WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Benjamin Beilharz, 2019/12/12 – Recent Advances in Seq2Seq Learning
Heidelberg University, Department of Computational Linguistics

OUTLINE

Task Introduction – What's speech?

Preliminaries

WaveNet

Experiments & Results

Conclusion

Discussion

TASK INTRODUCTION

TTS BACKGROUND – OR HOW COMPUTERS LEARNED TO SPEAK

- TTS synthesis: render natural sounding speech given a text
- Sequence mapping problem: text → speech signals (time series)
- Typical TTS pipeline:

NLP
- Sentence/Word Segmentation
- Text Normalization
- POS-Tagging

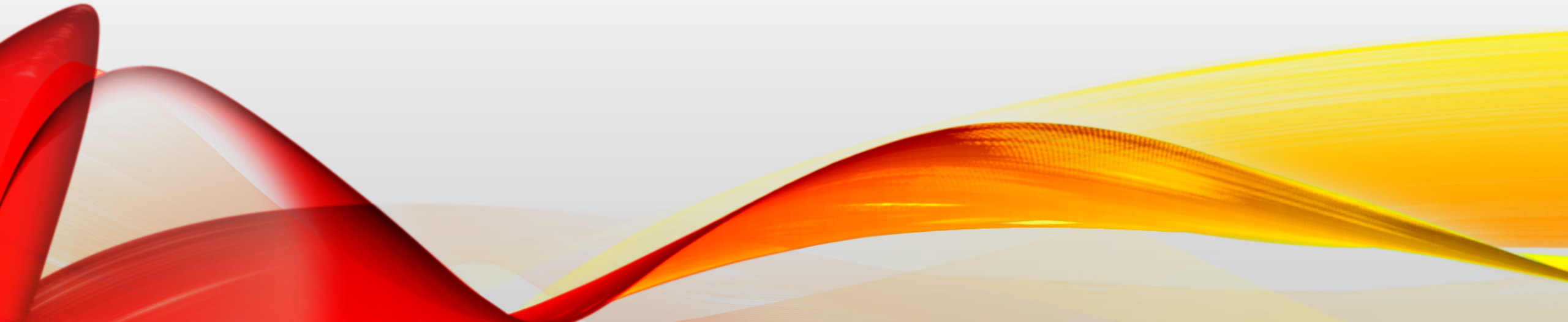
G2P
- Input: Word
- Output: Phoneme sequence

SS
- Input: Phoneme sequence
- Output: synthesized speech waveform

TTS BACKGROUND – OR HOW COMPUTERS LEARNED TO SPEAK

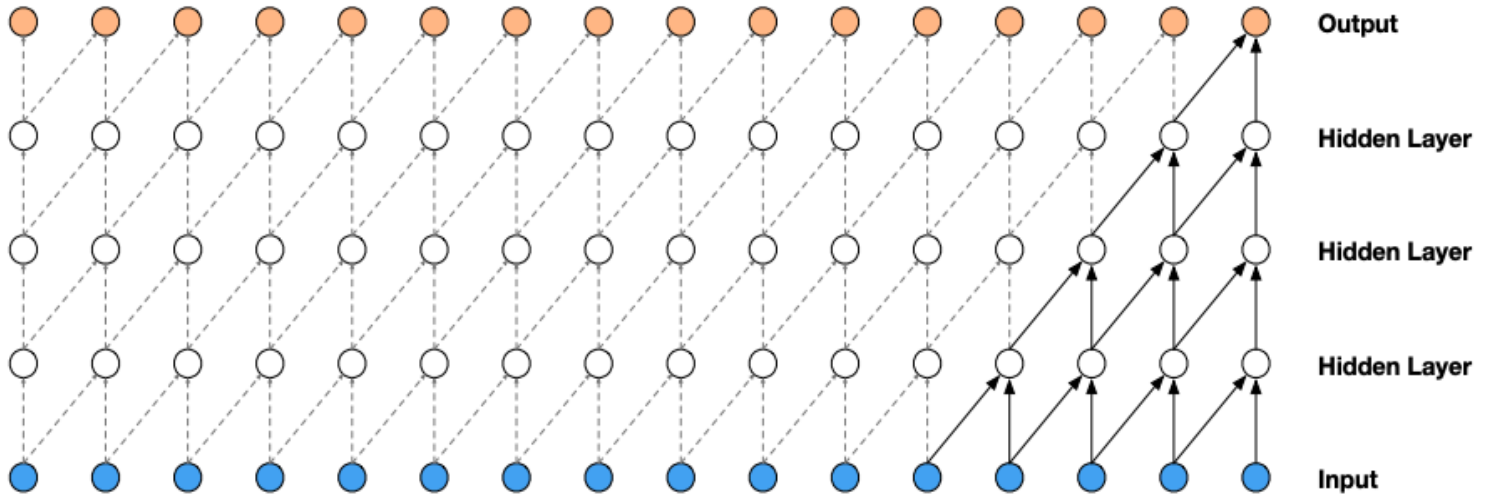
- Two approaches of speech synthesis:
  - Non-parametric
  - Parametric

PRELIMINARIES



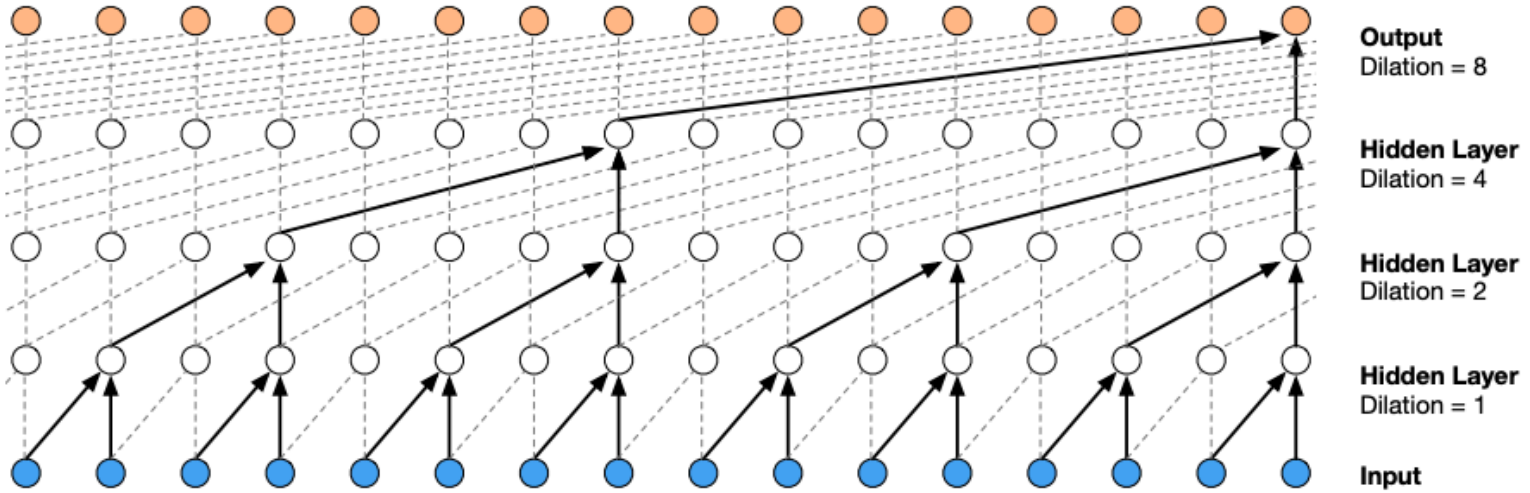
CAUSAL CONVOLUTIONS

- Equal to masked convolutions
- Does not look into the future of the sequence
- Element-wise multiplication of mask with kernel
- Problem: our receptive field is quite small



DILATED CAUSAL CONVOLUTIONS

- Dilation of 1 is equal to filter size
- Dilation scale is doubled for each layer (up to a factor of 512)
- receptive field grows exponential
- Total receptive field: filter size times dilation factor



SOFTMAX COMPLEXITY PROBLEM & μ-TRANSFORMATION

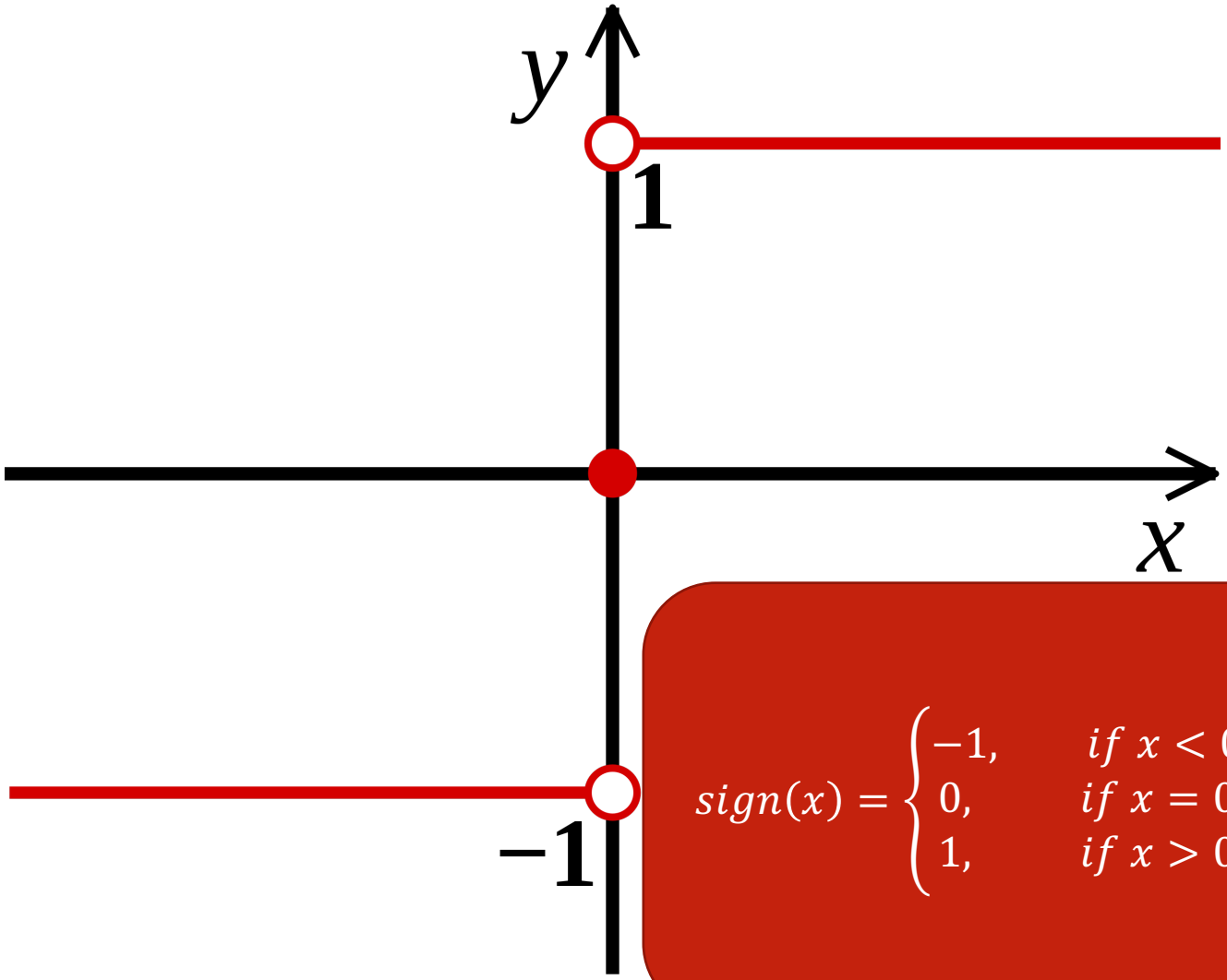
- Raw audio is saved in 16bit integers := 65536 probabilities each timestep
- Is solved by μ-law companding transformation
- μ-law is used to reduce the dynamic range of audio signals

Reminder: x_t is not between -1 and 1, the output of sign is

$$t) = sign(x_t)\ \frac{\ln(1+\mu|x_t|)}{\ln(1+\mu)},$$

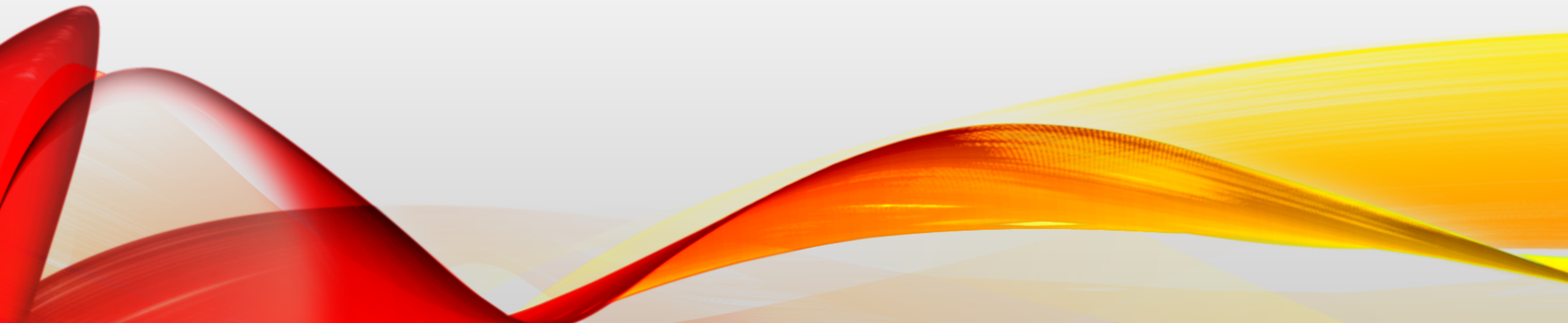
where $sign(x_t) \in \{-1, 0, 1\}$, $\mu = 255$

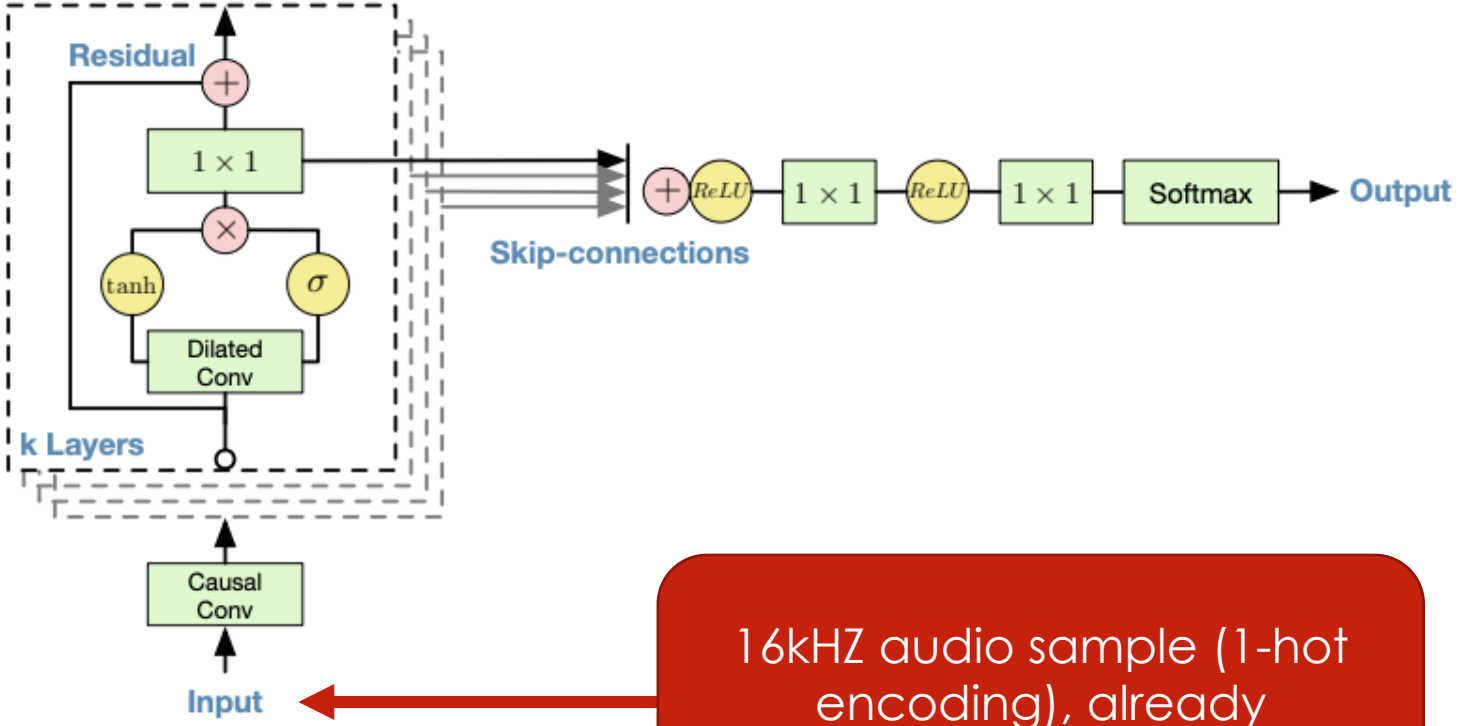
SIGNUM FUNCTION



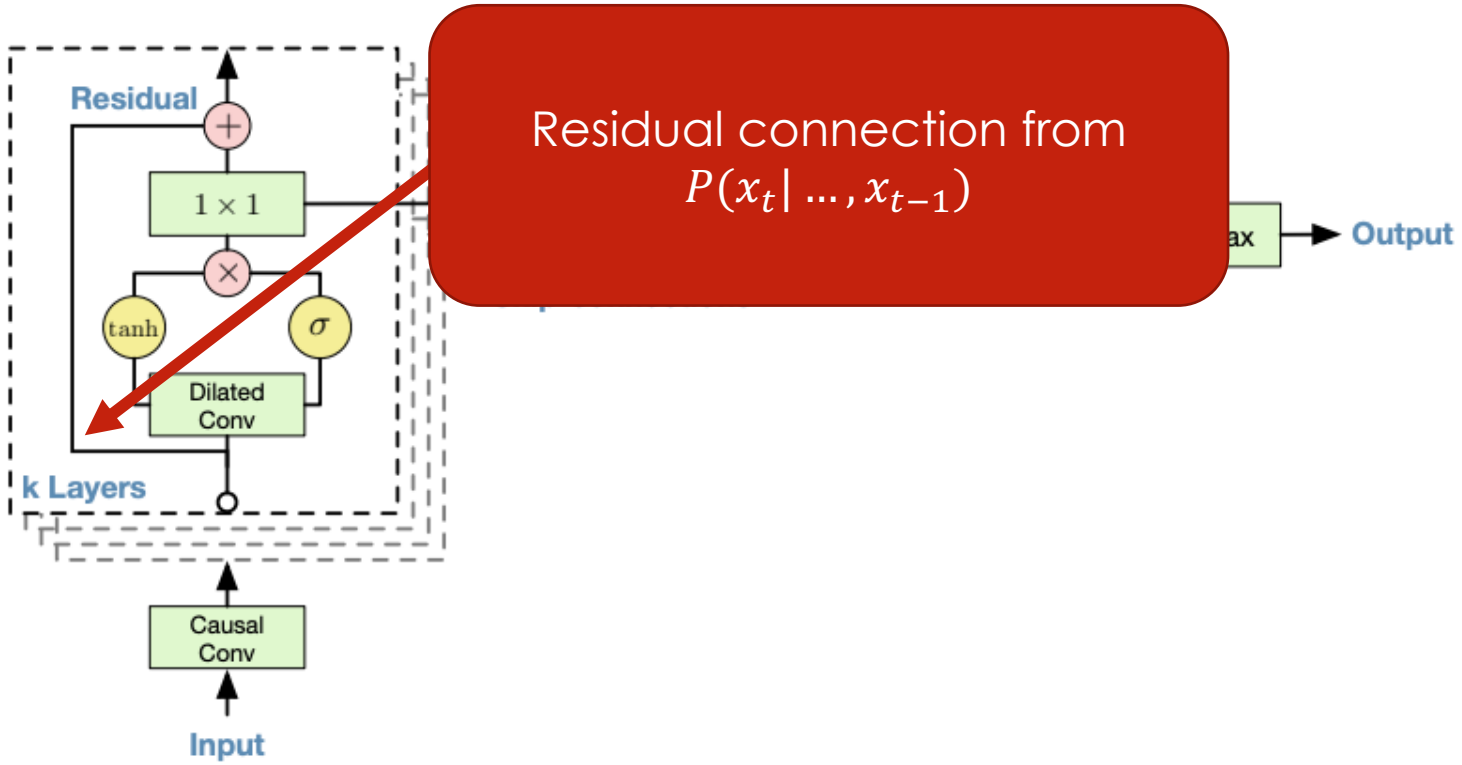
$$sign(x) = \begin{cases} -1, & if\ x < 0 \\ 0, & if\ x = 0 \\ 1, & if\ x > 0 \end{cases}$$

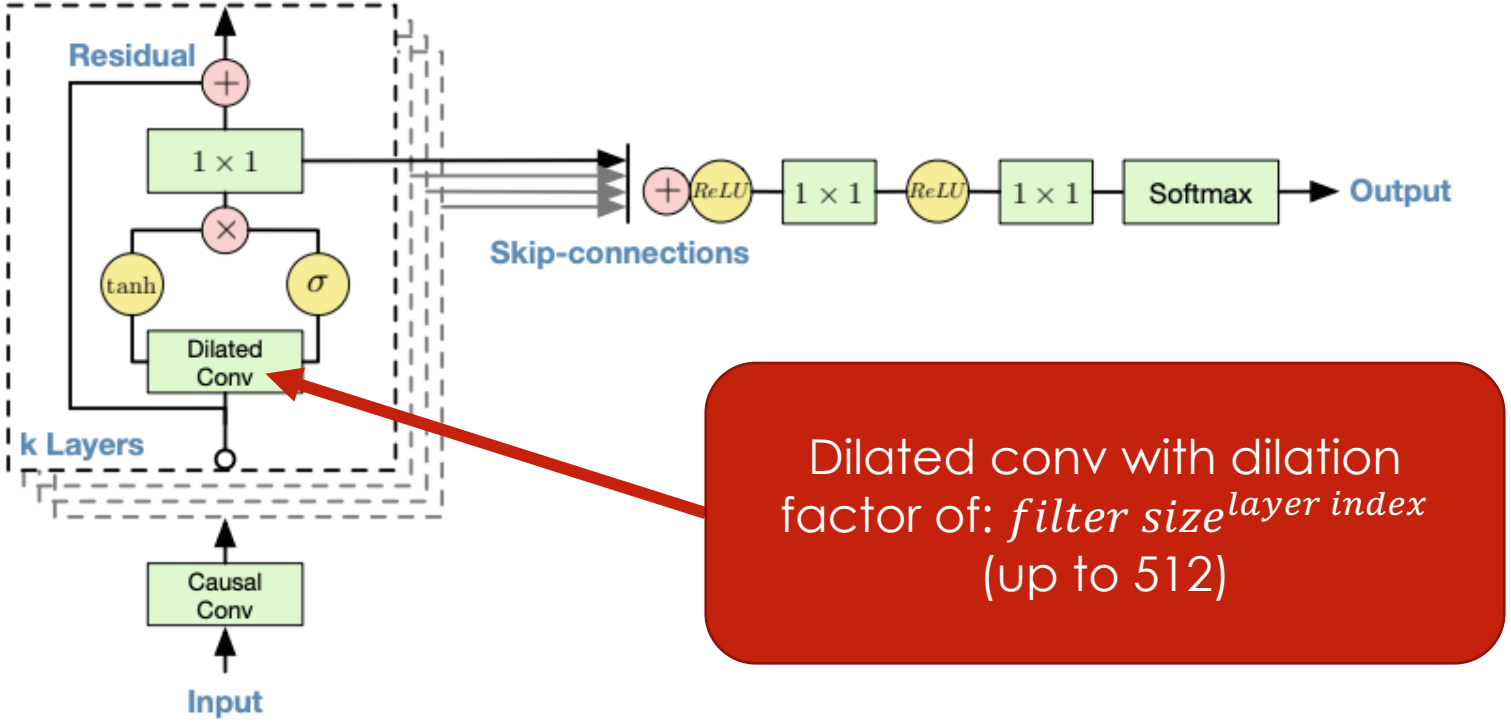
WAVENET





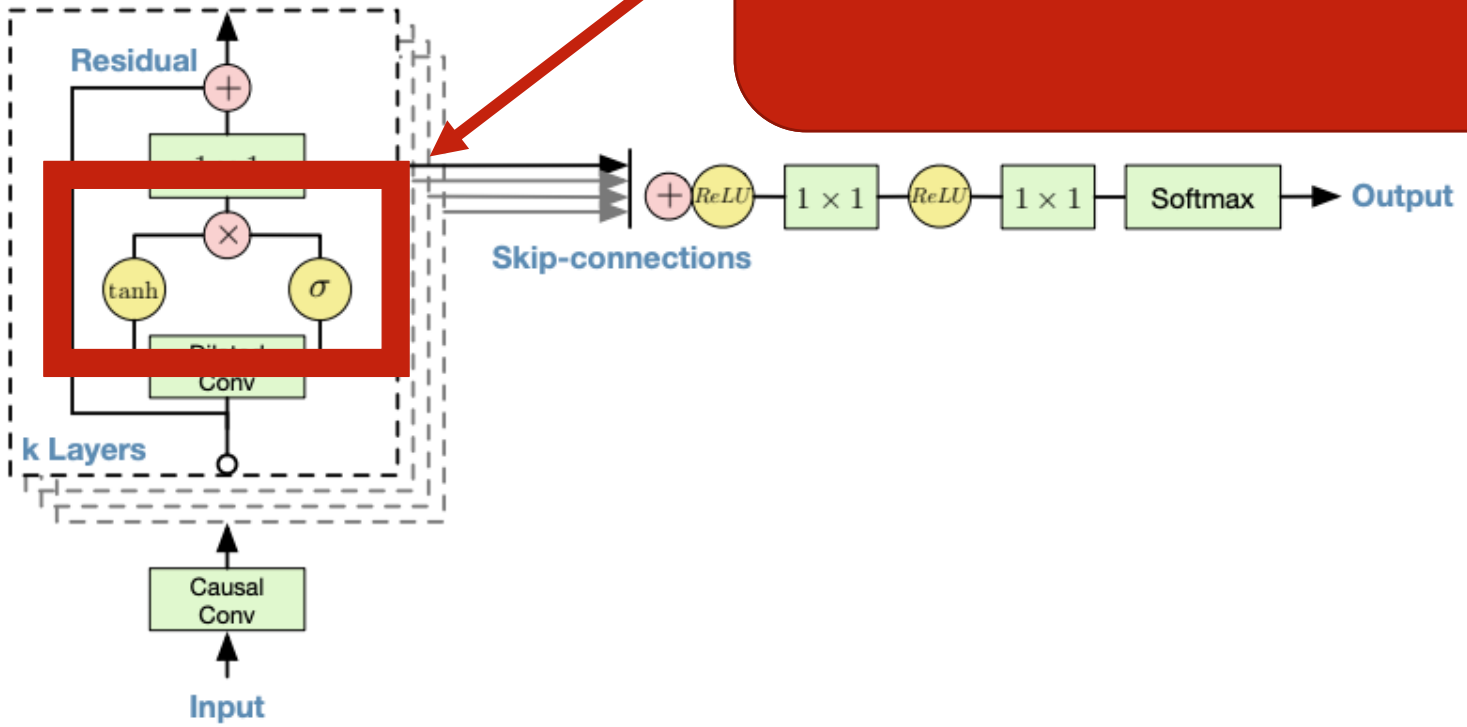
16kHZ audio sample (1-hot encoding), already transformed with μ-law


Residual
1 × 1
tanh
σ
Dilated Conv
k Layers
Causal Conv
Input
Residual connection from
$P(x_t|\ldots,x_{t-1})$
Output

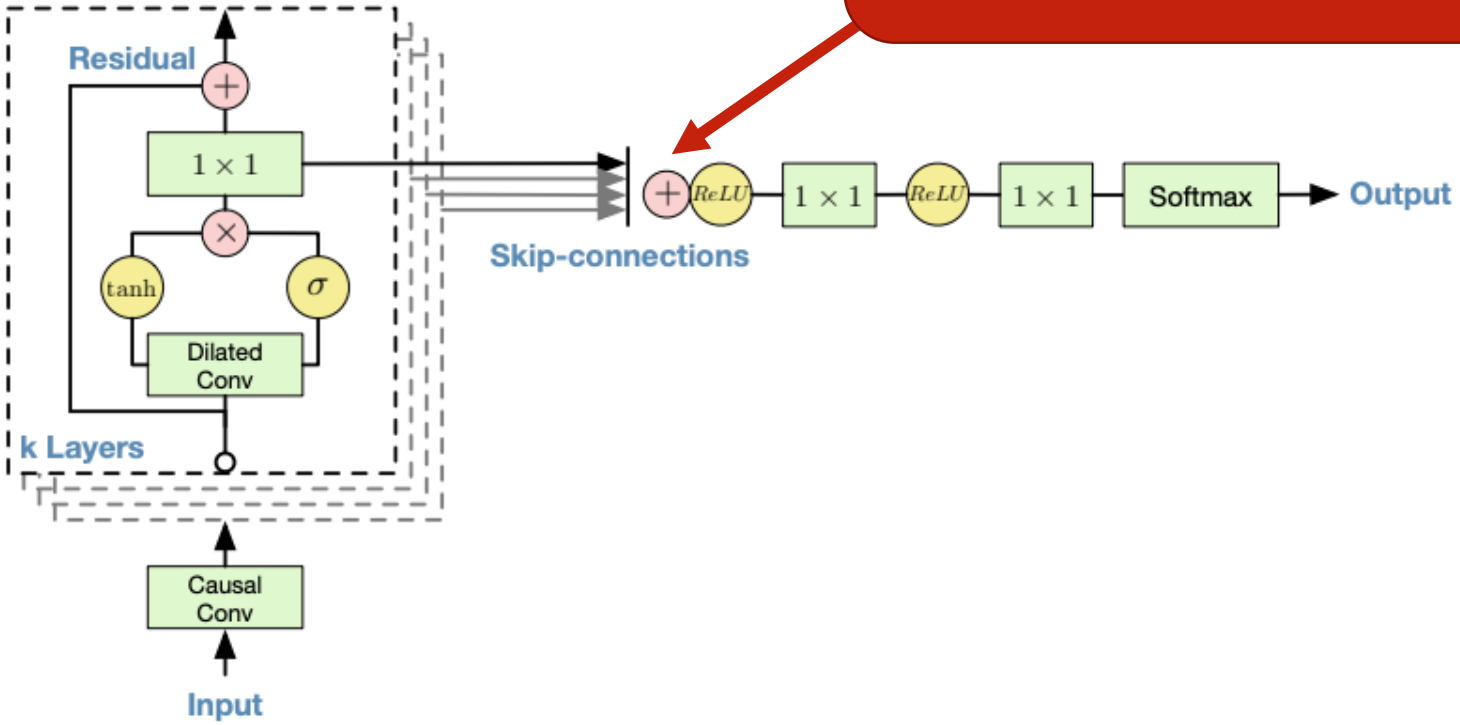


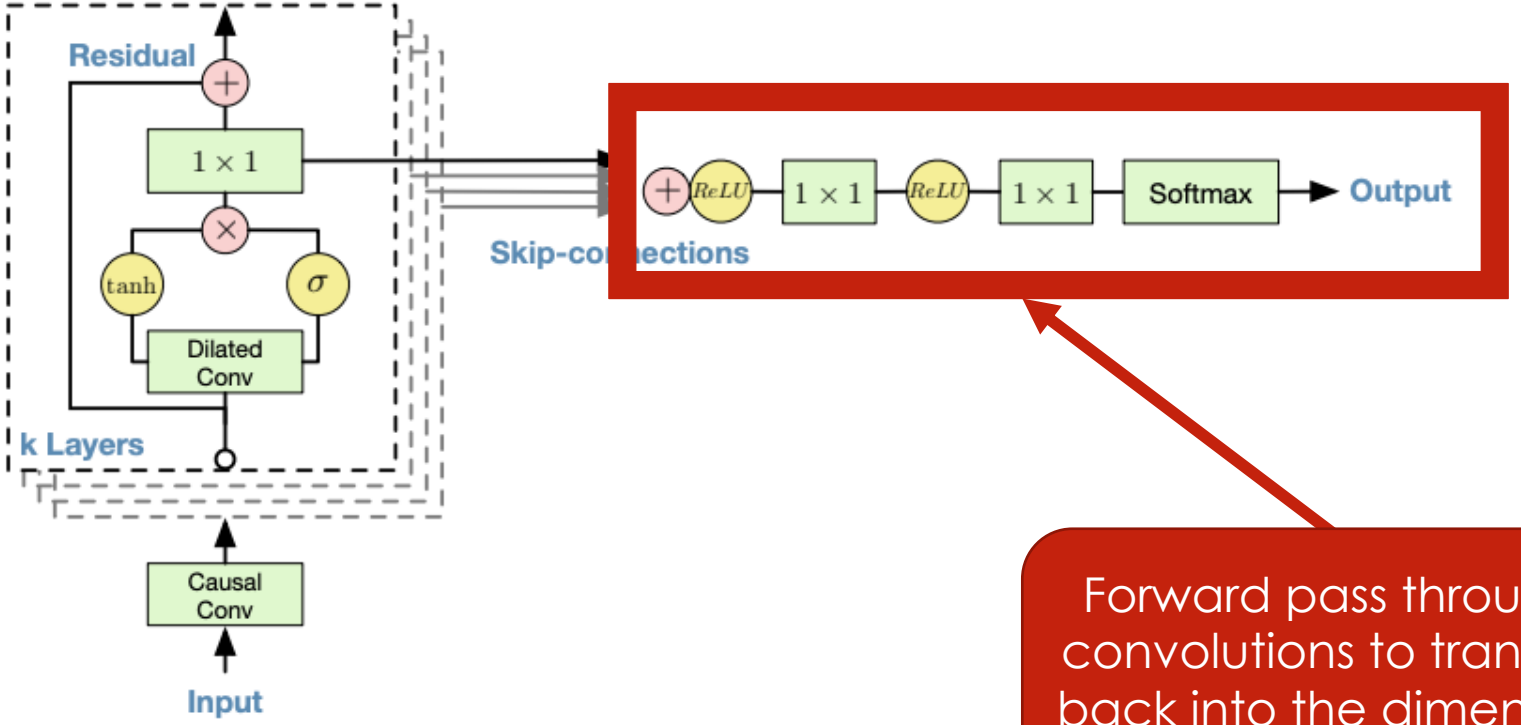
Dilated conv with dilation factor of: $filter\ size^{layer\ index}$ (up to 512)

$$z = \tanh\left(W_{f,k} * x + V_{f,k} * y\right) \cdot \sigma\left(W_{g,k} * x + V_{g,k} * y\right)$$



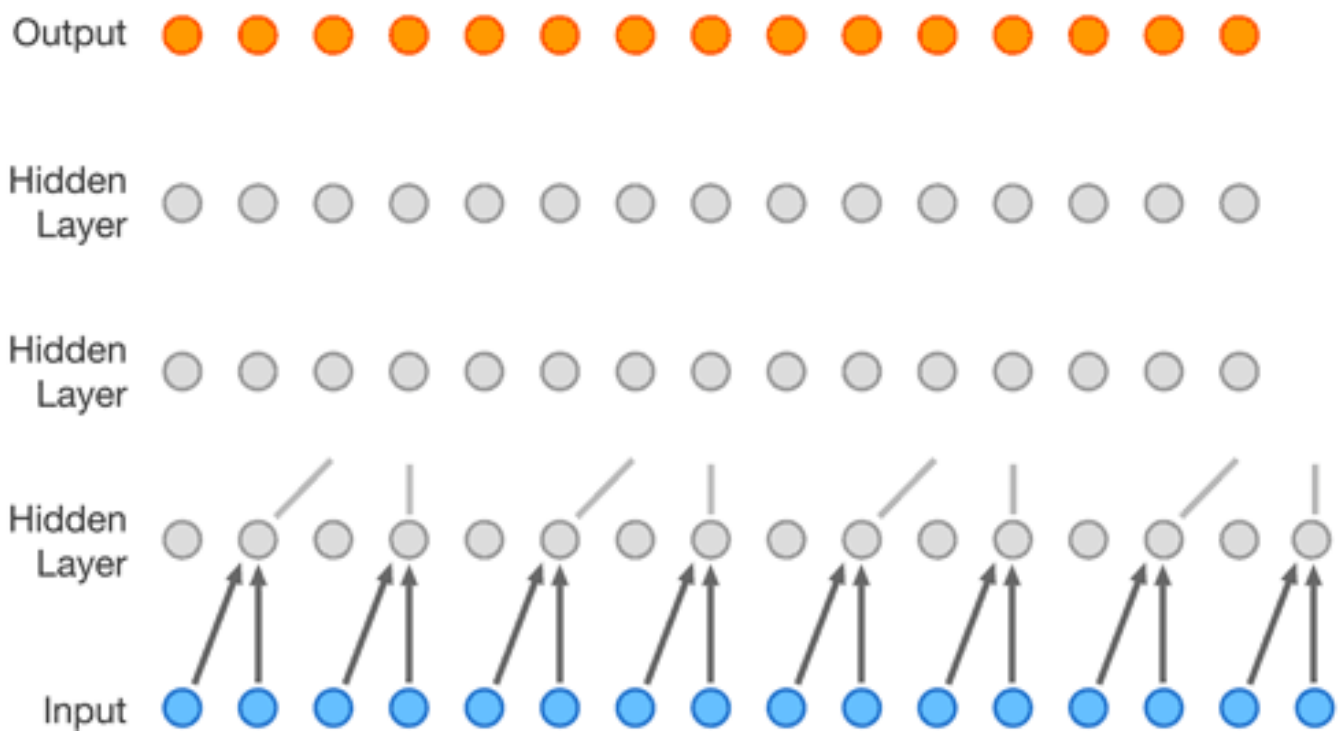
Sum all "*hidden states*", and pass them through a learnable skip connection weight matrix with ReLU





Forward pass through two 1D convolutions to transform them back into the dimensionality for the probability distribution

WAVENET



EXPERIMENTS & RESULTS

MULTISPEAKER SPEECH GENERATION	Model was only globally conditioned, not on linguistic features
	Generated somewhat natural voices with a hint of natural sounding prosody
	Model was able to learn all speakers characteristics

MULTISPEAKER SPEECH GENERATION EXAMPLES

TTS EXAMPLES

TTS

- Single speaker NA-English and Mandarin data used
- Local conditioning such as phones, syllables, word, phrase, and utterance (vocal features)
- Also global F_0-conditioning
- WaveNet beats baseline

	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

Table 1: Subjective 5-scale mean opinion scores of speech samples from LSTM-RNN-based statistical parametric, HMM-driven unit selection concatenative, and proposed WaveNet-based speech synthesizers, 8-bit μ-law encoded natural speech, and 16-bit linear pulse-code modulation (PCM) natural speech. WaveNet improved the previous state of the art significantly, reducing the gap between natural speech and best previous model by more than 50%.

TTS

- Single speaker NA-English and Mandarin data used
- Local conditioning such as phones, syllables, word, phrase, and utterance (vocal features)
- Also global F_0-conditioning
- WaveNet beats baseline

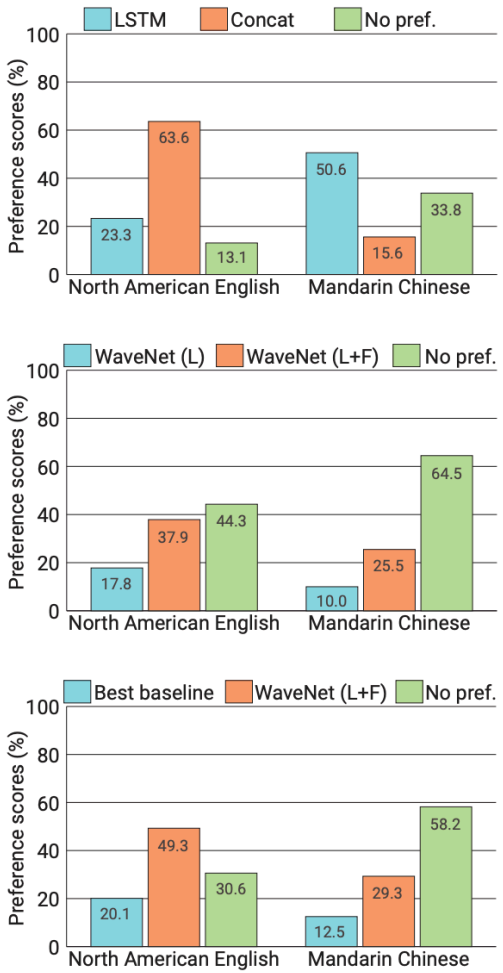


Figure 5: Subjective preference scores (%) of speech samples between (top) two baselines, (middle) two WaveNets, and (bottom) the best baseline and WaveNet. Note that LSTM and Concat correspond to LSTM-RNN-based statistical parametric and HMM-driven unit selection concatenative baseline synthesizers, and WaveNet (L) and WaveNet (L+F) correspond to the WaveNet conditioned on linguistic features only and that conditioned on both linguistic features and $\log F_0$ values.

MUSIC GENERATION

CONCLUSION

- Impressive and strong model, as far as you can understand it.
- WaveNet is a versatile model to use for different audio/speech tasks where raw audio is a good fit.
- Parallel WaveNet even faster
- Paper explained the architecture itself not that well, eventually even mistakes in the paper (see mu-law)?

THANKS FOR LISTENING!

DISCUSSION

- In section 2.2 of the paper, I don't understand the formula. The authors mention a μ-law companding transformation, what is that? How does it help to reduce the amount of probabilities that need to be considered?
  - Mu-law is a common compression technique used for audio signals, it‘s used with the mu constant of 255 (in NA, Japan). Mu-law helps us to eliminate the negative values of our 16bit integers and also compress them into 255 possibilities.
- In section 2.5 of the paper, for local conditioning, the authors use a transposed convolutional network to map the more coarse-grained linguistic features to the audio signal. What is a transposed CNN and how does it work? Why the 1x1 convolution mentioned below?
  - You switch the dimensions to learn not how to reduce the input but how to upsample it.

- What is exactly the one time step input of "16-bit linear pulse-code modulation (PCM)", do you know the pre-processing approach for obtaining these features?
  - Not deep enough into speech to answer this properly.
- With respect to doing conditioning the model on local conditioning: what is the intuition for conditioning on additional timeseries?
  - How else would you add your features?

- Why does the wavenet use softmax as an output layer when the value for each sample is continuous(ish) between -1 and 1? Would regression be possible instead?
  - It‘s not between -1 and 1, the output of sign(x) is.
- How do they feed the text to the model because it doesn't seem like there is an encoder layer. Is this done with the local conditioning of the wavenet model?
  - Correct, however they do use linguistic features extracted from text, like phones, words, phrases and other voice characteristics.

- About the output: What is the reason for the multiple combination of RELU and 1x1 Convolution?
  - My guess; to transform the features into a suitable softmax dimension.
- As far as I understand, the output keeps the time dimensionality of the input. This is done through the output from the softmax layer. What happens now to the summation of the residual values and the 1x1 convoluted values?
  - The elementwise addition after the first 1x1 conv. is passed back into the next layer.

- Could you explain how the non-linear quantization works (section 2.2) and why it works better than a linear quantization scheme?

- At the initial release of the WaveNet the authors found it required a lot of computational power. How was the performance of WaveNets imporved in terms of computational cost/ How could it be improved?
  - It‘s not the computational power, it‘s the complexity that makes it quite *slow*. When working with 16kHZ samples, we have an input of 16000 samples/sec. The figures showed only 3 layers of the network to cover a receptive field size of 16. Therefore we are dealing with about 30 layers for all to infer from w.r.t. to the fact that we only process 1 second of audio. See follow up paper.

- In Section 2.6, they mention the term dilation stages. What is that?
  - Dilation stage is the growth of the dilation factor (doubled for each layer).
- Also in Section 2.6, they describe context stacks, but I don't understand their description. What are context stacks?
  - Contextual representations over a limited receptive field learned by a seperate network which may also use pooling layers.
- What is the approximate training time for such a model?
  - 16000 samples per second, which need 30 layers to pass through plus 16000 times of inference. People said it may take 2 GPU minutes for a single inference.

- Have there been other approaches at producing classical music or are there more examples available which were produced by wavenet? The ones available on the website sound a bit clustered and not very diverse. Would it be possible to obtain different genres of music just as different types of voices (male, female, etc.) were produced?
  - Not that I know of. Yes, because we learn from raw audio signals.
- Is the global/local conditioning (sect. 2.5) applied in every layer?
  - In every time step rather than layer.
- How do the skip connections compare to soft-attention?
  - Skip connections let gradients flow more freely. They are also said that they provide faster convergence.

- What is a μ-law companding transformation and what is a logarithmic fundamental frequency value?
  - Mu-law answered. Fundamental Frequency (F_0) is the frequency of the lowest-frequency component of a complex sound, as evident in the repetition rate of the waveform, or the the rate of vocal fold vibration during sound production
- Regarding the context stack: Could you explain what exactly is done here, why is the context stack the best/competitive way to increase the receptive field size of WaveNet?
  - Think of a kind of fine-tuning for the local conditioning in WaveNet.
- Can you explain how the inputs to equation in section 2.2 can be between -1 and 1 when raw audio is stored as 16 bit integers?
  - Been there done that.
- What is a speaker embedding in a TTS Model? Where do we get it from? How does it work?
  - It‘s a learned embedding through another (vocoder) network.