UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

3546 – Deep Learning

Module 10 – Speech and Music Synthesis

Learning Outcomes for this Module

- Use tools to extract features from audio
- Experiment with tools for analyzing and synthesizing human speech
- Explore the history of computer composition
- Work with neural nets for generating speech, sound and music

Topics for this Module

- **10.1** Introduction
- **10.2** Audio Recognition
- **10.3** Speech Synthesis
- **10.4** Music Synthesis
- **10.5** Resources and Wrap-up

Module 10 – Section 1

Introduction

Applications (part 1)

- Voice recognition
- Biometric authentication
- Voice generation
- Speech2Face
- Video Magnification
- Spoken Language Processing
- Music composition
- Song recognition
- Song recommenders
- Morphing
- Part separation

Applications (part 2)

- Music transcription
- Instrument recognition
- Audio scene understanding
- Remote sensing

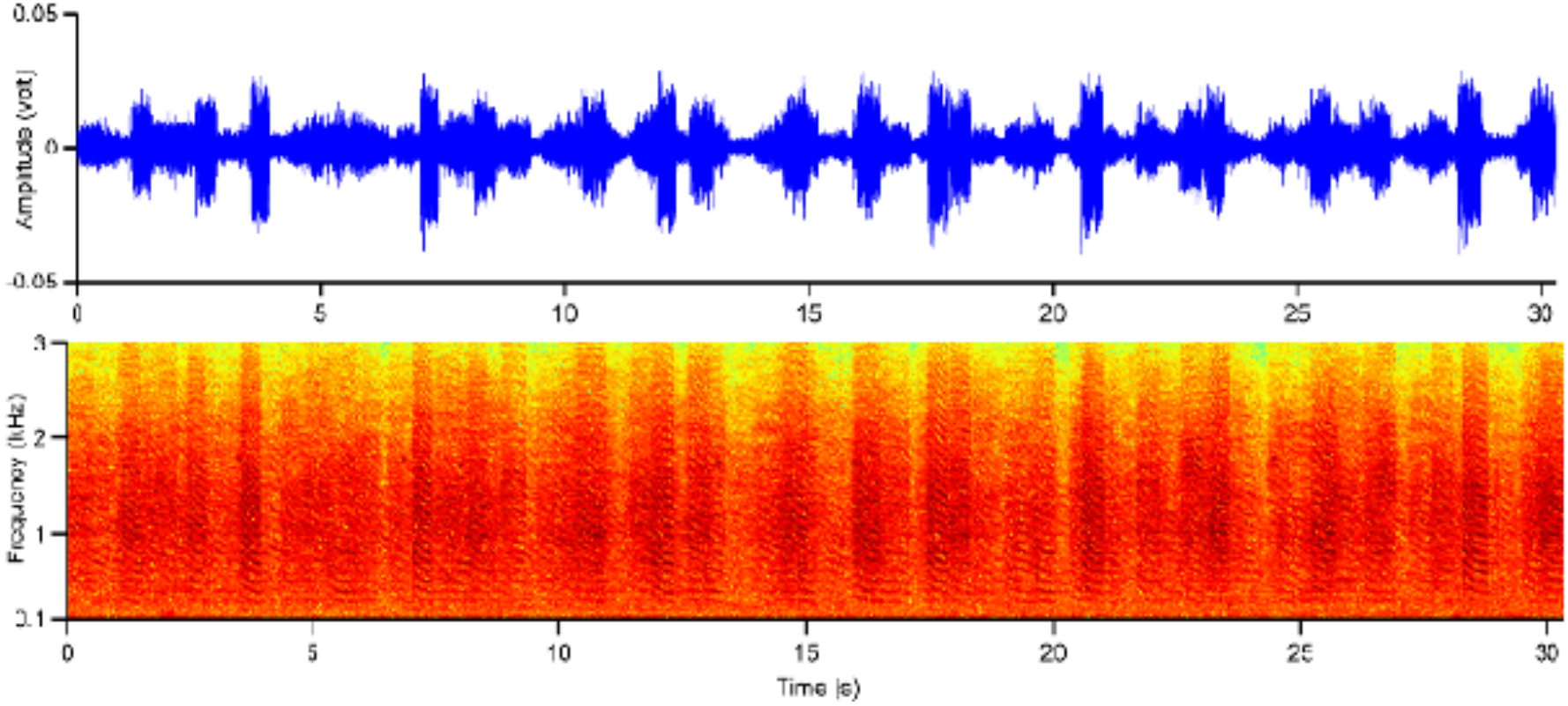
The Time Domain

- Sound is nothing more than ripples in the air
- If we record the air pressure at a point in space between a speaker and a listener and plot it as a function of time we would see a repeating wave-like pattern
- This is exactly what a microphone does
- A time series like this is often referred to as a *signal*

The Frequency Domain

- An alternative way of thinking about sound is as a progression of mixtures of different frequencies of air vibration
- We perceive frequency as pitch
- When different frequencies are related by whole number ratios we usually perceive it as being a note at the lowest frequency in the mix
- The range of frequencies we can hear runs from about 20 to 20,000 Hz
- We can visualize the amount of energy at each frequency over time using a diagram called a *spectrogram*

Time and Frequency Representation of Signals



Spectrogram Synthesis

- Discrete Cosine Transformations
- Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between
- SonicPhoto

Audio Files

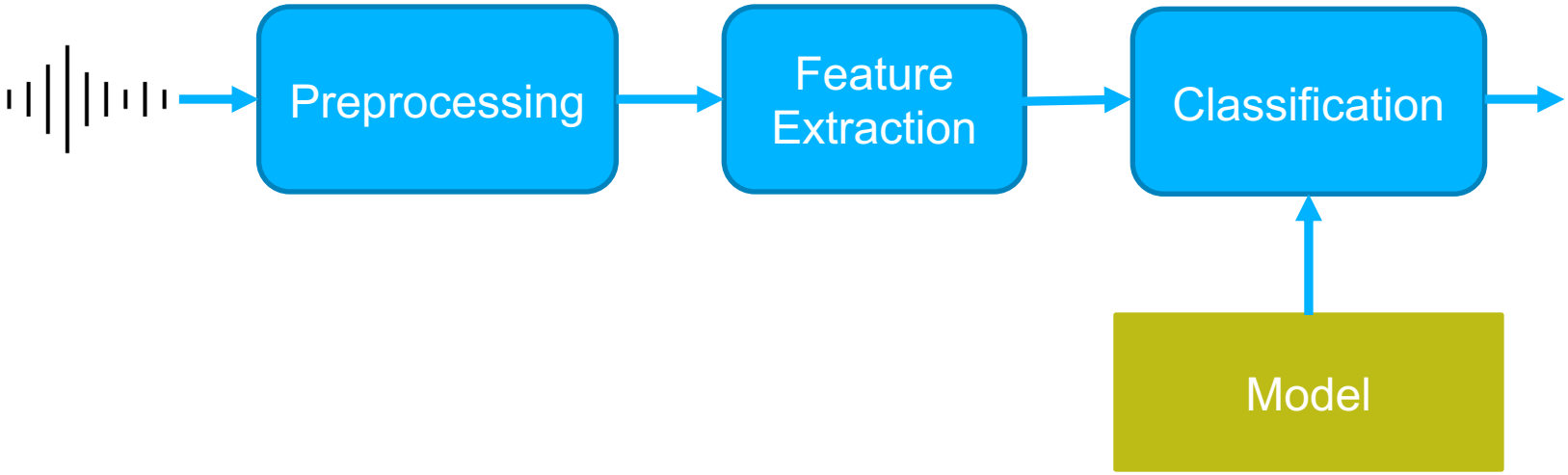
- Audio files contain a time series along with some metadata such as the sample rate
- Each data point is called a *sample*
- Audio files can be monophonic or stereo
- **Nyquist-Shannon sampling rate**: To reliably capture a frequency in a signal you need to sample it at least twice as often as that frequency
- 8,000 samples per second is sufficient for clearly recognizable human speech
- Compressed audio files (e.g. mp3 and 4) encode audio taking advantage of psychoacoustics

MIDI

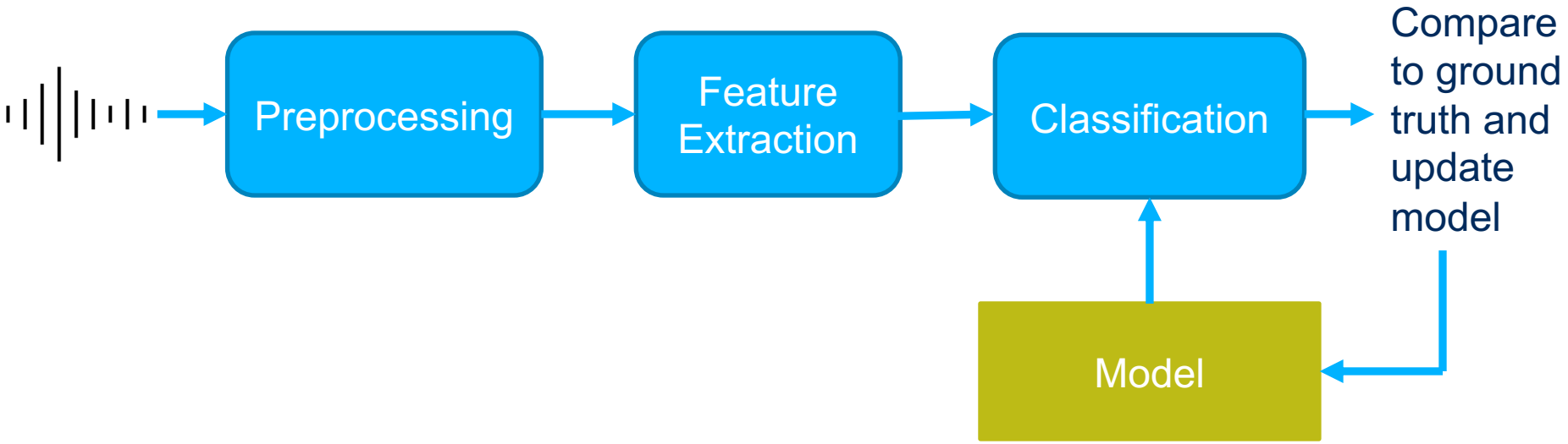
- Musical Instrument Digital Interface
- Encoding for musical events
  - Note on with its velocity and channel
  - Note off
  - Control signal value change
  - Instrument change

Audio Analysis: Unsupervised

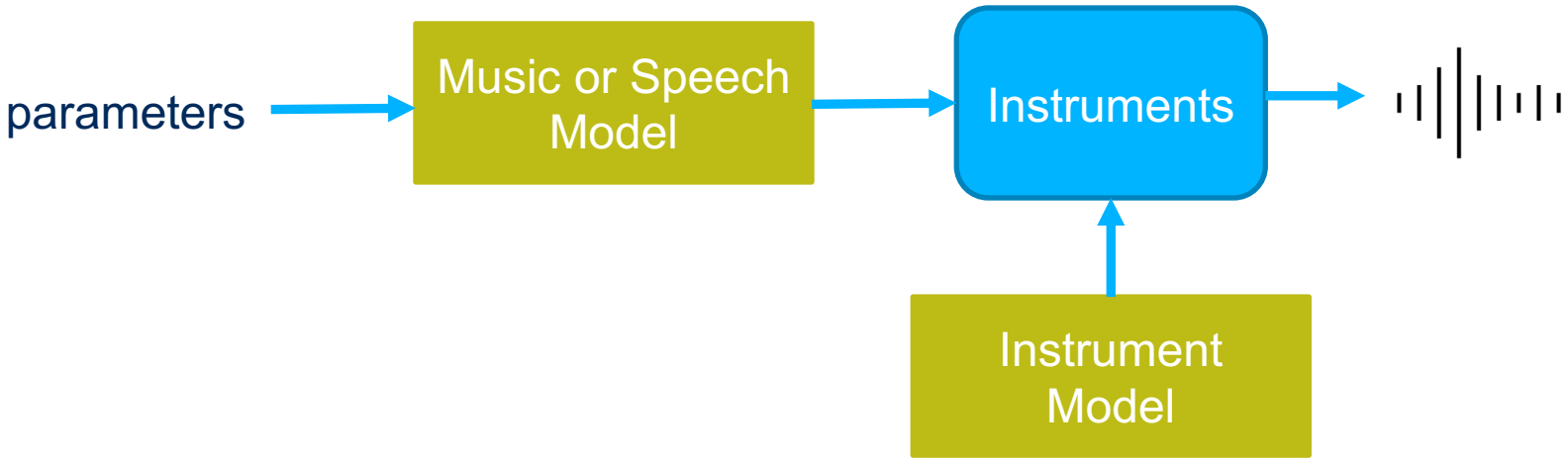
Audio Analysis: Classification



Audio Analysis: Training for Classification



Audio Synthesis



UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

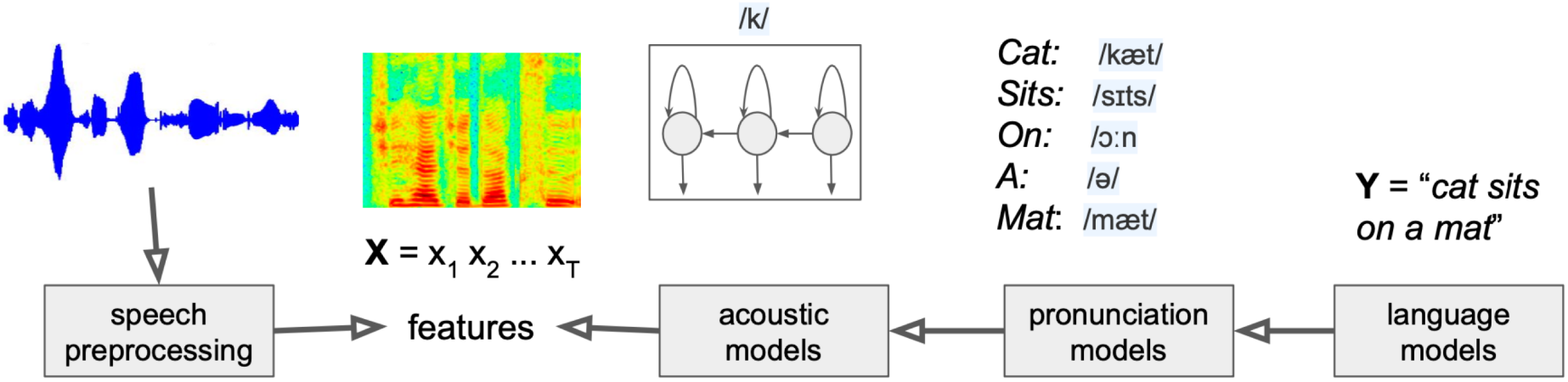
Module 10 – Section 2

Audio Recognition

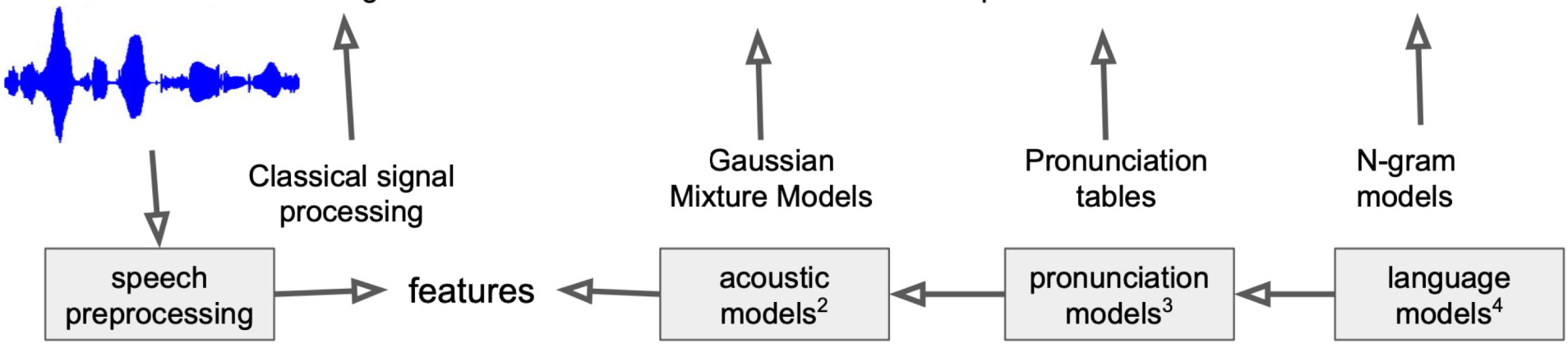
Why Automatic Speech Recognition?

- Speech is a natural interface for human communication
  - Hands free communication.
  - No need to learn new techniques
- Applications are endless
  - Controlling simple devices -- cars, homes, handhelds
  - Interacting with intelligent devices -- chat bots, call center help desks, etc
  - Analysis of call center data

Speech Recognition (classic way)



The Neural Network Invasion

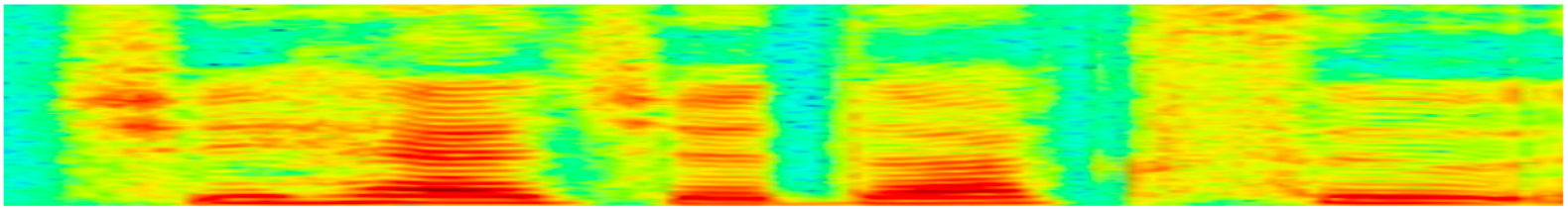


Connectionist Temporal Classification (CTC)

- CTC - a probabilitistic model p(**Y**|**X**), where
  - **X** = $x_1 x_2 \ldots x_T$,
  - **Y**= $y_1 y_2 \ldots y_L$
  - T>= L
- Has a specific structure that is suited for speech

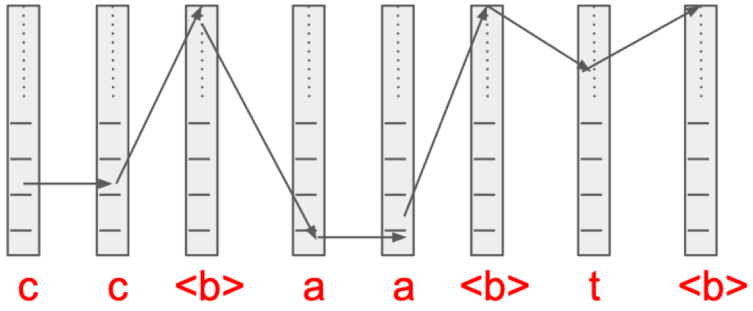
Y = This is a spectrogram

X =



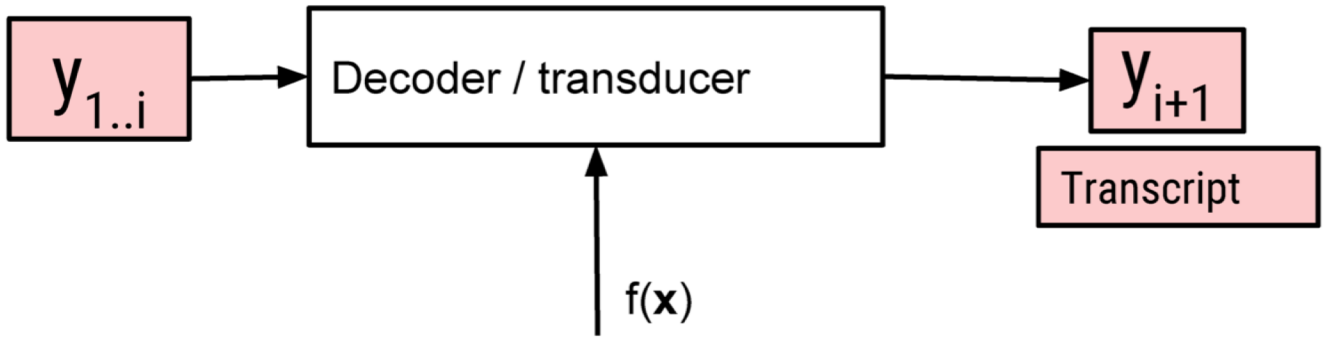
Connectionist Temporal Classification (CTC)

- Repeated tokens are deduplicated
  - cc <b> aa <b> t <b>
- Any original transcript, maps to all possible paths in the duplicated space:
  - cc<b>aa<b>t<b> maps to cat
  - cc<b><b>a<b>t<b> maps to cat
  - ccccc<b>aaaaa<b>tttttt<b> maps to cat
  - ccccc<b>aaaaa<b>tttttt<b> maps to cat
- The score (log probability) of any path is the sum of the scores of individual categories at the different time steps
- The probability of any transcript is the sum of probabilities of all paths that correspond to that transcript

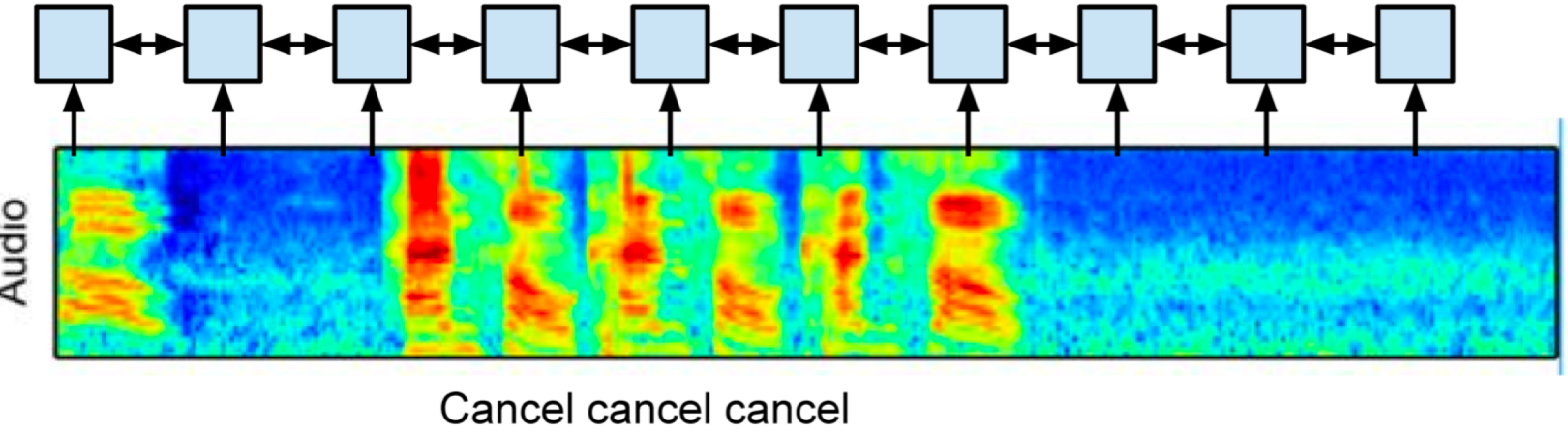


*Because of dynamic programming, it is possible to compute both the log probability p(**Y**|**X**) and its gradient exactly! This gradient can be propagated to neural network whose parameters can then be adjusted by your favorite optimizer!*

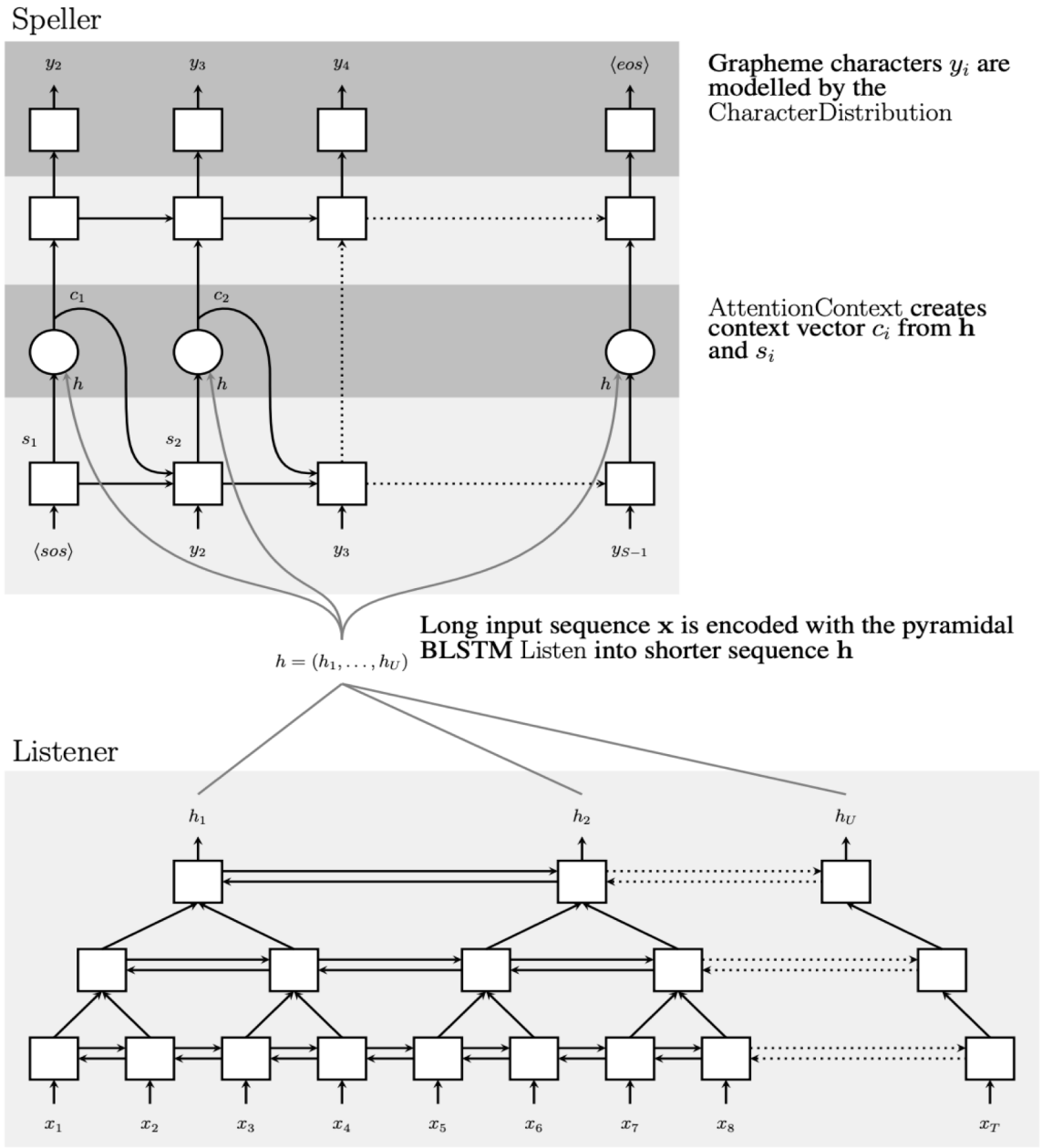
Sequence to Sequence with attention for speech



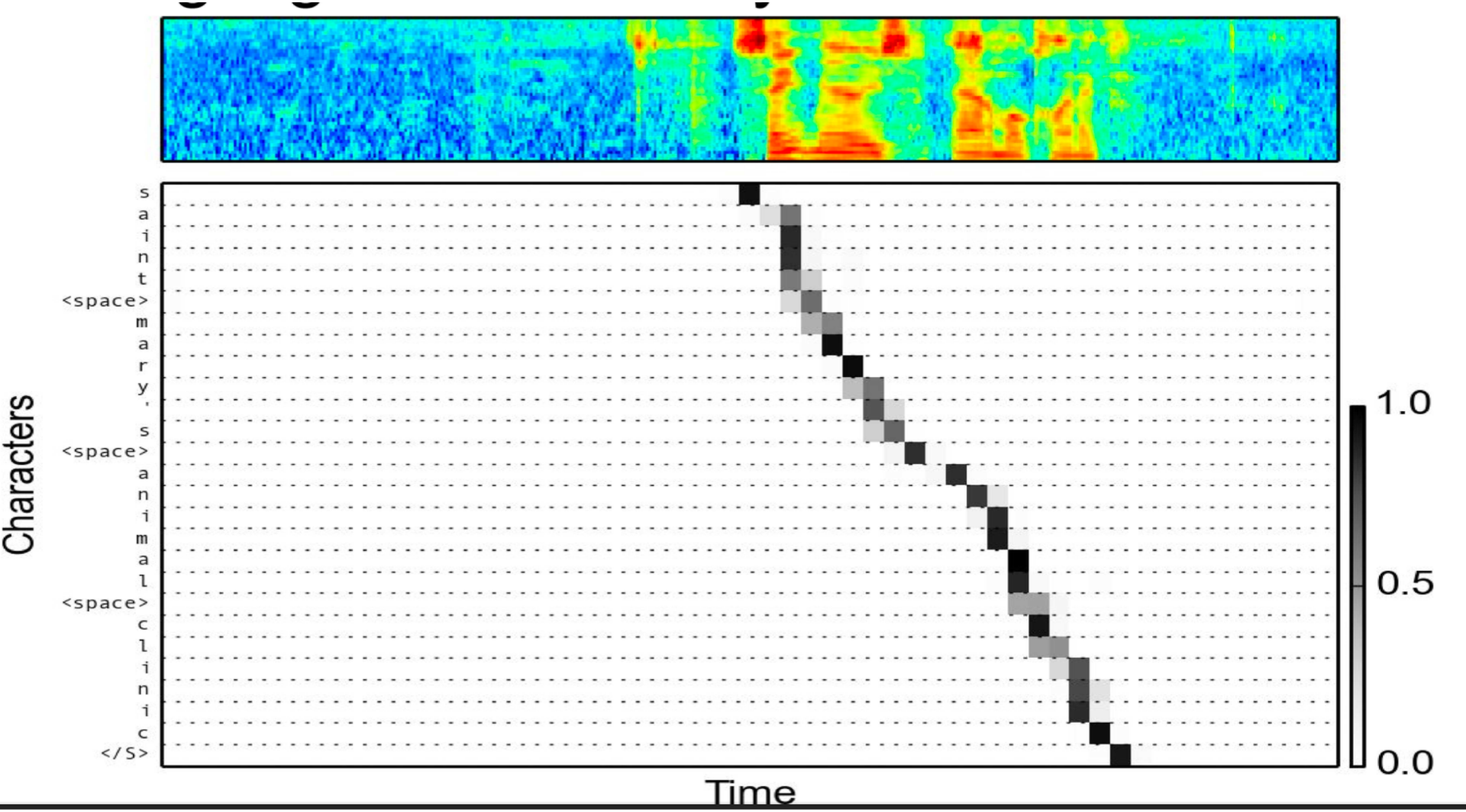
$$p(y_{i+1}|y_{1..i}, x)$$



Listen Attend and Spell (LAS)



Listen Attend and Spell (LAS)



Song Recognition

- Music recognition systems (e.g. SoundHound, Shazam) work by fingerprinting audio files
- They do this by measuring the most prominent frequencies in particular ranges per unit time (a compressed spectrogram)
- When presented a new music snippet they take its fingerprint then do a search for it within the fingerprints of all the songs in the service's library
- Emotion Recognition

Recurrent NN's and the Alignment Problem

- Recurrent NN's are a natural choice for signal processing
- NN's have sufficient capacity to also learn about useful information between phonemes
- We call a snippet of speech we want to recognize as a unit an *utterance*
- An utterance can start at any time and has a definite structure through time so we would like to put a timeframe around it
- But identifying the beginning of a frame, particularly in a noisy environment, can be challenging
- So we must be continually starting new frames, running in parallel, and discard the ones that don't seem to form a coherent utterance

The Universal Solution: Transformers

- Transformers are auto-regressive: they predict the next token given an input sequence and the output so far
- Treat audio features as a sequence of tokens
- Solves alignment problem
- But: complexity of self-attention is quadratic in sequence length

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 10 – Section 3

Speech Synthesis

History and Development of Speech Synthesis

Christian Gottlieb Kratzenstein.
Copper etching based on drawing by Paul Ipsen, 1781.

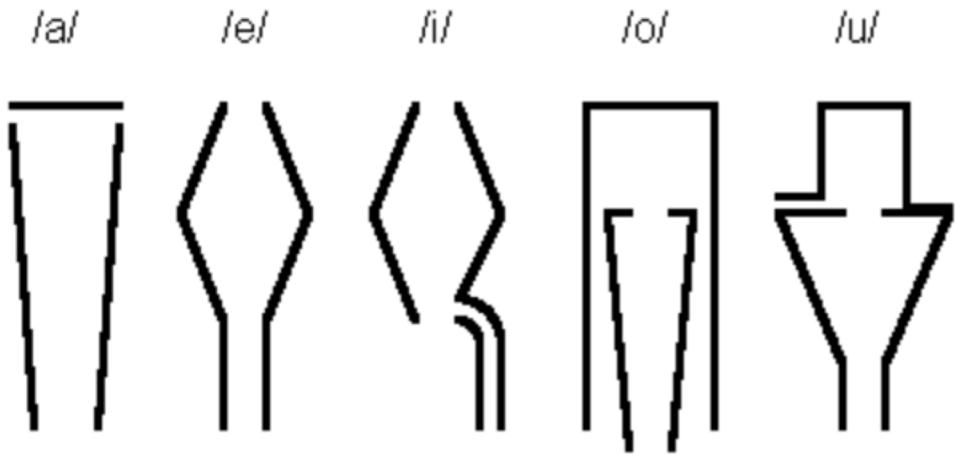
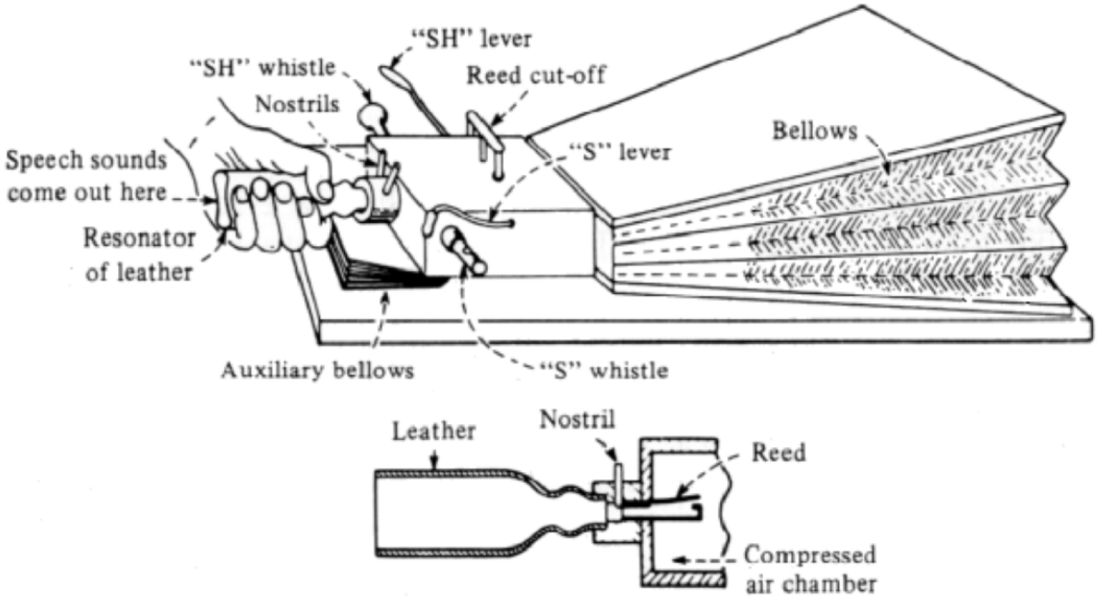


Fig. 2.1. Kratzenstein's resonators (Schroeder 1993).

History and Development of Speech Synthesis

Wolfgang von Kempelen



History and Development of Speech Synthesis

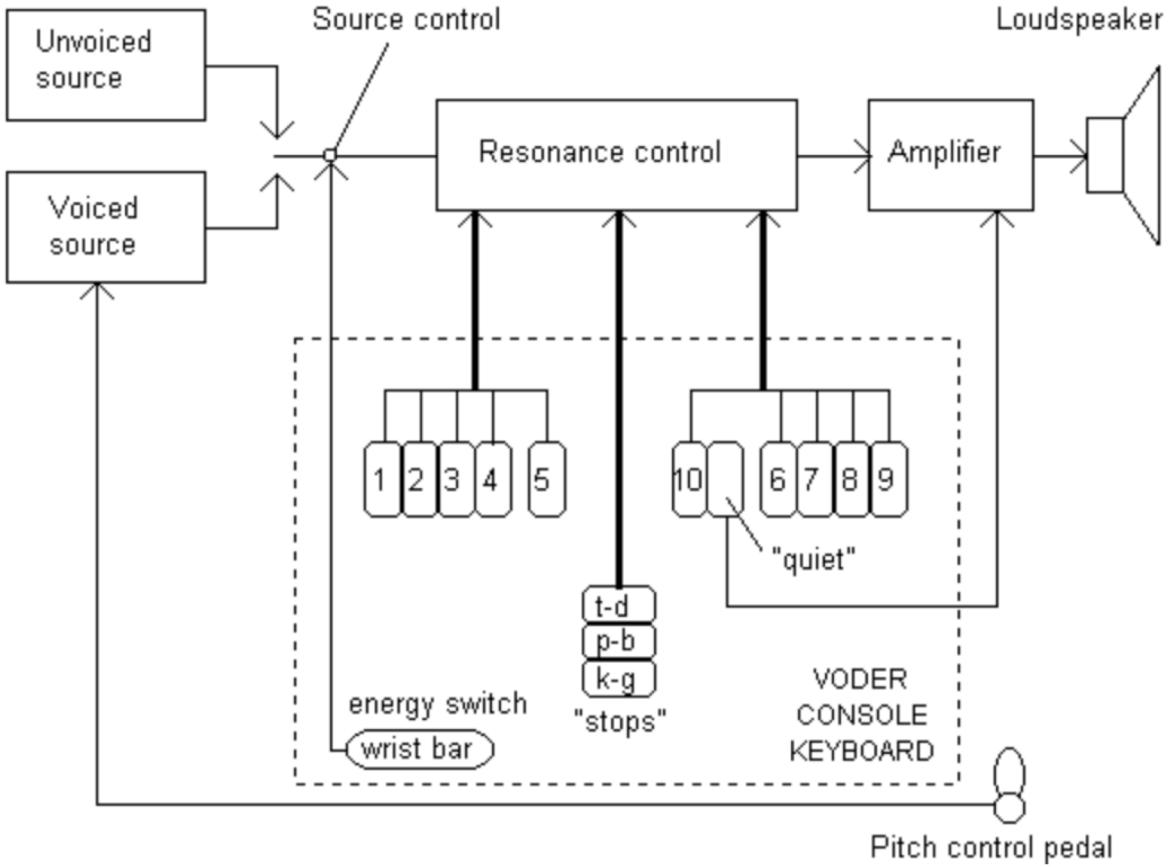
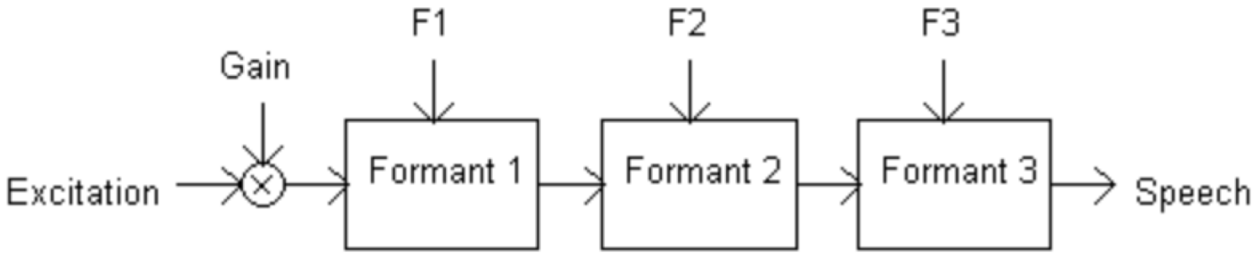


Fig. 2.3. The VODER speech synthesizer (Klatt 1987).

Sample: https://www.youtube.com/watch?v=TsdOej_nC1M

Methods

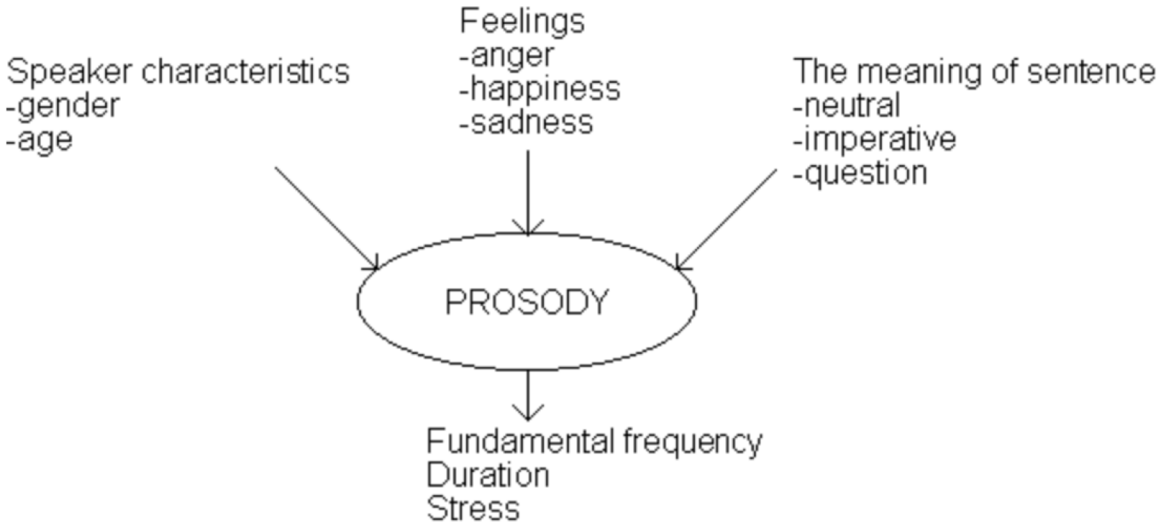
- **Articulatory Synthesis**
  - Articulatory synthesis tries to model the human vocal organs as perfectly as possible
- **Formant Synthesis**
  - Probably the most widely used synthesis method during last decades has been formant synthesis which is based on the source-filter-model



- **Concatenative Synthesis**
  - Connecting prerecorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech.

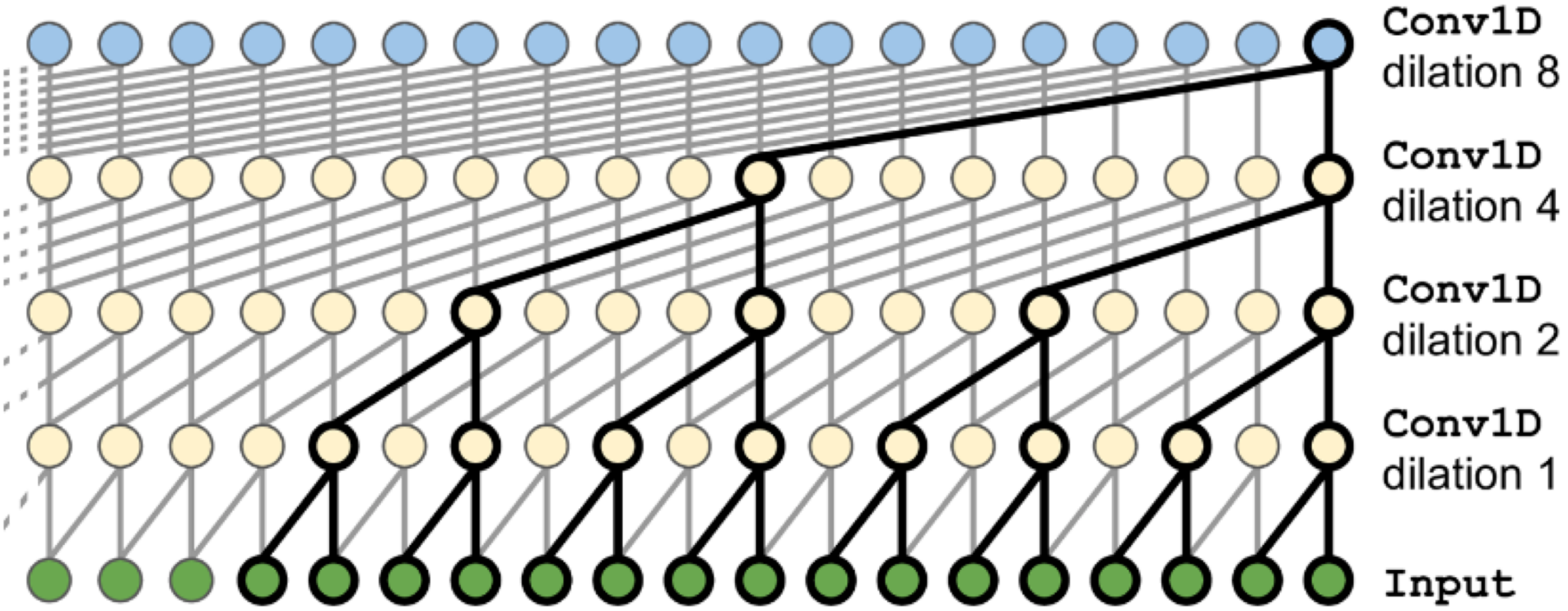
Problems in Speech Synthesis

- *Text preprocessing*
  - For example in English, numeral 243 would be expanded as *two hundred and forty-three* and 1750 as *seventeen-fifty* (if year) or *one-thousand seven-hundred and fifty* (if measure).
- *Pronunciation*
  - The word *lives* is for example pronounced differently in sentences "Three *lives* were lost" and "One *lives* to eat".
- *Prosody*



WaveNet

- DeepMind (2016) developed WaveNet for raw audio generation

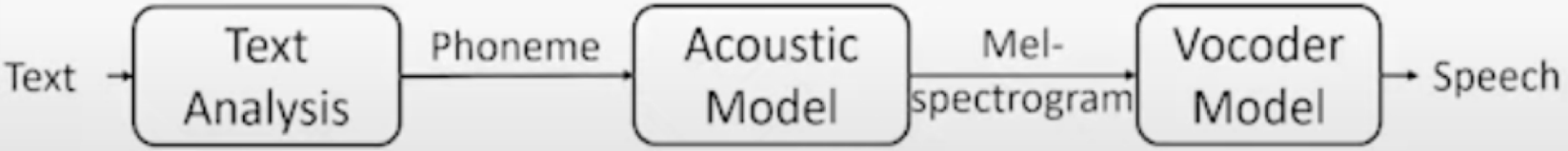


WaveNet Implementation

```
model = keras.models.Sequential()
model.add(keras.layers.InputLayer(input_shape=[None, 1]))
for rate in (1, 2, 4, 8) * 2:
    model.add(keras.layers.Conv1D(filters=20, kernel_size=2, padding="causal",
                                  activation="relu", dilation_rate=rate))
model.add(keras.layers.Conv1D(filters=10, kernel_size=1))
model.compile(loss="mse", optimizer="adam", metrics=[last_time_step_mse])
history = model.fit(X_train, Y_train, epochs=20,
                    validation_data=(X_valid, Y_valid))
```

Text to speech synthesis

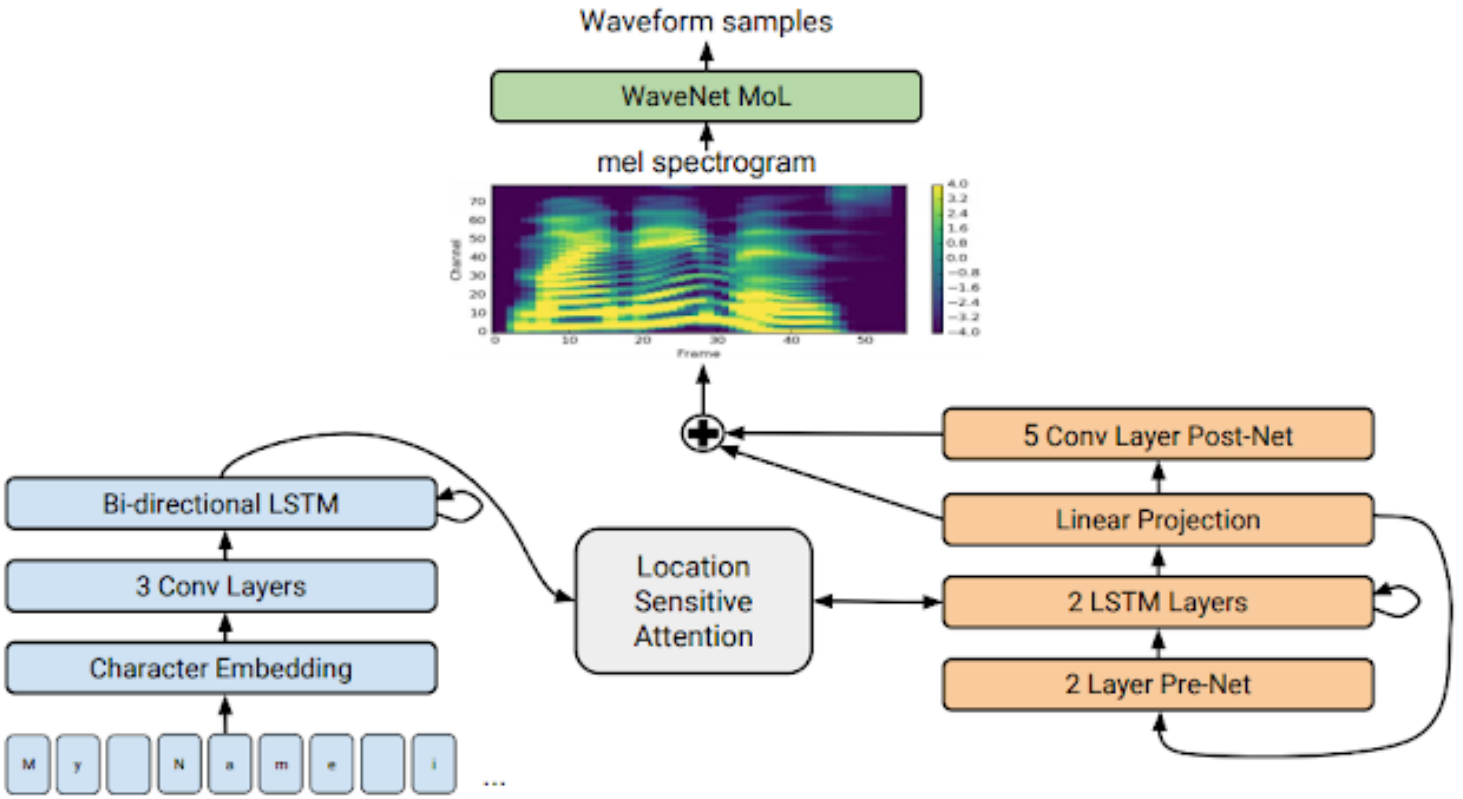
- Neural based end-to-end speech synthesis
  - Text Analysis: text → phoneme (e.g., *Jan.* → *January* → *dʒænjueri)*
    - Text normalization, grapheme-to-phoneme conversion, polyphone disambiguation
  - Acoustic Model: phoneme → mel-spectrogram
    - Tacotron 2, DeepVoice 3, TransformerTTS, FastSpeech 1/2
  - Vocoder: Mel-spectrogram → waveform (neural model)
    - WaveNet, WaveRNN, LPCNet, WaveGlow, MelGAN



Reference: https://www.youtube.com/watch?v=MA8PCvmr8B0

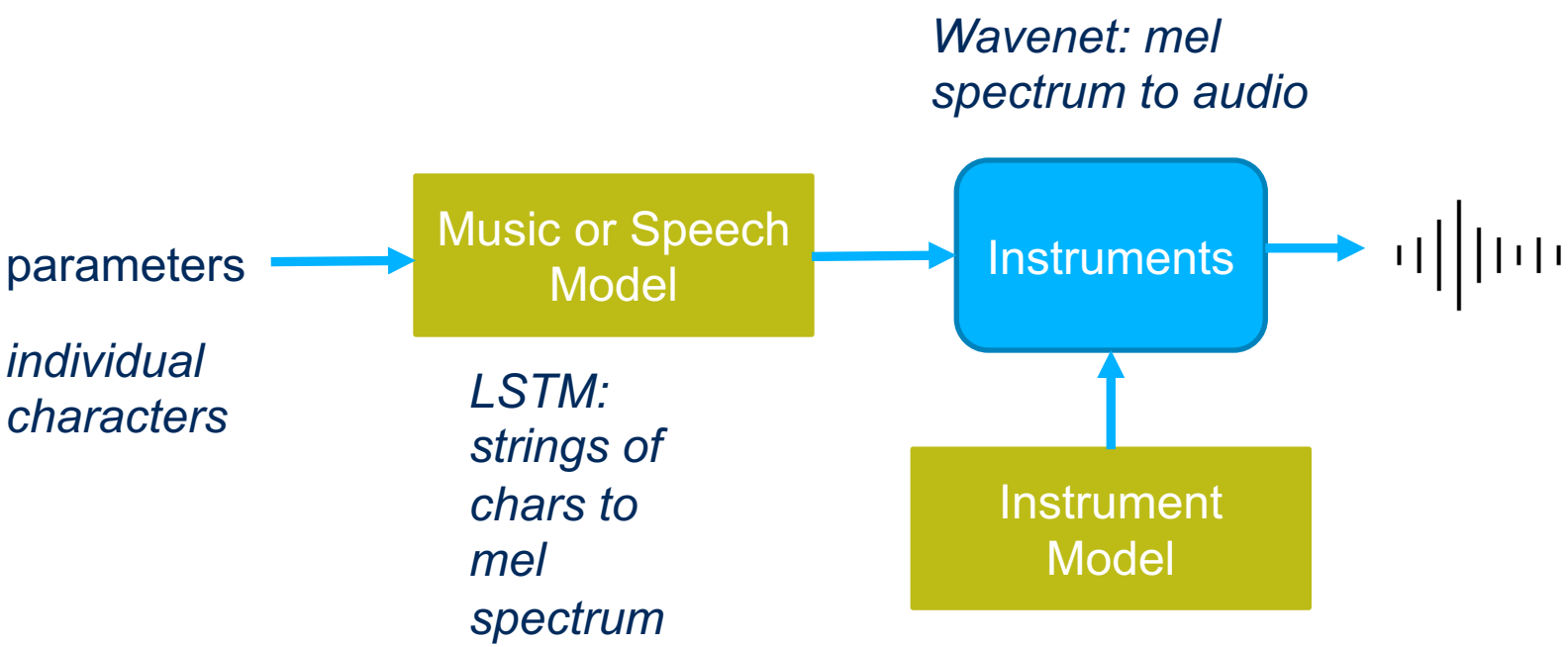
Google Tacotron 2

- https://google.github.io/tacotron/
- https://google.github.io/tacotron/publications/tacotron2/index.html



Source: https://ai.googleblog.com/2017/12/tacotron-2-generating-human-like-speech.html

Tacotron 2's Approach



Other Models

- Deep Voice
- Fast Speech

FastSpeech

- Problems: Previous autoregressive TTS models (Tacotron 2, DeepVoice 3, Transformer TTS) suffer from
  - Slow inference speed: autoregressive mel-spectrogram generation is slow for long sequence;
  - Not robust: words skipping and repeating;
  - Lack of controllability: hard to control the voice speed/prosody in the autoregressive generation

You can call me directly at 4257037344 or my cell 4254447474 or send me a meeting request with all the appropriate information.

Reference: https://www.youtube.com/watch?v=MA8PCvmr8B0

Module 10 – Section 4

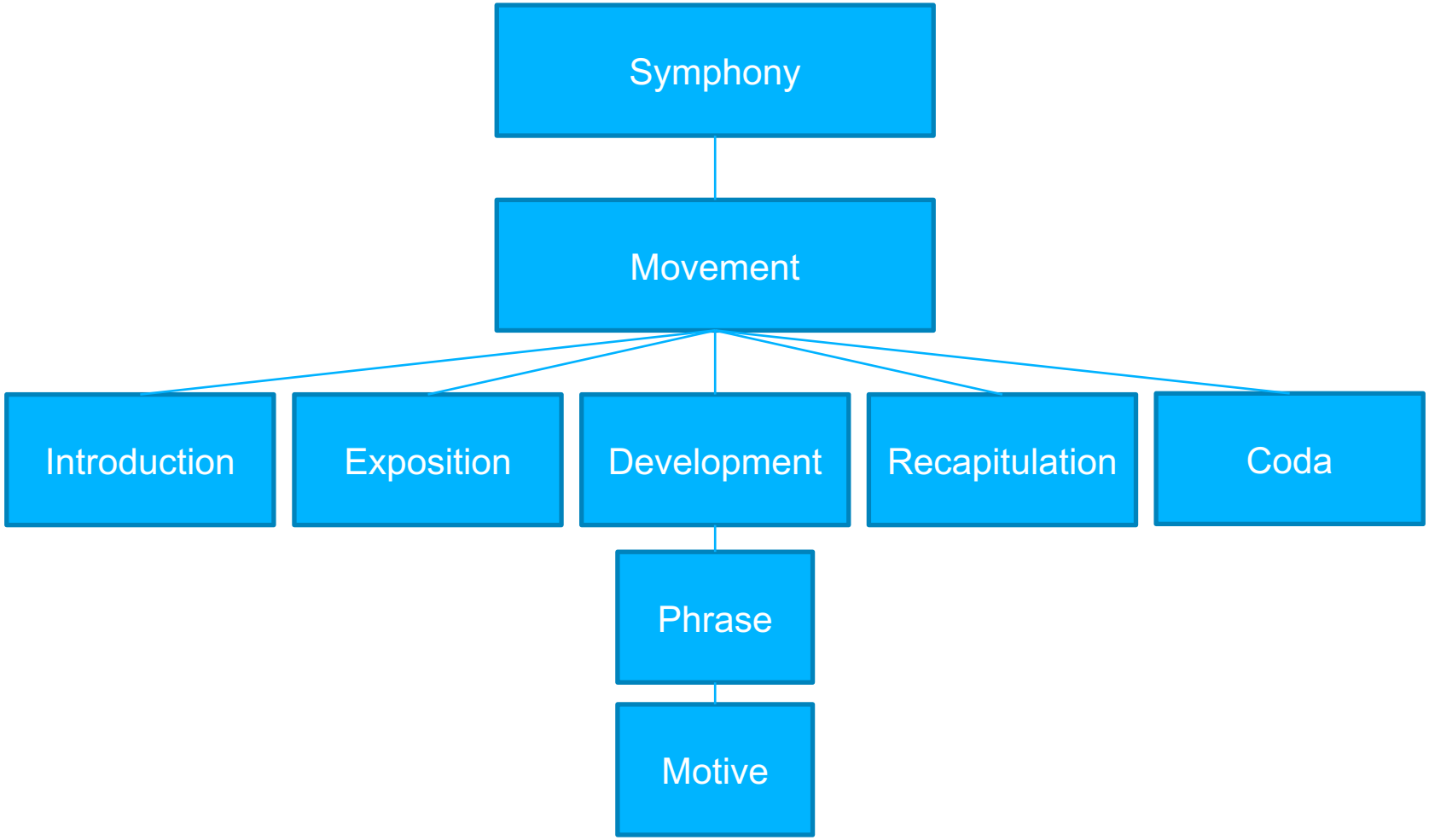
Music Synthesis

Knowledge Engineering vs. Machine Learning

Programmed-in Intelligence

Learned by imitation or exploring

Structure in Music: Form



Structure in Music: Performance

Aleatoric Music and the Dice Game

Illiac Suite (1957)

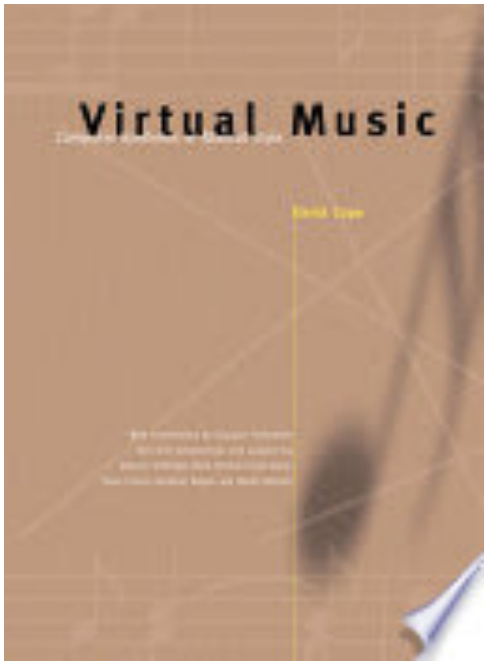
- First substantial computer-composed composition
- https://www.youtube.com/watch?v=n0njBFLQSk8&index=12&list=RDCgG1HipAayU

Cybernetic Composer (1987)

- Charles Ames https://charlesames.net/
- Grammar-based generative system
- Generative grammars can be used for generating fractal-like structures like plants: https://www.youtube.com/watch?v=feNVBEPXAcE
- Cybernetic Composer Latin jazz example: http://charlesames.net/cybernetic-composer/Latin%20Jazz%201.mp3

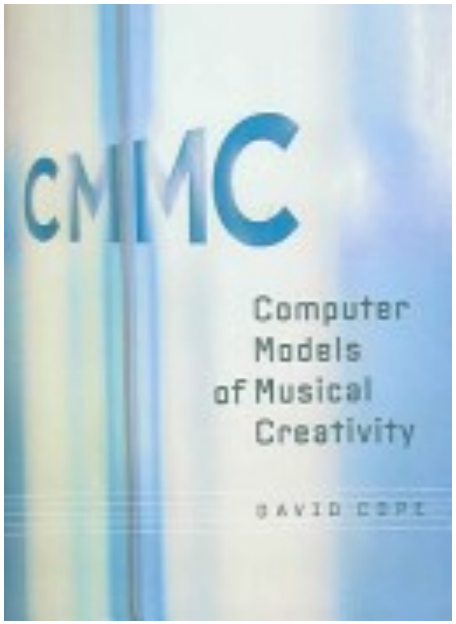
Experiments in Musical Intelligence (1980's-90's)

- David Cope, U. of California @ Santa Cruz
- "Recombinant" Music
- Machine Learning from hand-coded features
- https://www.youtube.com/watch?v=CgG1HipAayU&list=RDCgG1HipAayU&t=128



Emily Howell (2000's)

- Cope's successor to EMI
- Takes feedback from listeners to develop its own style
- https://github.com/HeinrichApfelmus/david-cope-cmmc
- https://www.youtube.com/watch?v=QEjdiE0AoCU

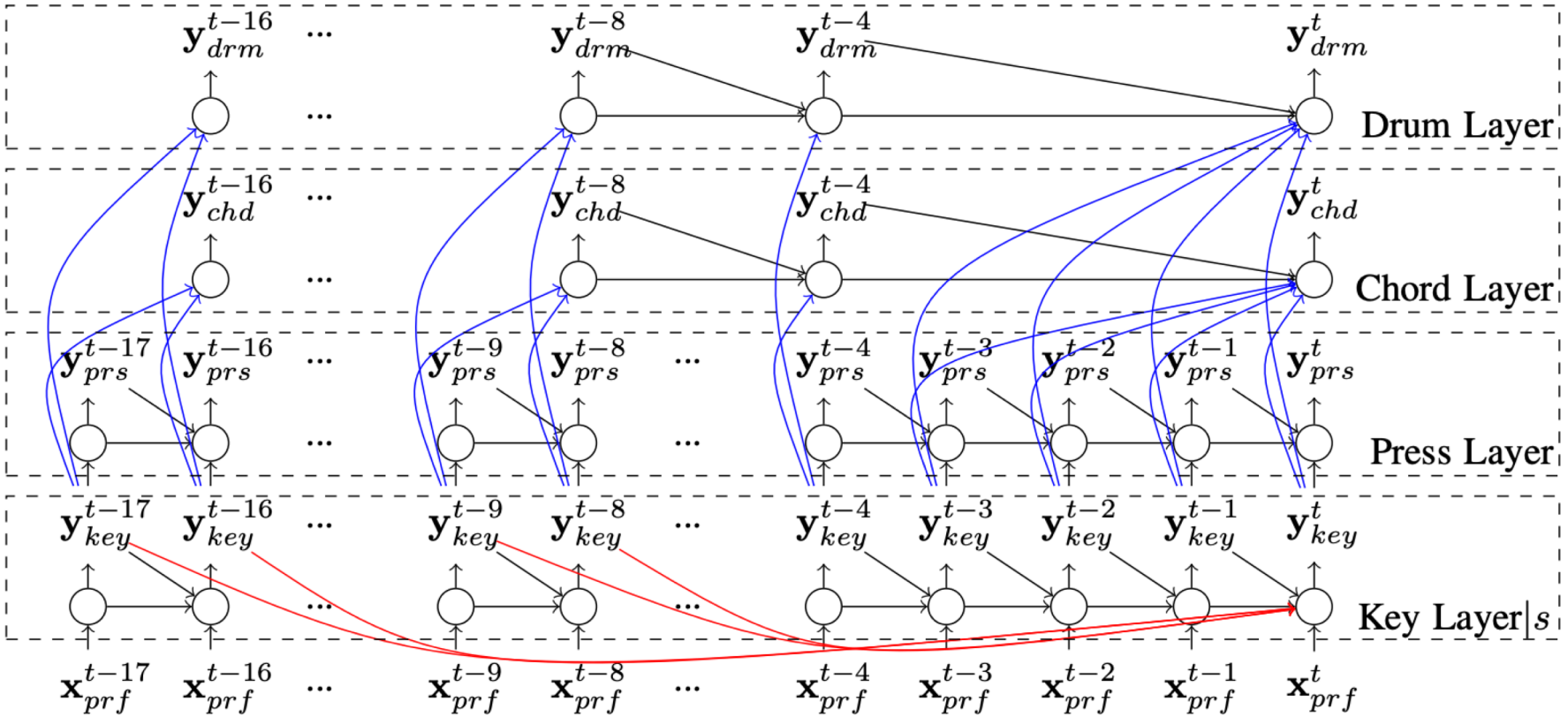


AI Lyrics

- Taryn Southern: https://www.youtube.com/watch?v=XUs6CznN8pw
- These Lyrics Do Not Exist: https://theselyricsdonotexist.com/
- Keyword to Lyrics: https://lyrics.mathigatti.com/
- Lyric Studio: https://lyricstudio.net/
- https://github.com/microsoft/muzic

RNN's

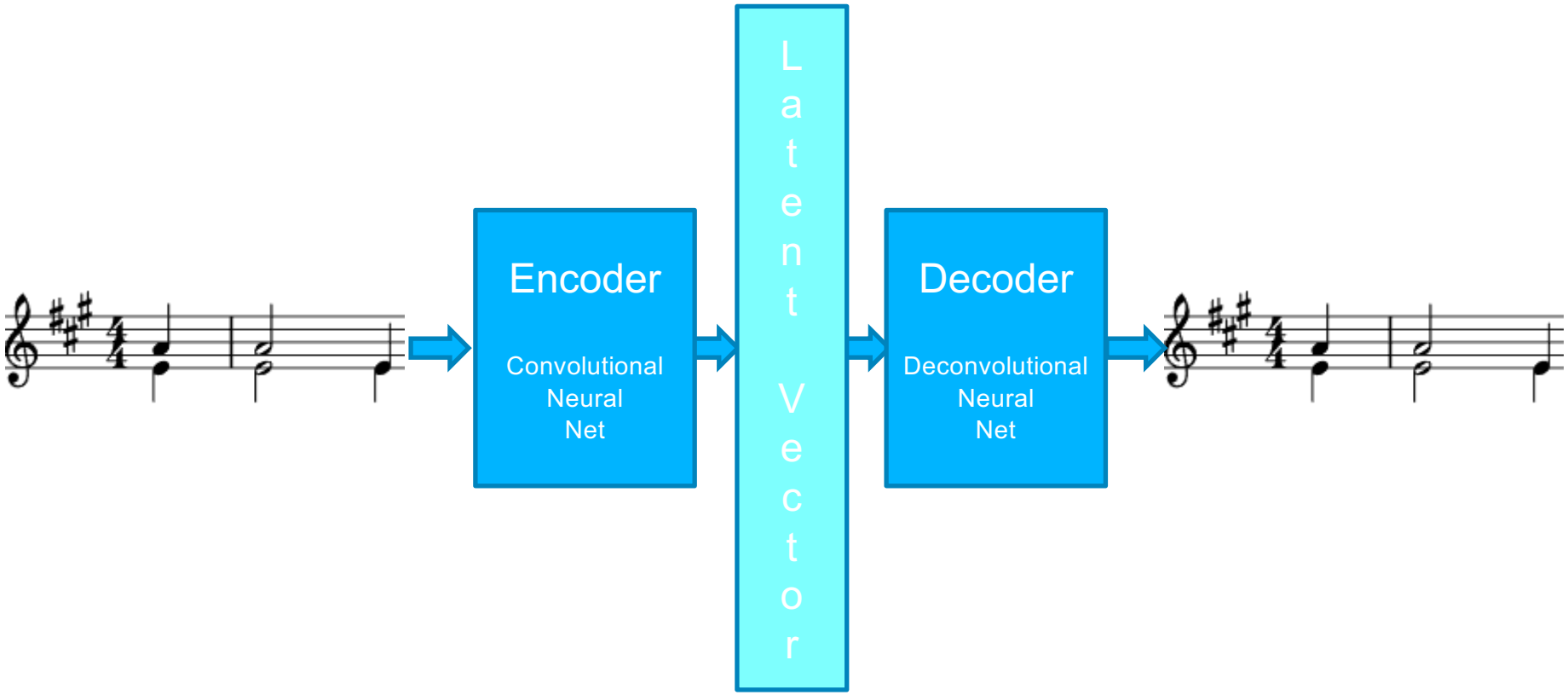
- Song from PI (Chu, Urtasun, Fidler at U. Toronto 2016): http://www.cs.toronto.edu/songfrompi/



Google Magenta

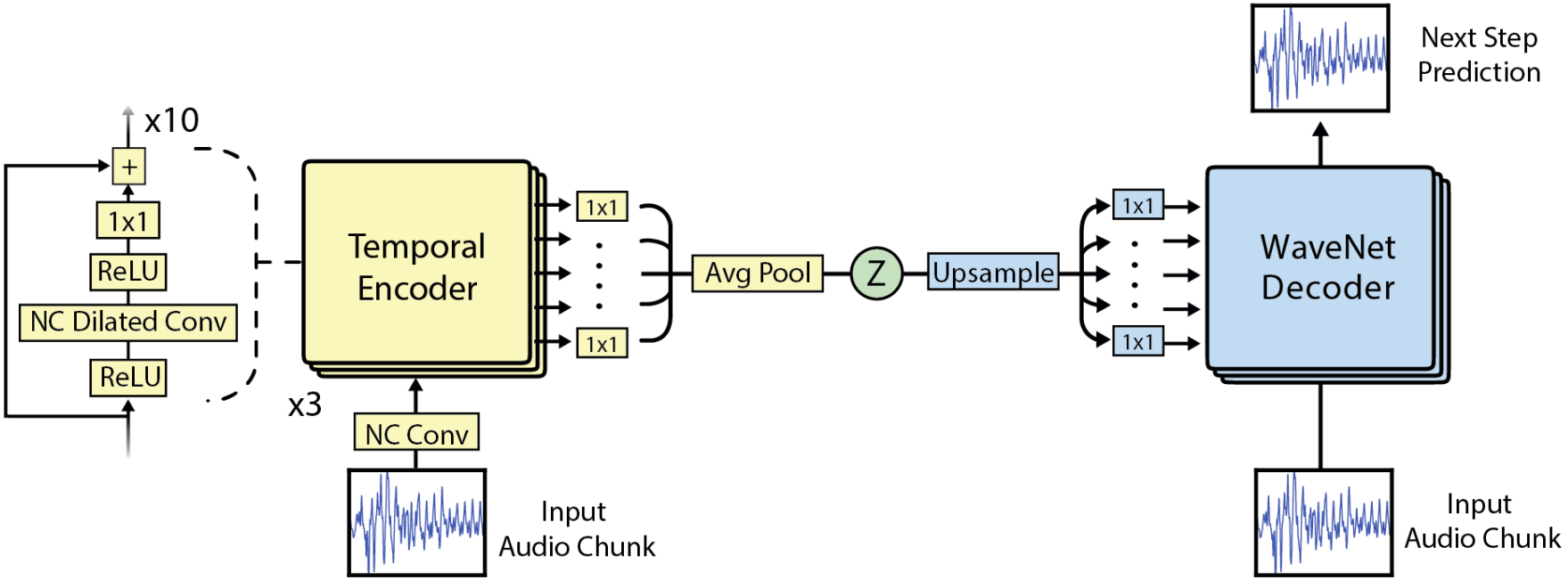
- Beat Blender: https://experiments.withgoogle.com/ai/beat-blender/view/
- NSynth: https://magenta.tensorflow.org/nsynth
- NSynth SoundMaker: https://experiments.withgoogle.com/ai/sound-maker/view/
-

Modeling Music Using Latent Vectors



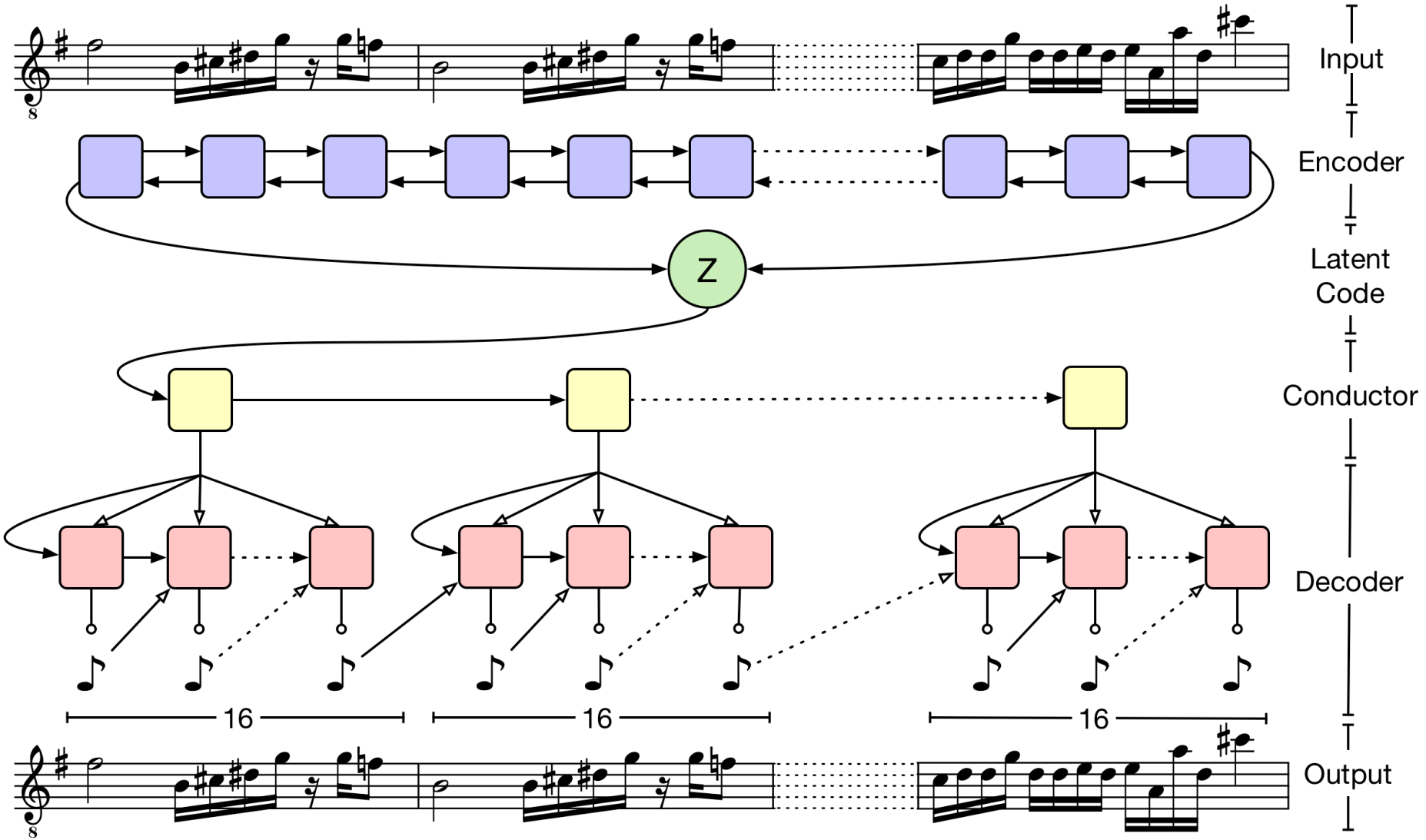
Variational Autoencoder

NSynth: Neural Audio Synthesis



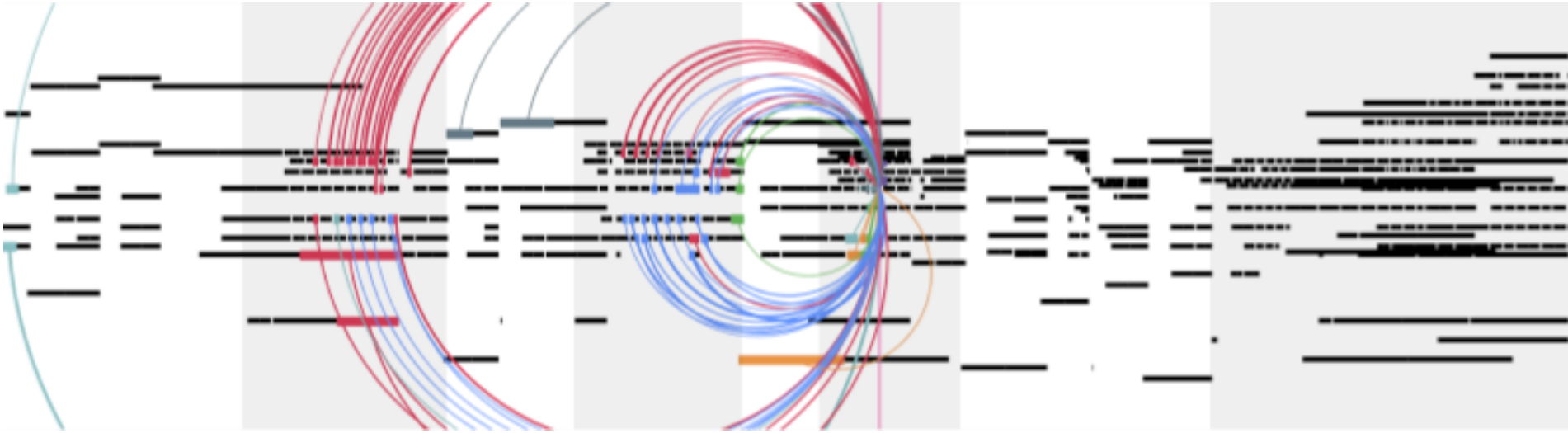
Reference: https://magenta.tensorflow.org/nsynth

MusicVAE



Reference: https://magenta.tensorflow.org/music-vae

Music Transformer

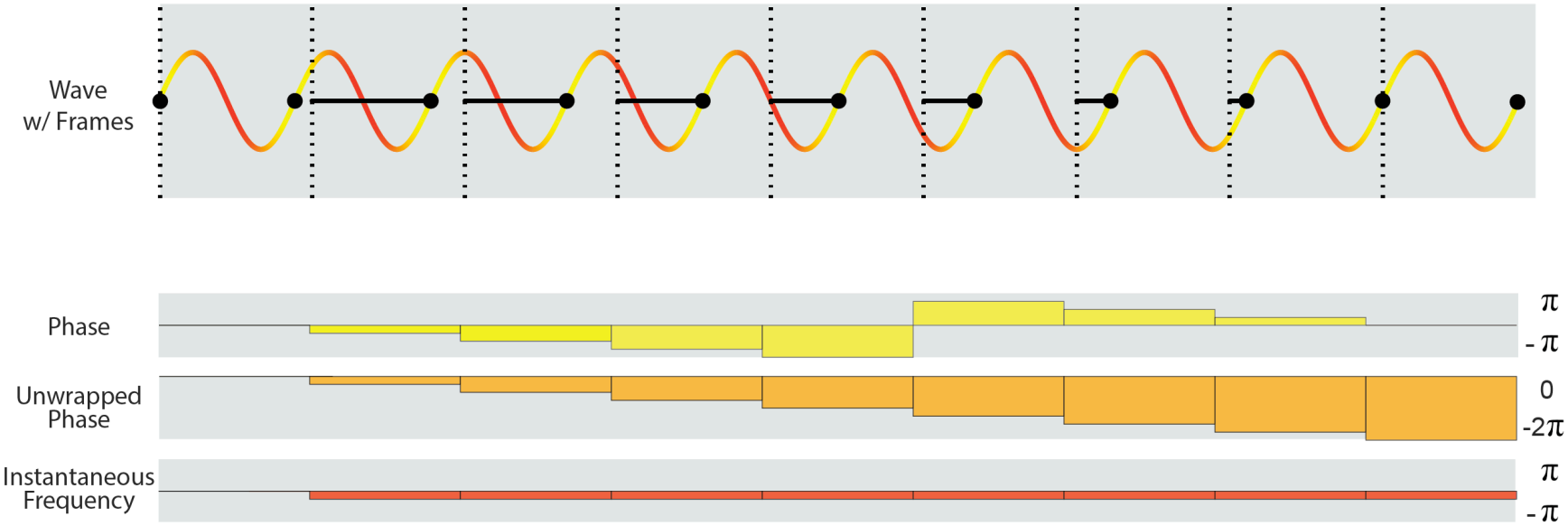


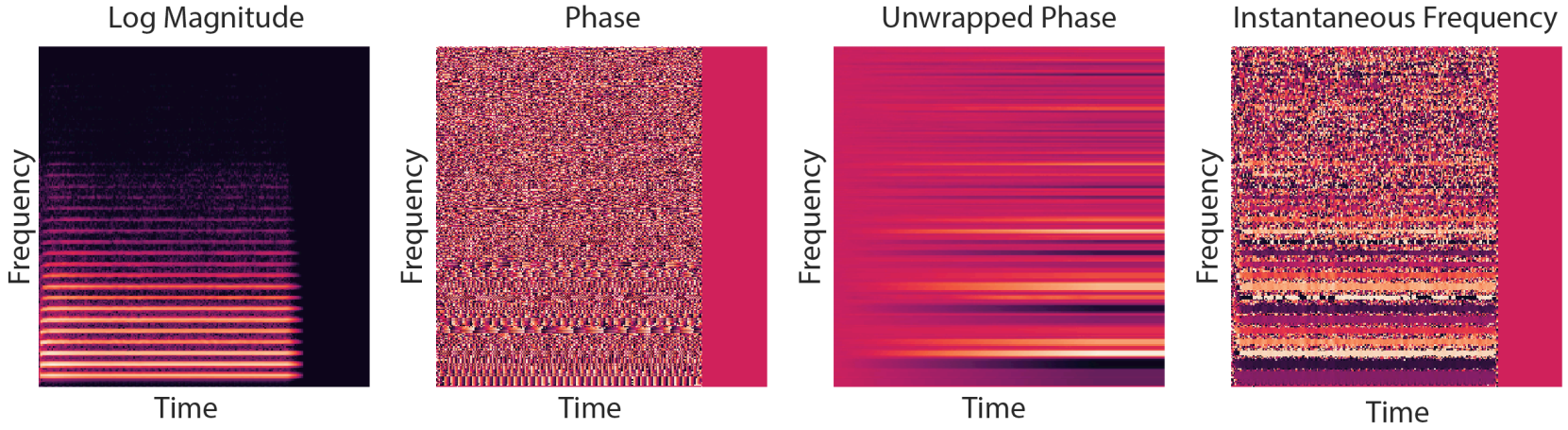
Reference: https://magenta.tensorflow.org/music-transformer

Music Transformers

- OpenAI MuseNet: https://openai.com/blog/musenet/
- Music Autobot: https://musicautobot.com/
- OpenAI Jukebox: https://openai.com/blog/jukebox/
- https://magenta.tensorflow.org/piano-transformer
- https://sites.research.google/tonetransfer

GANSynth: Making music with GANs

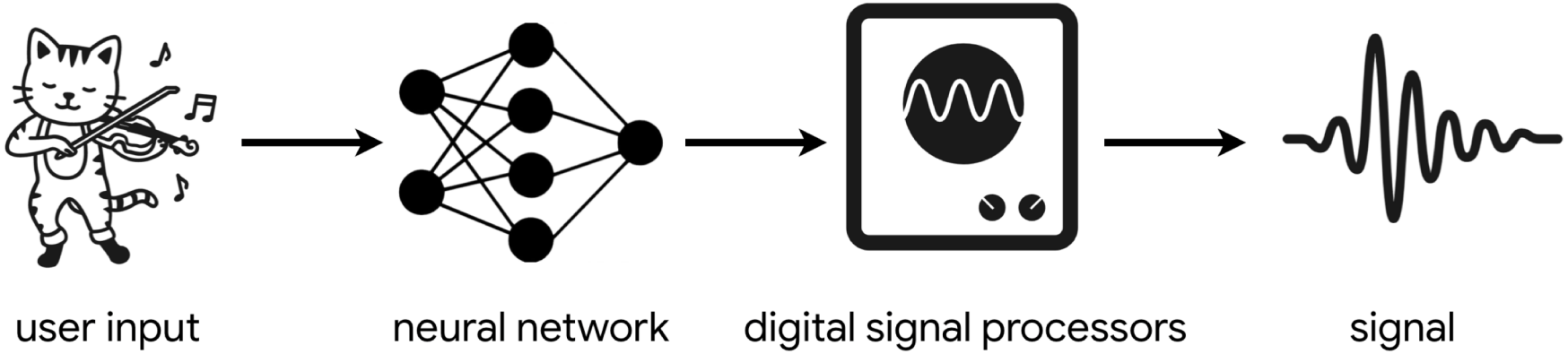




Reference: https://magenta.tensorflow.org/gansynth

Magenta Differentiable Digital Signal Processing

- https://magenta.tensorflow.org/ddsp



UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 10 – Section 5

Resources and Wrap-up

Resources

- Speech To Text Demo: https://dictation.io/
- Cocktail party effect: https://en.wikipedia.org/wiki/Cocktail_party_effect
- Blind Audio: https://cnl.salk.edu/~tewon/Blind/blind_audio.html
- Speech2Face: https://speech2face.github.io/
- Speech2Face Repo: https://github.com/imatge-upc/speech2face
- Spoken Language Processing: http://www.cs.toronto.edu/~frank/csc2518/
- Revealing Invisible Changes In The World: https://www.youtube.com/watch?v=e9ASH8IBJ2U
- Video Magnification: http://people.csail.mit.edu/mrub/vidmag/#code
- Discrete Cosine Transformations: http://datagenetics.com/blog/november32012/index.html
- Speech Processing for Machine Learning: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html
- Torch Audio: https://github.com/pytorch/audio
- Torch Audio Tutorial: https://pytorch.org/tutorials/beginner/audio_preprocessing_tutorial.html

Resources (cont'd)

- Speech Recognition — Deep Speech, CTC, Listen, Attend, and Spell: https://medium.com/@jonathan_hui/speech-recognition-deep-speech-ctc-listen-attend-and-spell-d05e940e9ed1
- End to End Models for speech recognition: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf
- Correction of Automatic Speech Recognition with Transformer Sequence-to-sequence Model: https://arxiv.org/abs/1910.10697
- Streaming Automatic Speech Recognition with the Transformer Model: https://arxiv.org/pdf/2001.02674.pdf

Resources (cont'd)

- Dice Game: https://mozart.vician.cz/
- Magenta Publication: https://ai.google/research/pubs/?collection=magenta
- https://experiments.withgoogle.com/ai/sound-maker/view/
- A Hierarchical Recurrent Neural Network for Symbolic Melody Generation: https://arxiv.org/pdf/1712.05274.pdf
- Music VAE: https://magenta.tensorflow.org/music-vae
- Transformers and Self-Attention For Generative Models: http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture14-transformers.pdf
- Music Transformer: https://magenta.tensorflow.org/music-transformer
- Generating Piano Music with Transformer: https://colab.research.google.com/notebooks/magenta/piano_transformer/piano_transformer.ipynb
- GANSynth: https://magenta.tensorflow.org/gansynth

Resources (cont'd)

- History of text-to-speech: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/chap2.html
- Speech Synthesis Tutorial: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/contents.html
- The Voder: 1939, the worlds first electronic voice synthesizer:https://www.youtube.com/watch?v=TsdOej_nC1M
- -Dennis Klatt's development of speech synthesizers 1939-1985 voice demonstration: https://www.youtube.com/watch?v=huq2TSV99hI
- WaveNet: A generative model for raw audio: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio
- WaveNet Demo: https://cloud.google.com/text-to-speech
- Tacotron: https://google.github.io/tacotron/
- Tacotron2 implementation in PyTorch: https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/
- Expressive Speech Synthesis: https://ai.googleblog.com/2018/03/expressive-speech-synthesis-with.html

Resources (cont'd)

- https://medium.com/@saxenauts/speech-synthesis-techniques-using-deep-neural-networks-38699e943861
- http://greenteapress.com/wp/think-dsp/
- https://github.com/AllenDowney/ThinkDSP
- https://musicinformationretrieval.com/
- https://librosa.github.io/
- https://github.com/tyiannak/pyAudioAnalysis
- https://deepmind.com/blog/wavenet-generative-model-raw-audio/
- https://www.audacityteam.org/
- http://deepsound.io/
- http://www.gitxiv.com/posts/kQhs4N6rRohXDcp89/samplernn-an-unconditional-end-to-end-neural-audio
- https://magenta.tensorflow.org/
- https://github.com/ybayle/awesome-deep-learning-music
- https://research.mozilla.org/machine-learning
- https://github.com/robmsmt/KerasDeepSpeech
- https://github.com/ShankHarinath/DeepSpeech2-Keras
- TimbreTron: https://www.youtube.com/watch?v=YQAupr7JxNY
- https://arxiv.org/pdf/2011.06801.pdf

Next Class

- Project Presentations!

Follow us on social

Join the conversation with us online:

facebook.com/uoftscs

@uoftscs

linkedin.com/company/university-of-toronto-school-of-continuing-studies

@uoftscs

Any questions?

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Thank You

Thank you for choosing the University of Toronto School of Continuing Studies