When lowering the amount of labeled data to one hour, wav2vec 2.0 outperforms the previous state of the art on the 100 hour subset while using 100 times less labeled data. “*

*“ Using just ten minutes of labeled data and pre-training on 53k hours of unlabeled data still achieves 4.8/8.2 WER.*

Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2020

# Multilingual vs Code-Switched Acoustic Models

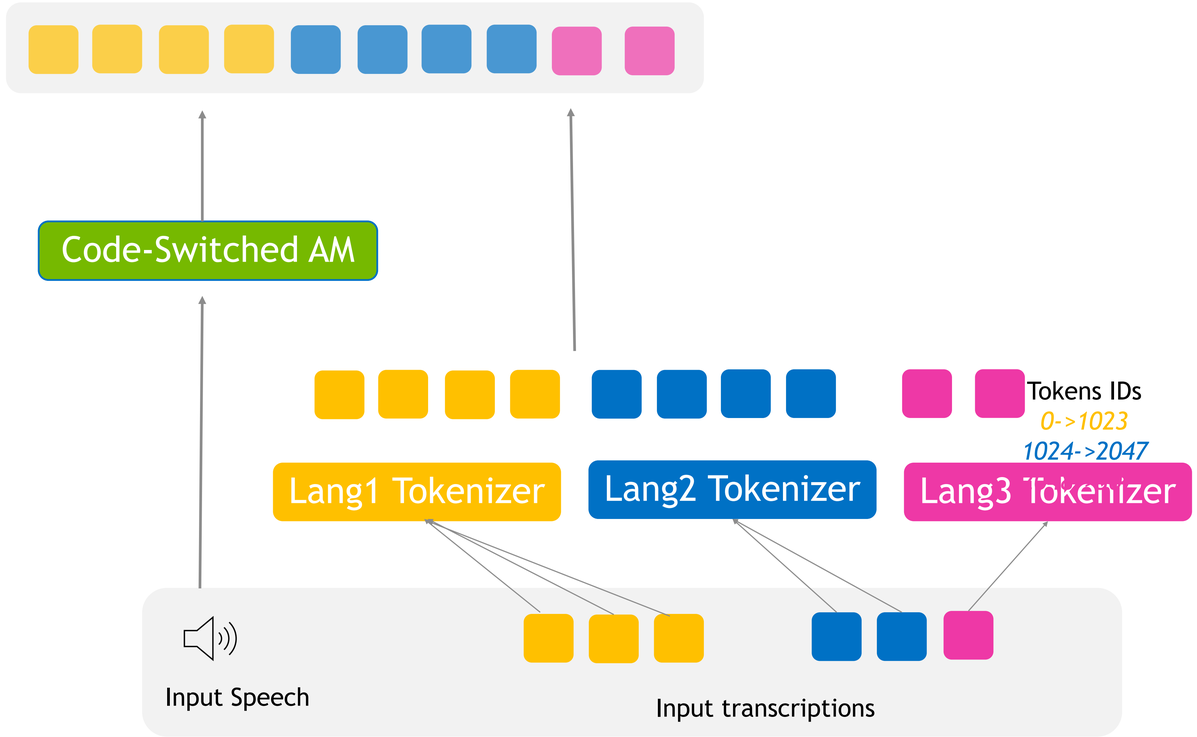


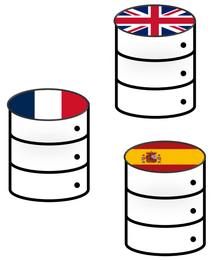







```
{"audio_filepath": "data1.wav", "duration": 3.04, "text": “once upon a time ", "lang": "en"}
{"audio_filepath": "data2.wav", "duration": 5.33, "text": “Bonjour, je souhaite connaitre la facture", "lang": “fr"}
....
```







```
{"audio_filepath": "/data2.wav", "duration": 17.32425, "text": [{"lang": "en", "str": "the mentality is that of a slave-owning community with a mutilated multitude of men tied to its commercial and political treadmill "}, {"lang": "es", "str": "y este "}]}
....
```

https://developer.nvidia.com/blog/multilingual-and-code-switched-automatic-speech-recognition-with-nvidia-nemo/

# Multilingual Acoustic Models

## Whisper

### Multitask training data (680k hours)

English transcription

- 🗣️ "Ask not what your country can do for ⋯"
- 📝 Ask not what your country can do for ⋯

Any-to-English speech translation

- 🗣️ "El rápido zorro marrón salta sobre ⋯"
- 📝 The quick brown fox jumps over ⋯

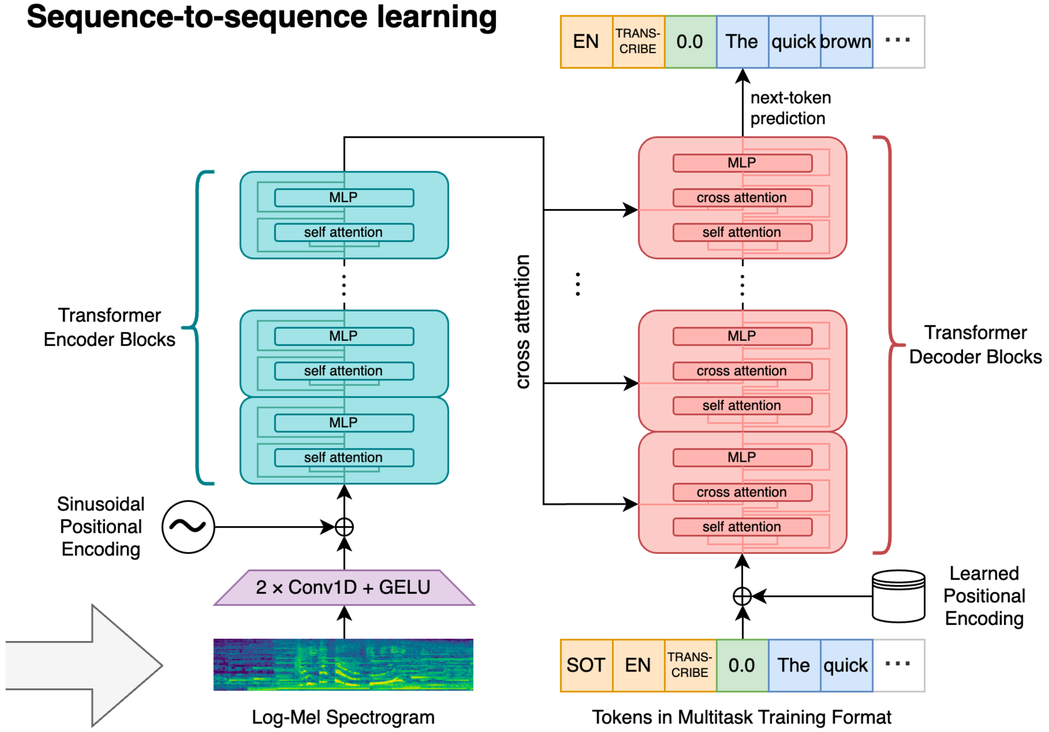
Non-English transcription

- 🗣️ "언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
- 📝 언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯

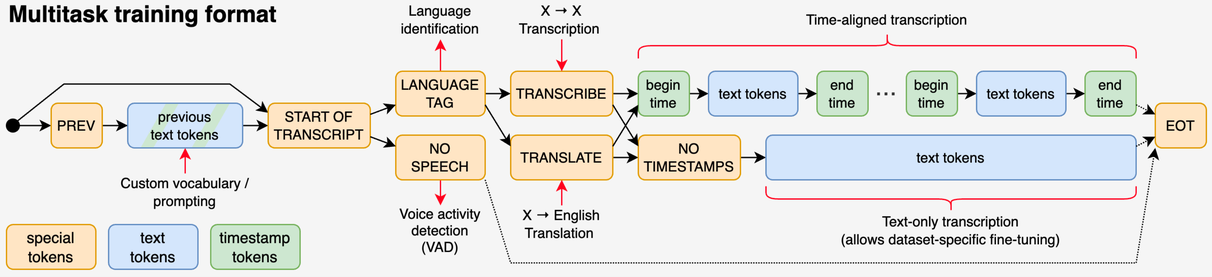
No speech

- 🔊 (background music playing)
- 📝 ∅

### Sequence-to-sequence learning




### Multitask training format



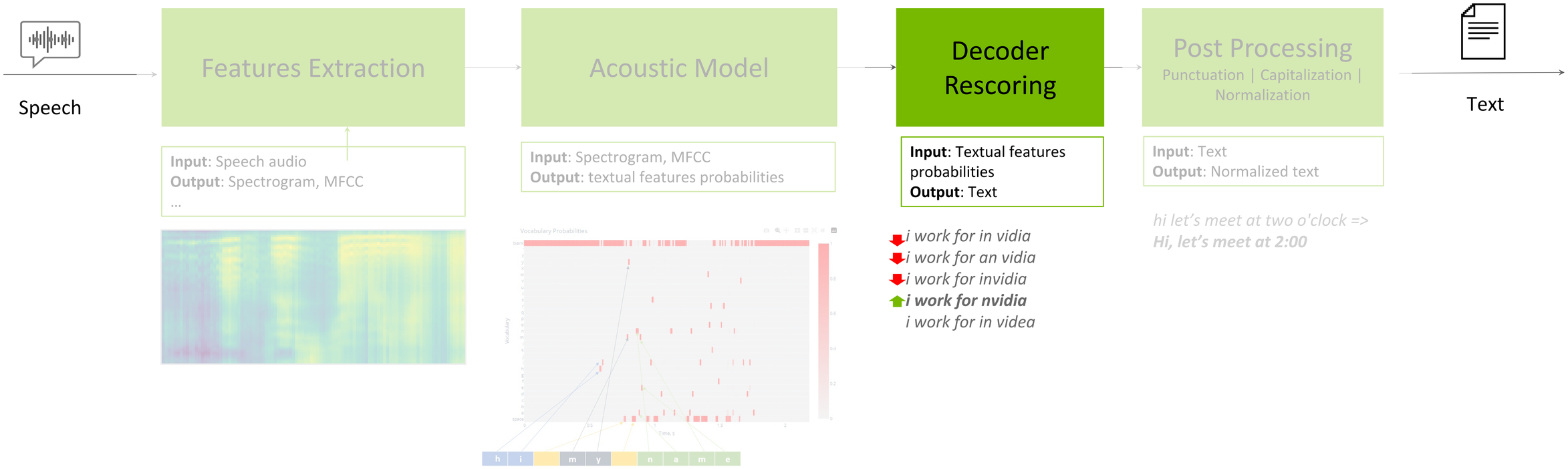

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey., Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision.2022

# Acoustic Models

Recap

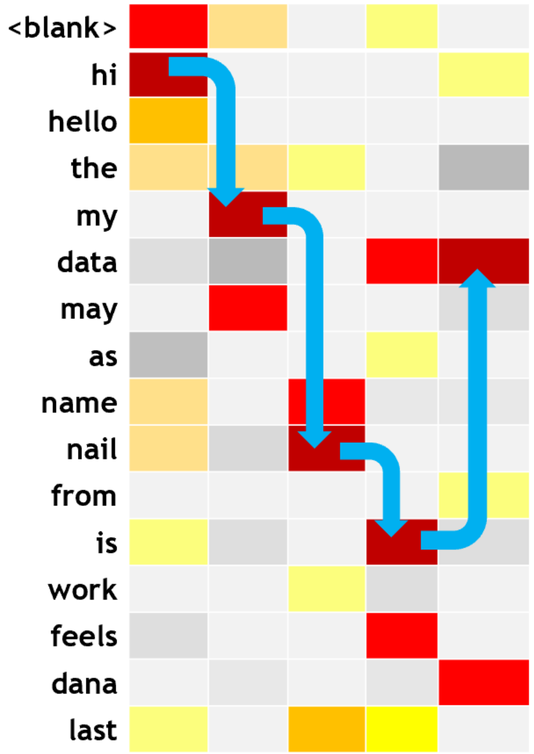
| Models | Tasks | Approach | Training | Comments |
|---|---|---|---|---|
| DeepSpeech | ASR | RNN-based + CTC | Supervised | **Autoregressive** |
| DeepSpeech2 | ASR | RNN-based + CTC | Supervised | **Autoregressive** |
| Wave2vec 2.0 | | Transformer + Convolutional | Self-Supervised + Supervised finetuning | Non Autoregressive |
| Citrinet | ASR | Convolutional + CTC | Supervised | Non Autoregressive |
| Conformer-CTC | ASR | Transformer + Convolutional + CTC | Supervised | Non Autoregressive |
| Conformer-RNNTransducer | ASR | Transformer + Convolutional + LSTM | Supervised | **Autoregressive** |
| Wave2Letter | ASR | Convolutional + CTC | Supervised | Non Autoregressive |
| Whisper | Multilingual Multitask | Transformer + Convolutional | Supervised | **Autoregressive** |

# ASR Pipeline

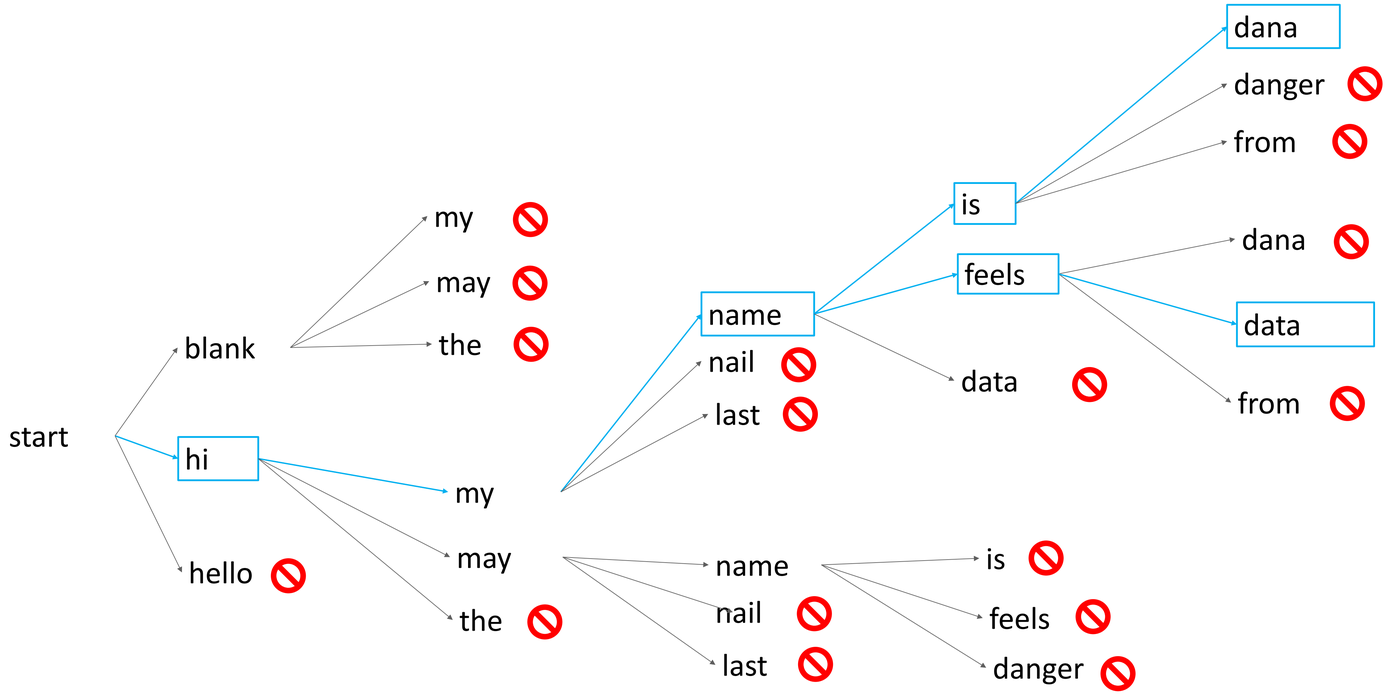


# Decoder Rescoring

Greedy decoder | beam search decoder




*Hi my nail is data*




*hi my name feels data | score 0.78*
*hi my name is dana | score = 0.72*

P(hi my name is dana) = p(hi) * p(my/hi) * p(name/hi my) * p(is/hi my name) * p(dana/hi my name is)

https://medium.com/@jessica_lopez/understanding-greedy-search-and-beam-search-98c1e3cd821d

# Decoder Rescoring

## Beam search decoder with a language model

- Language Model: The probability of a word depends its previous words.

P(hi my name is dana) = p(hi) * p(my/hi) * p(name/hi my) * p(is/hi my name) * p(dana/hi my name is)

- N-gram LM: The probability of a word depends on a fixed **N** previous words.

Example 2-gram: P(hi my name is dana) = p(hi) * p(my/hi) * p(name/hi my) * p(is/~~hi~~ my name) * p(dana/~~hi my~~ name is)

final_score = acoustic_score + **α** lm_score + β seq_length

*hi my name feels data | score 0.66*
*hi my name is dana | score = 0.79*

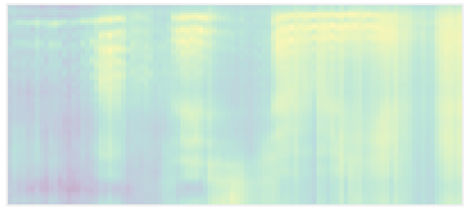
https://medium.com/@jessica_lopez/understanding-greedy-search-and-beam-search-98c1e3cd821d

# ASR Pipeline

Speech

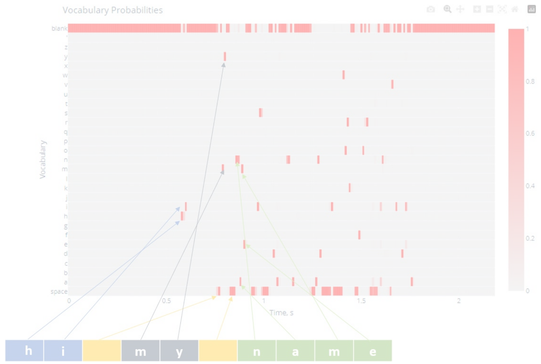
## Features Extraction

**Input**: Speech audio
**Output**: Spectrogram, MFCC
...



## Acoustic Model

**Input**: Spectrogram, MFCC
**Output**: textual features probabilities




## Decoder Rescoring

**Input**: Textual features probabilities
**Output**: Text

*i work for in vidia*
*i work for an vidia*
*i work for invidia*
***i work for nvidia***
*i work for in videa*

## Post Processing

Punctuation | Capitalization | Normalization

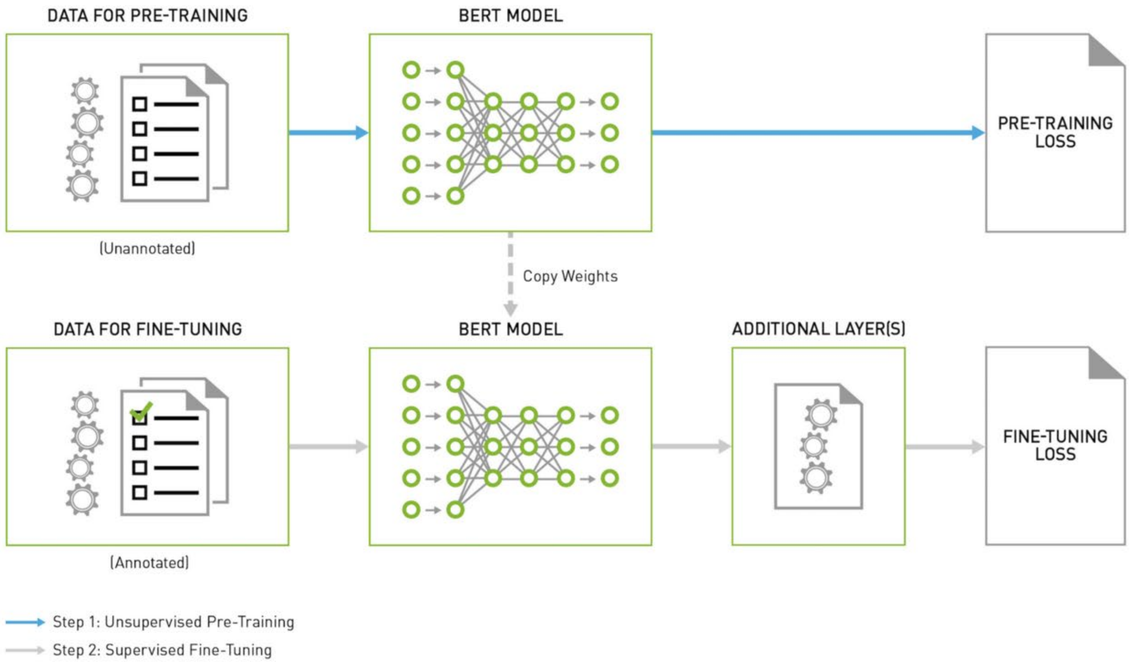
**Input**: Text
**Output**: Normalized text

*hi let's meet at two o'clock =>*
***Hi, let's meet at 2:00***

Text

# Post Processing

## Punctuation, Capitalization




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| when | is | the | next | flight | to | new | York |
| OU | OO | OO | OO | OO | OO | OU | ?U |
| When | is | the | next | flight | to | New | York? |

# Post Processing

## Normalization: Grammar Rules

Weighted finite-state transducer (WFST) grammars

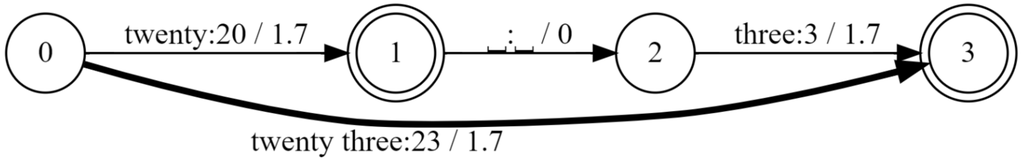



Figure 5: *In case of ambiguity the path with the smallest sum of weights will be chosen. Here "twenty three" is transduced to "23" instead of "20 3".*

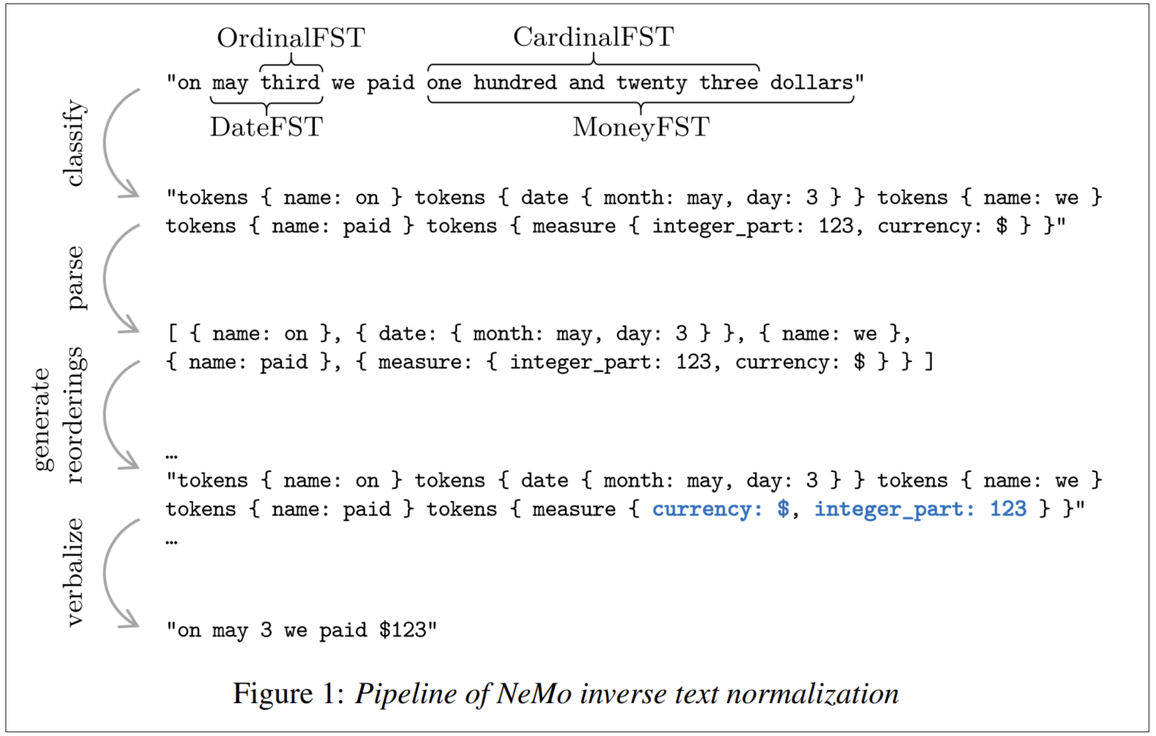



Figure 1: *Pipeline of NeMo inverse text normalization*

Yang Zhang , Evelina Bakhturina, Kyle Gorman , Boris Ginsburg. NeMo Inverse Text Normalization: From Development To Production. 2021

# Other Speech Tasks

# Speech Tasks

## Speech Command Recognition

The task of classifying an input audio pattern into a discrete set of classes.
E.g.: Yes, No, ”Go”, ”Stop”, ”Left”, ”Down”

## Audio Sentiment Classification

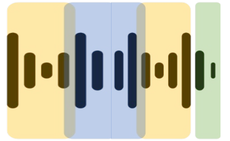
Extends the conventional text-based sentiment analysis to depend on the acoustic features extracted from speech.

## Voice Activity Detection (VAD)

The task of predicting which parts of input audio contain speech versus background noise.

## Speaker Diarization

Group same speakers



## Spoken Language Identification (LangID)

The task of recognizing the language of the spoken utterance automatically.


# ASR (Part 1)

## Lecture

- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

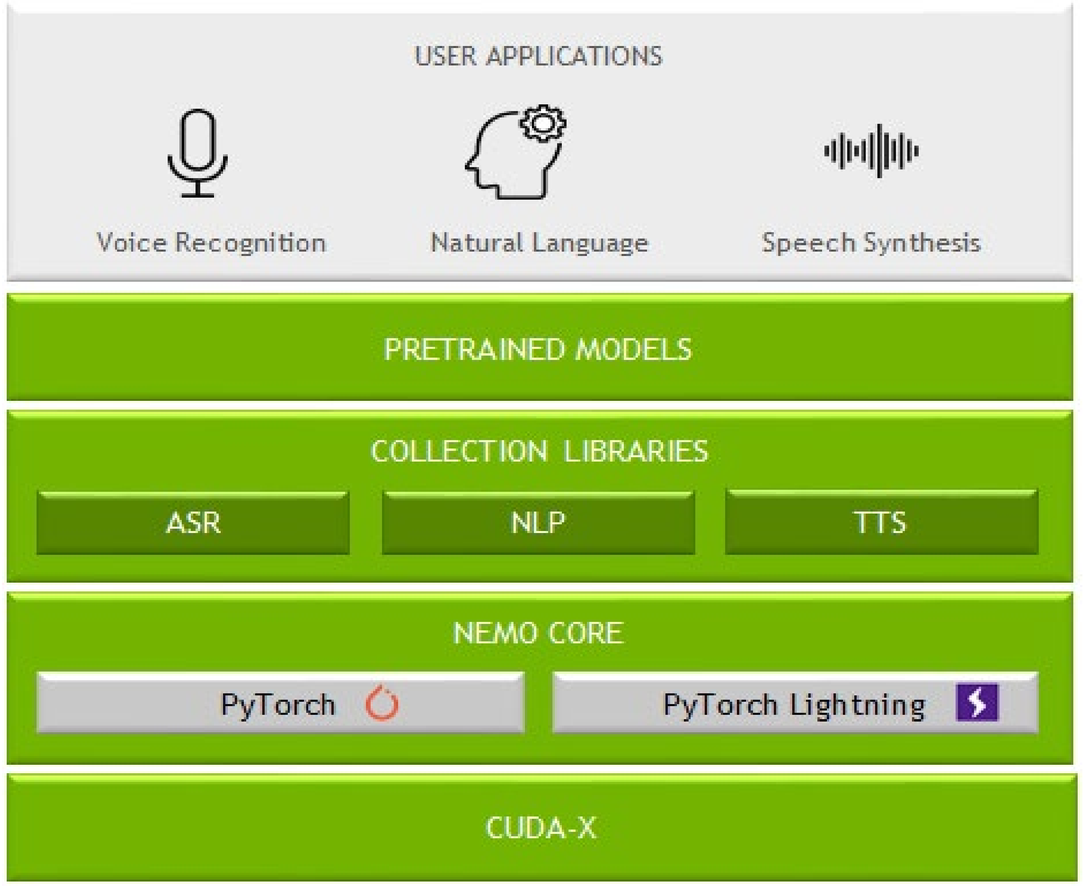
## Lab

- Investigate ASR Pipeline with NeMo and Riva

# What is NeMo?

Toolkit to build state-of-the-art conversational AI models

- DL-Based Speech & Language Understanding Models
- Include Semantic Checking for Correct-by-Construction Architectures
- Open-Sourced
- Integrated with PyTorch & PyTorch Lightning
- Easy-to-Use APIs
- Optimized Training Performance
- 100+ Pre-Trained GPU-Optimized Checkpoints
- Scale to 1000s of NVIDIA GPUs

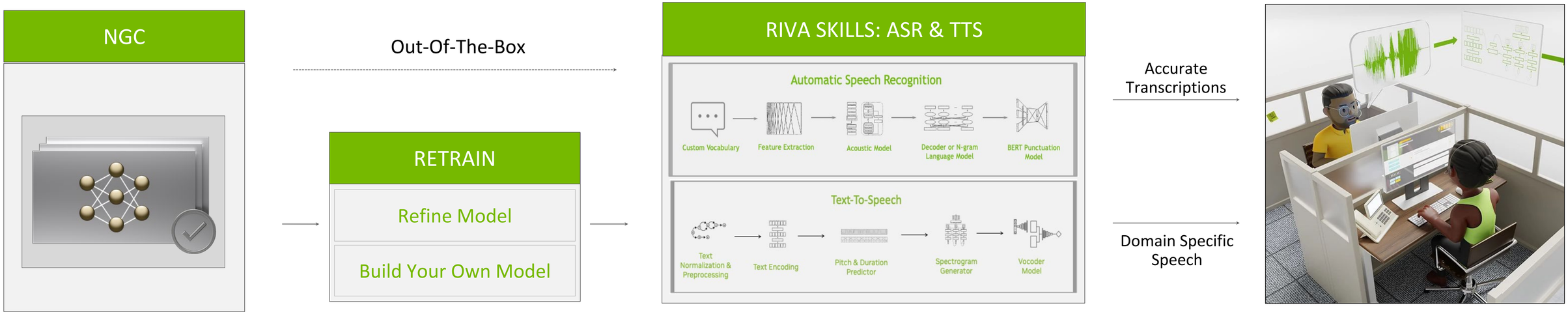



https://ngc.nvidia.com/catalog/containers/nvidia:nemo

https://github.com/NVIDIA/NeMo

# NVIDIA Riva

## Fully customizable, GPU-accelerated SDK for real-time Speech AI




SOTA Pre-Trained Models

Train for Any Domain and Language

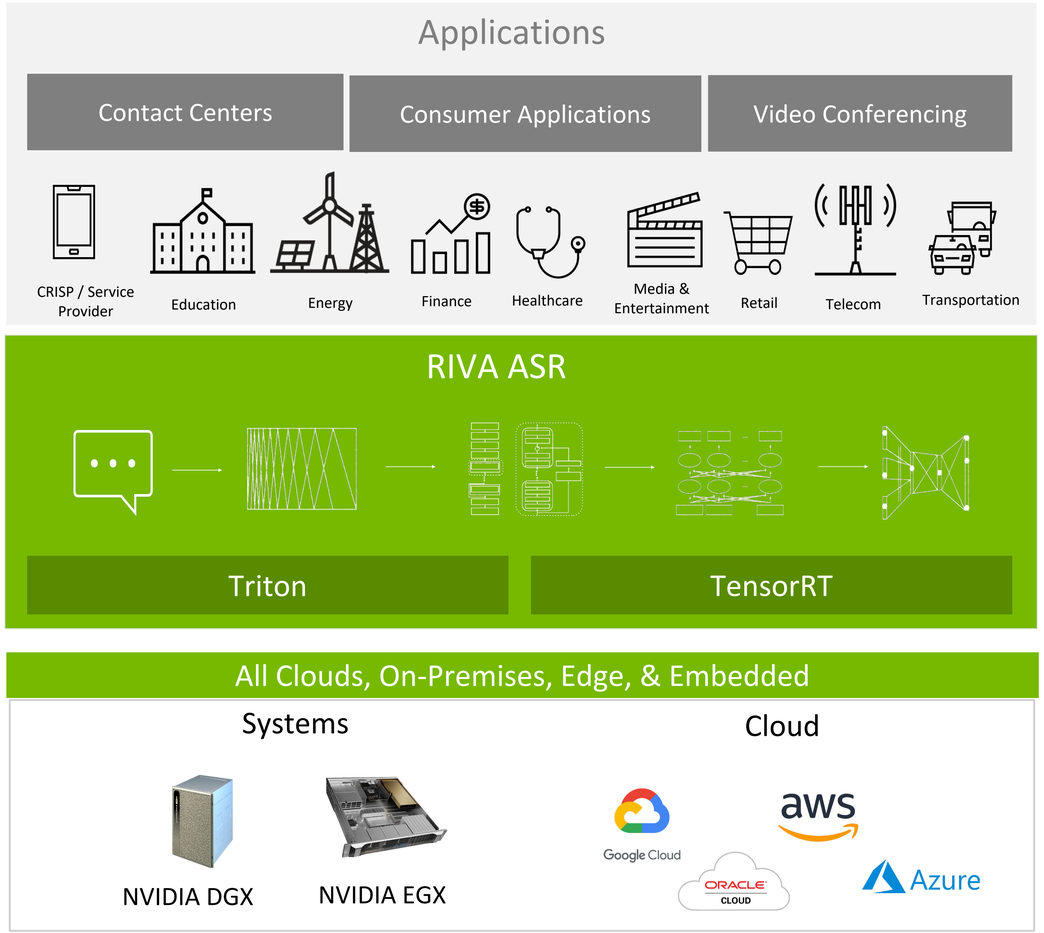
- Fully Customizable for the Best Possible Accuracy
- GPU-Accelerated, Real-Time
- Scale to Hundreds of Thousands of Users
- On-Prem, in Any Cloud, at the Edge, or Embedded

SOTA – State-Of-The-Art | ASR – Automatic Speech Recognition | TTS – Text-To-Speech

# NVIDIA Riva Automatic Speech Recognition

The best possible ASR with SOTA deep learning

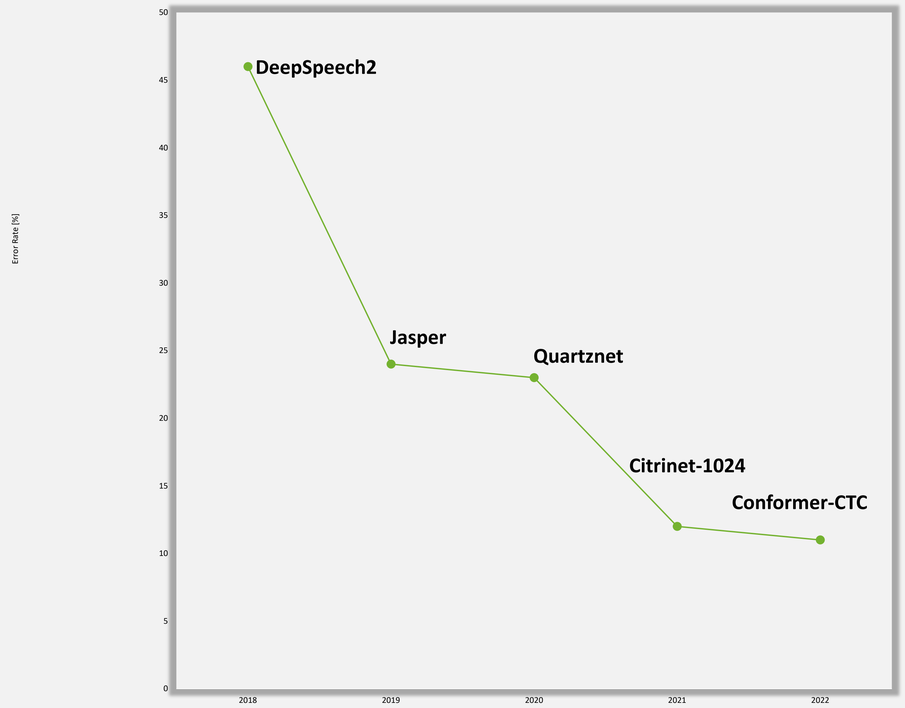
- **Accurate ASR with SOTA* models** trained for 1M+ hrs on 70K hrs of speech
- Support for:
  - **Languages**: English, Spanish, Mandarin, Hindi, Russian, Korean, German, French, Arabic, Italian, Japanese, Brazilian-Portuguese, European-Spanish
- **2X accuracy improvement** with customizations for:
  - Industry specific jargon
  - Accents & dialects
  - Noisy environments
- **Real-time** performance far below 300ms for interactive speech apps
- **High scale** of 100s thousands of concurrent streams
- **Runs anywhere**: all clouds, on-premises, at the edge, embedded




SOTA – State-Of-The-Art | ASR – Automatic Speech Recognition | TTS – Text-To-Speech

# NVIDIA Riva Delivers World-Class ASR

## 4X Accuracy with SOTA Model Architectures



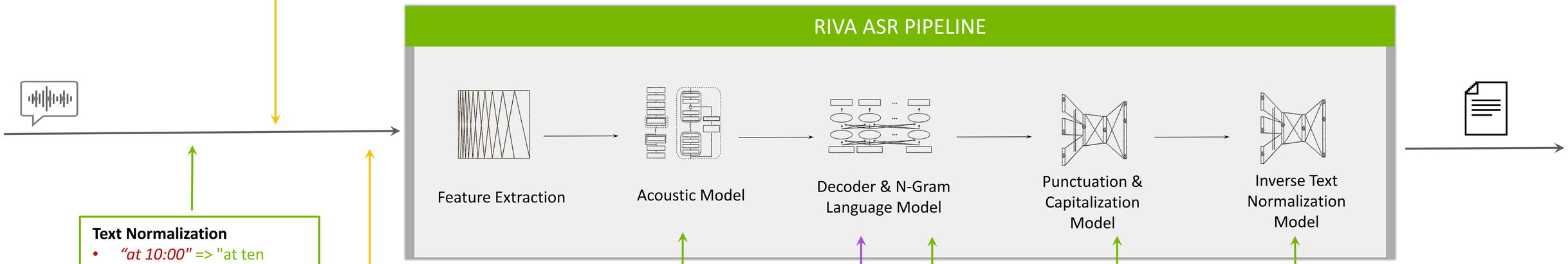

## Additional 2X Accuracy with Customizations


ASR – Automatic Speech Recognition | SOTA – State-Of-The-Art

# Customizations Across Riva ASR Pipeline

Accurate | Customizable | Real-Time

**Word Boosting*** for:
- Product Names
- Jargon
- Context-Specific Knowledge




**Text Normalization**
- *"at 10:00"* => "at ten o'clock "

**Speech Hints*** for:
- Priming Model to Expect Specific Information Based on Conversation Flow

(Credit Card Info, Order Number, Home Address, etc.)

**Fine-Tuning* or Retraining** for:
- Different Accents
- Noisy Environments
- Specific Pronunciation

**Retraining LM***

**Fine-Tuning or Custom Model***

**Adding Custom Pronunciation & Vocabulary** for:
- Domain-Specific Language
- Lingo

**Rule Modification***
- *"at ten o'clock"* => "at 10:00"
- *"confirmation number one tee four oh seven en"* => "Conf #1T407N"

Training (server-side) – Need to rebuild/restart server

Deployment

Inference (client-side) – No need to rebuild/start Riva server

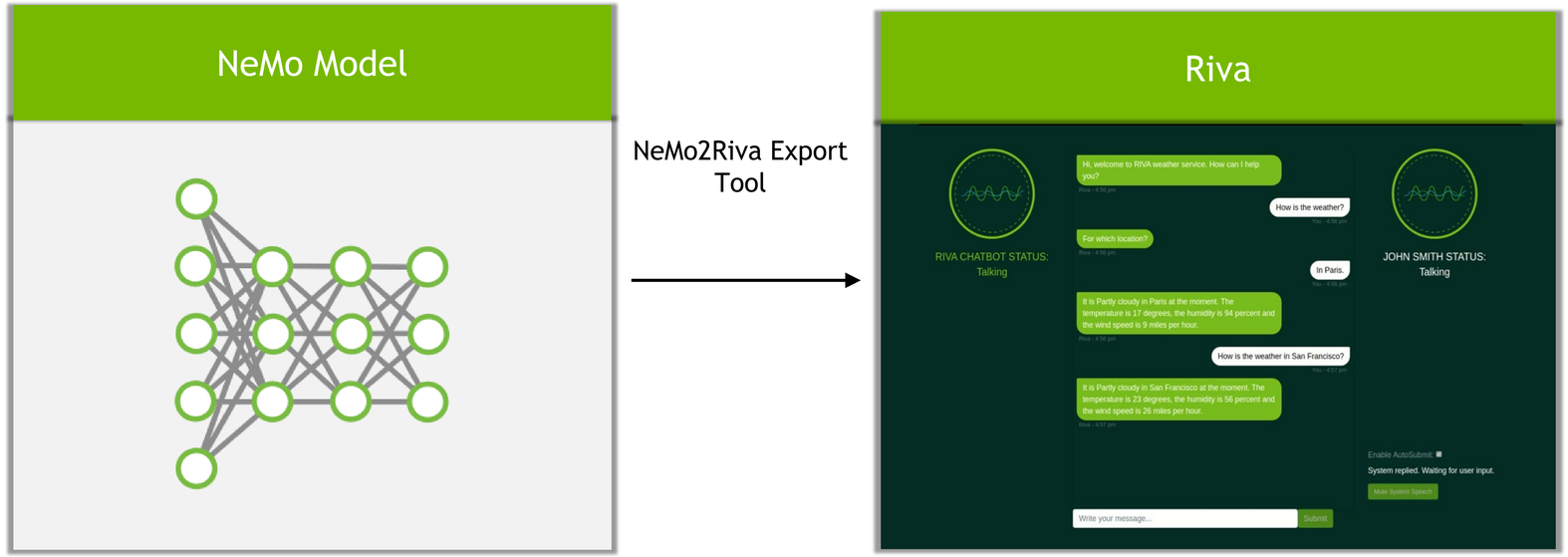
* Generic ASR Services offer black box with rigid & selectively limited customization

NVIDIA

# Deploy to Production

## Generate high-performance inference

- Quickly export NeMo models to Riva
- Support for speech and language models across multiple languages
- Step-by-step deployment instructions in documentation




Exporting NeMo Models to Riva


# ASR (Part 1)

## Lecture

- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

## Lab

- Investigate ASR Pipeline with NeMo and Riva

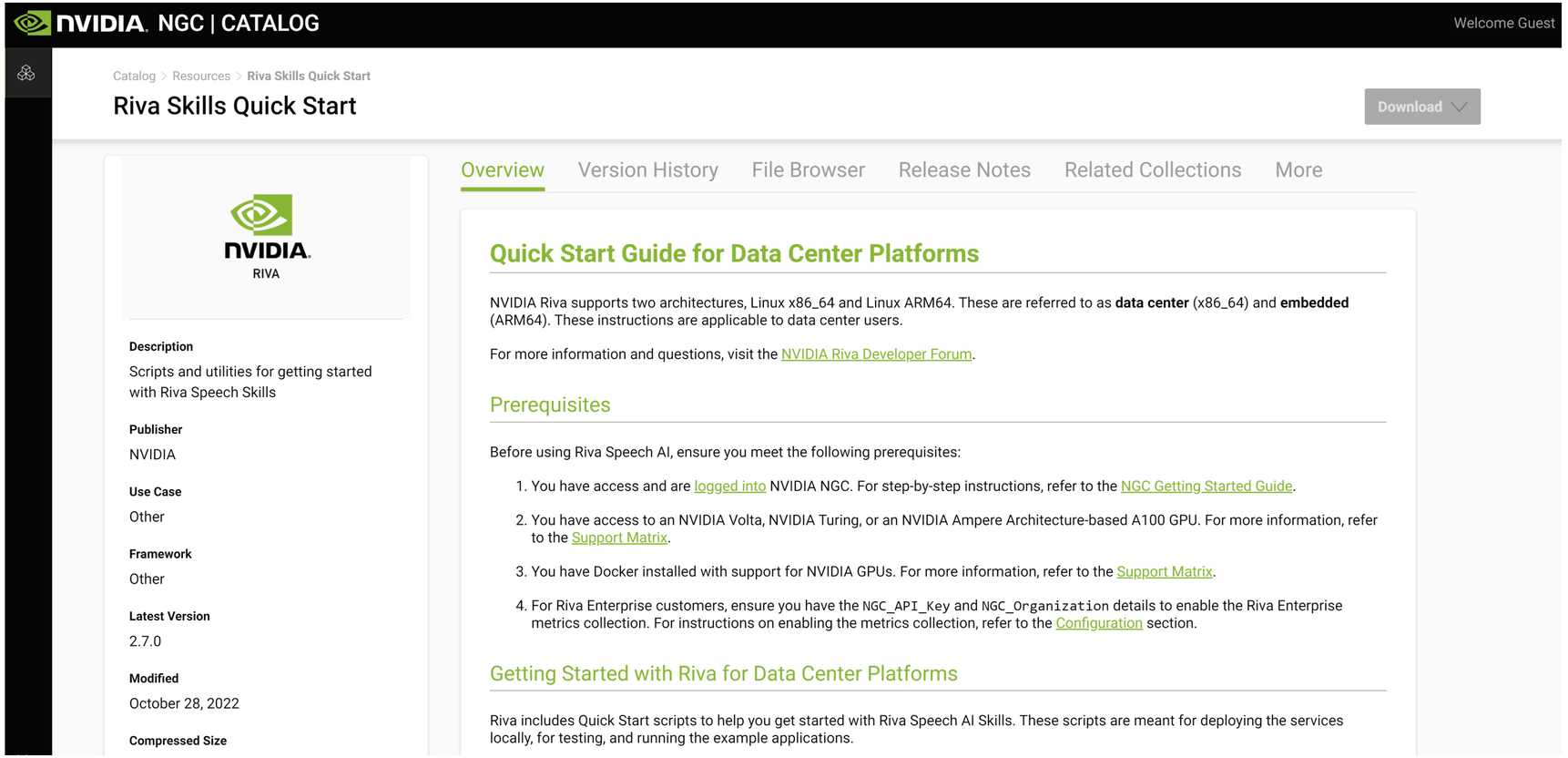
# Lab Overview

## ASR Pipeline

1. NGC Credential (Required for Riva deployment)
2. Investigate AI models for Automatic Speech Recognition
   - Gain an understanding about the ASR pipeline and the various acoustic models
   - Use transcript decoders to select the best transcriptions
   - Use ITN, capitalization, and punctuation to improve written transcriptions
   - Use a language identification model to identify what language was spoken
3. ASR deployment with Riva in several languages
   - Launch Riva ASR service for English and Spanish
   - Request the ASR service using a Python client API

# Nvidia RIVA

## Where to start?




1. **Configure Riva**
2. **Initialize Riva**
3. **Start Riva**

Download Riva from NGC

# Nvidia RIVA

## Launch Riva ASR services

### 1 Service Configuration

*Modify the config.sh file*

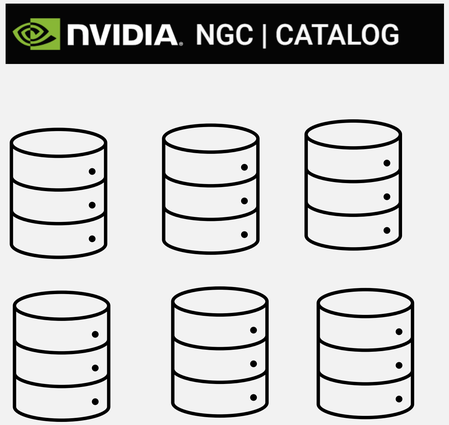
**Config.sh**

```
…
# Enable or Disable Riva Services
service_enabled_asr=true
service_enabled_nlp=false
service_enabled_tts=false
...
# Language code to fetch models of a specify language
# Currently only ASR supports languages other than English
# Supported language codes: en-US, en-GB, de-DE, es-US, ru-RU,
zh-CN, hi-IN, fr-FR, ko-KR, pt-BR
language_code=("en-US")
...
# Models ($riva_model_loc/models)
# During the riva_init process, the RMIR files in
$riva_model_loc/rmir
# are inspected and optimized for deployment. The optimized
versions are
# stored in $riva_model_loc/models. The riva server
exclusively uses these
# optimized versions.
riva_model_loc="/path/to/riva-model-repo"
```

### 2 Service Initialization

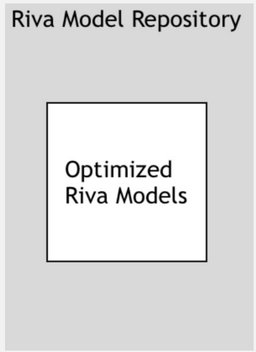
*Run: bash riva_init.sh*

**Download Models**




Download containers and the prebuild RMIR files
Riva Model Intermediate Representation

**riva-deploy**




Convert RMIR into the Riva model repository.
Neural networks are exported and optimized to run on the target platform

### 3 Service Start

*Run: bash riva_start.sh*

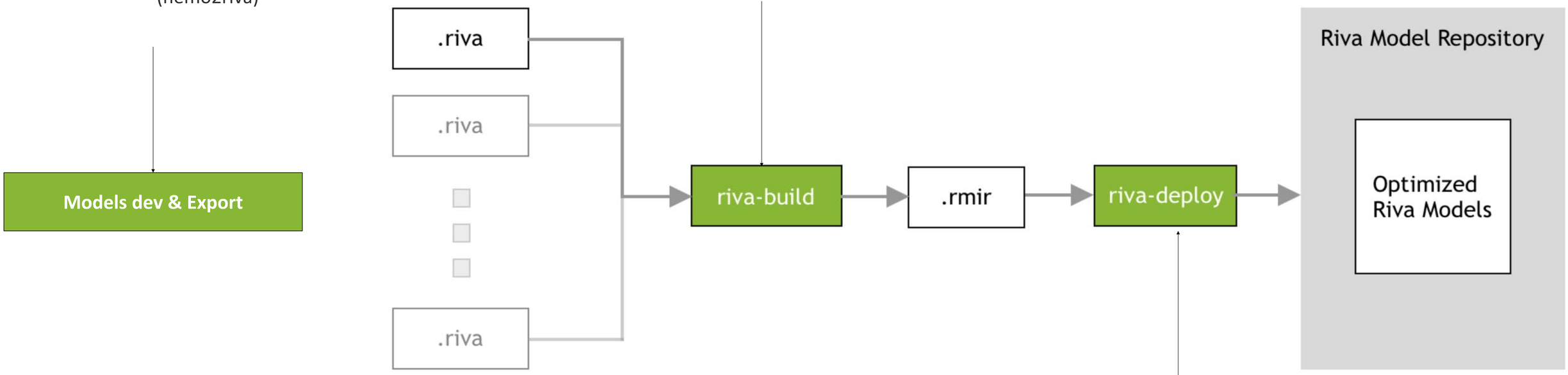
Download Riva from NGC

NVIDIA

# NVIDIA Riva

Phase 1
Convert model checkpoints to .riva format.
(nemo2riva)

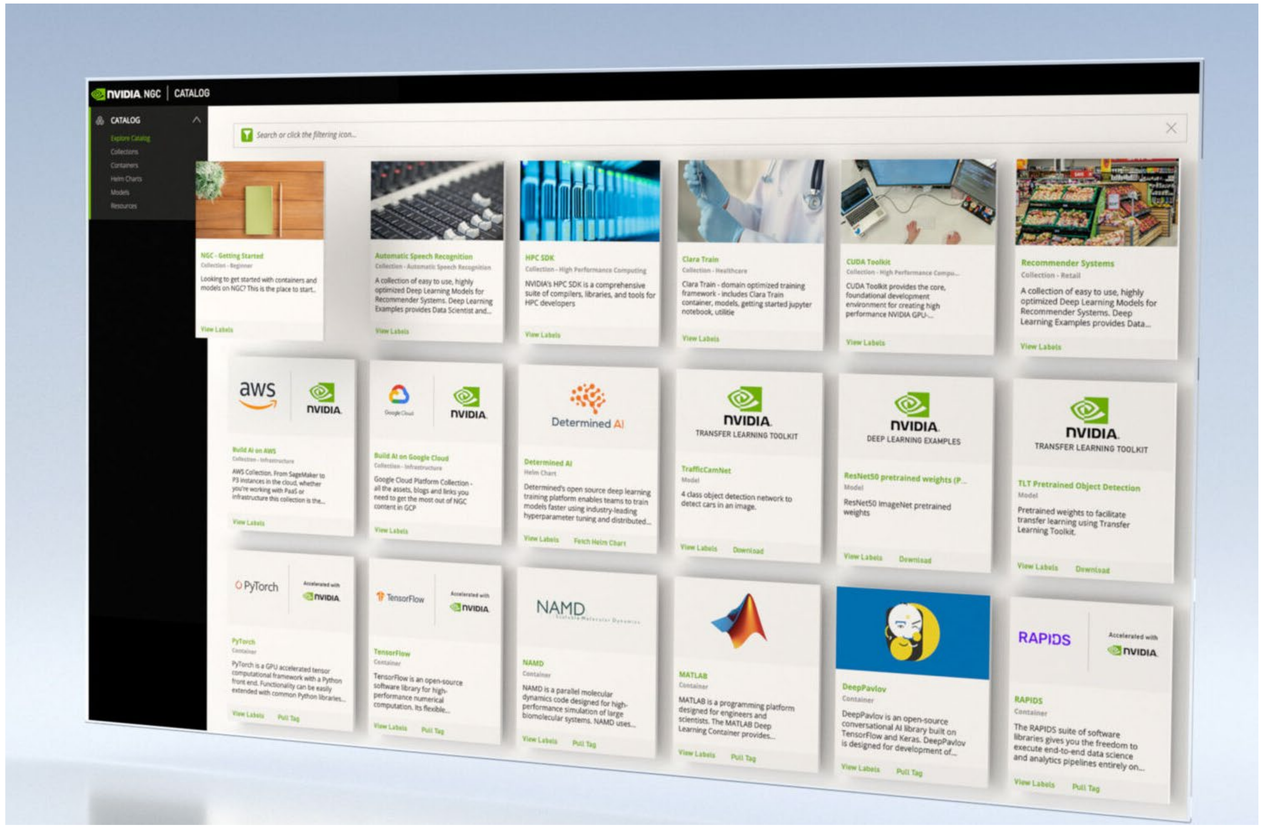
Phase 2
Gather necessary artifacts (models, files, configurations, and user settings) into RMIR (Riva Model Intermediate Representation)

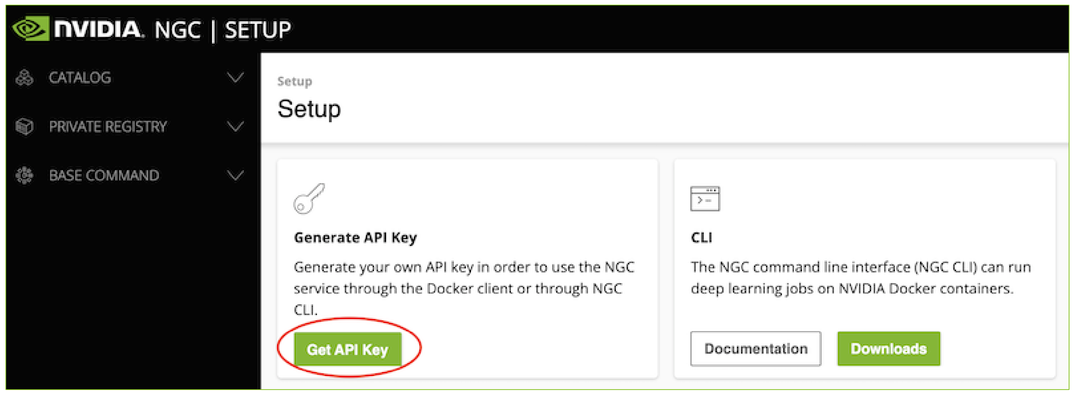



Phase 3
Convert RMIR into the Riva model repository.
Neural networks are exported and optimized to run on the target platform

# NVIDIA Riva

## NVIDIA GPU Cloud (NGC) setup




### Usage

Use your API key to log in to the NGC registry by entering the following command and following the prompts:

**NGC CLI**

```
$ ngc config set
```

**Docker™**

For the username, enter '$oauthtoken' exactly as shown. It is a special authentication token for all users.

```
$ docker login nvcr.io

Username: $oauthtoken
Password: <Your Key>
```

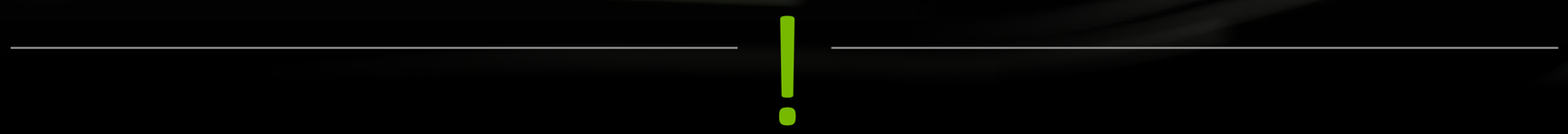

# ASR (Part 1)

## Lecture

- What is Conversational AI? Complexity of Deploying Conversational AI Applications
- Deep Learning Fundamentals
- Speech and Text Processing
- Automatic Speech Recognition
- ASR Tools: Riva, NGC, and NeMo
- Lab Overview

## Lab

- Investigate ASR Pipeline with NeMo and Riva



Lab


NVIDIA