Generative Modelling of Sequential Data

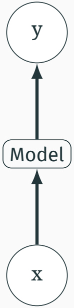
M.Sc. thesis in collaboration with Corti ApS

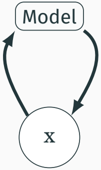
Magnus Berg Sletfjerding

February 9th, 2022

Contents

Introduction





Supervised Learning

Learns a mapping from data $\mathbf{x}$ to labels $\mathbf{y}$:

$$p(\mathbf{x}|\mathbf{y}) = \sum_{i=1}^{N} p(y_i|x_i)$$

Unsupervised Learning

Learns the structure of the data $\mathbf{x}$:

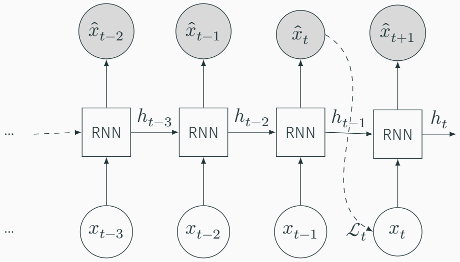
$$p(\mathbf{x}) = \sum_{i=1}^{N} p(x_t)$$

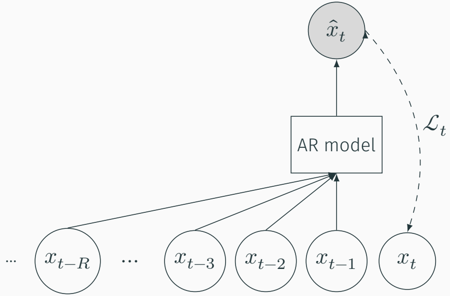
Why study hierarchies of information in sequences?

- Most data we work with has some hierarchical structure
  - Text
  - Video
  - Proteins/DNA
- Human brains process hierarchies of information natively
  - Human-like AI requires hierarchical processing
- All real-world data has a sequential dimension - time!

Sequence modeling optimize the model's likelihood $p(\cdot)$ over the data $\mathbf{x}$, by conditioning the probability of x_t on previous timesteps:

$$p(\mathbf{x}) = \prod_{t=1}^{N} p(x_t | x_{<t}), \qquad \mathbf{x} \in \mathbb{R}^N$$





Recurrent architectures

Condition $p(x_t|x_{<t})$ through one or more hidden states h_t passed between timesteps:

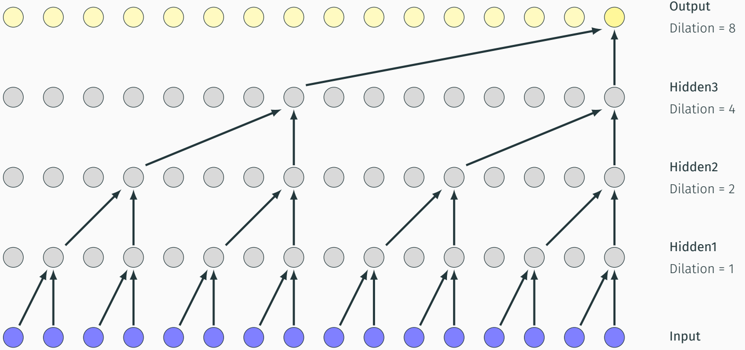
$$p(x_t, h_t|x_{<t}) = p(x_t|x_{t-1}, h_{t-1})$$

Autoregressive Architectures

Condition $p(x_t|x_{<t})$ by viewing a receptive field of size R of the input sequence.

$$p(\mathbf{x}) = \prod_{t=R+1}^{N} p(x_t|x_{\geq t-R+1,<t})$$

WaveNet - Convolutional Autoregressive Sequence Modelling



- Common vocoder in Speech To Text production systems
- Makes use of dilated convolution to inflate receptive field
- No “hidden state” for representing earlier timesteps
- Constrained to look back within receptive field

Problem and Hypotheses

Main Problem with WaveNet ...

- excelling at capturing local signal structure
- missing long-range correlations
- low receptive field (300ms)
- audio generation sounds like babbling

Hypotheses Investigated

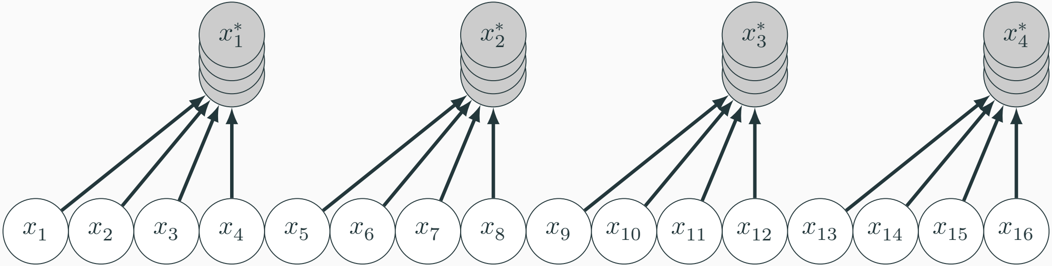
1. WaveNet's receptive field is the main limiting factor for modeling long-range dependencies.
2. WaveNet's stacked convolutional layers learn good representations of speech.
3. WaveNet's hierarchical structure makes it suitable to learn priors over representations of speech such as text.
4. A large WaveNet architecture trained on speech can generate coherent words and sentence fragments

Experiments

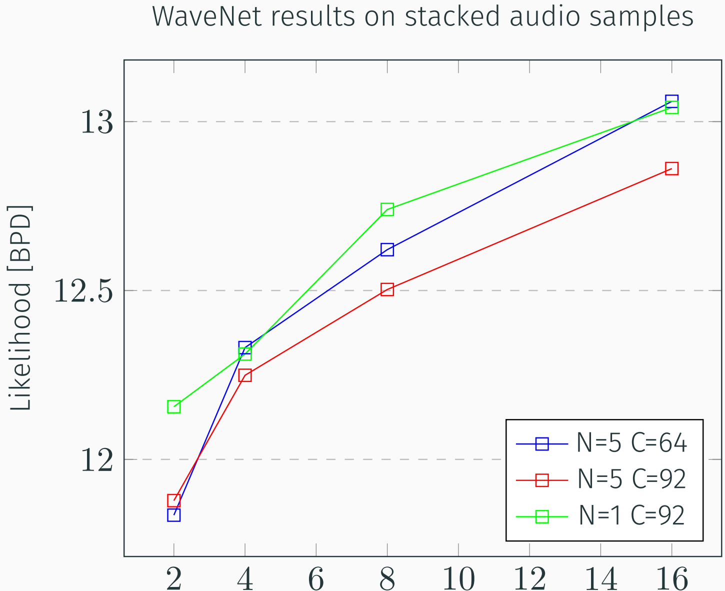
Experiment overview

1. Expanding Receptive Field by Stacking
2. Latent Space of Stacked WaveNets
3. WaveNet as a Language Model
4. WaveNet as an ASR preprocessor

Expanding Receptive Field by Stacking - Setup



Expanding Receptive Field by Stacking - Results

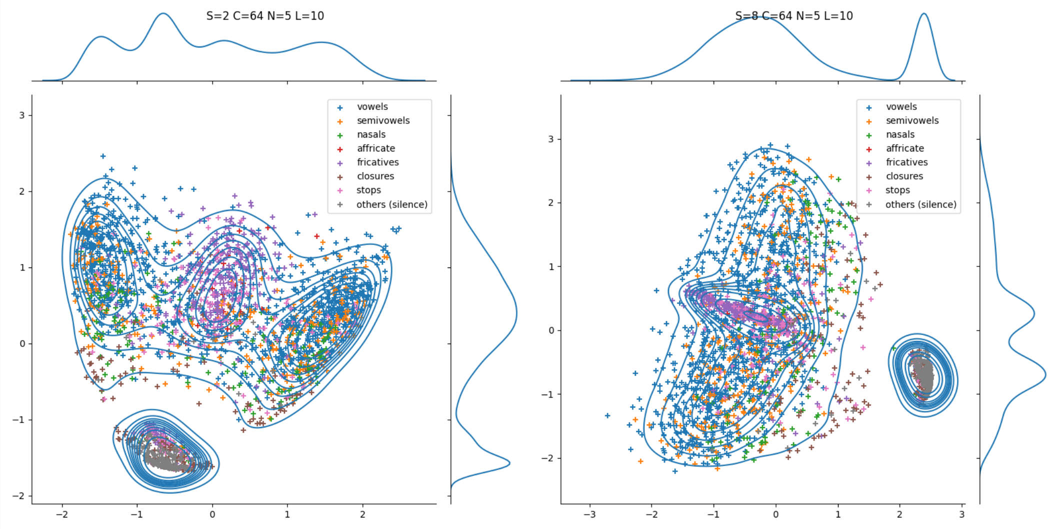


Expanding Receptive Field by Stacking - Conclusions

1. Increasing the stacking does not improve likelihoods significantly.
2. Increasing the number of residual channels increases evaluation likelihoods.
3.

Latent space of stacked WaveNet - Setup

Latent space of stacked WaveNet - Results



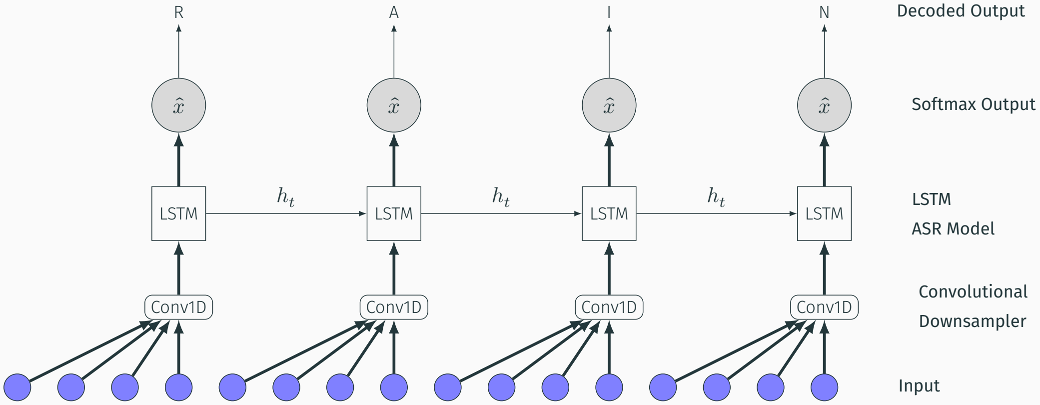
WaveNet as a Language Model - Setup

WaveNet as a Language Model - Results

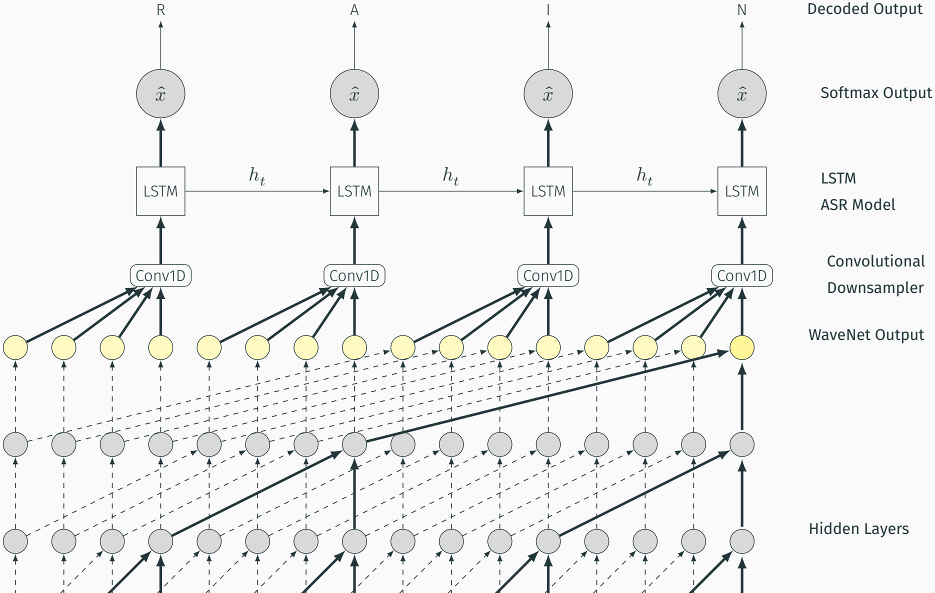
Model	Dataset	BPD (test)
Mogrifier LSTM [?]	PTB	1.083
Temporal Convolutional Network [?]	PTB	1.31
WaveNet N=5 L=4 R=24 [RF 126]	PTB	1.835
WaveNet N=5 L=4 R=32 [RF 126]	PTB	**1.666**
WaveNet N=5 L=4 R=48 [RF 126]	PTB	1.678

WaveNet as a Language Model - Conclusions

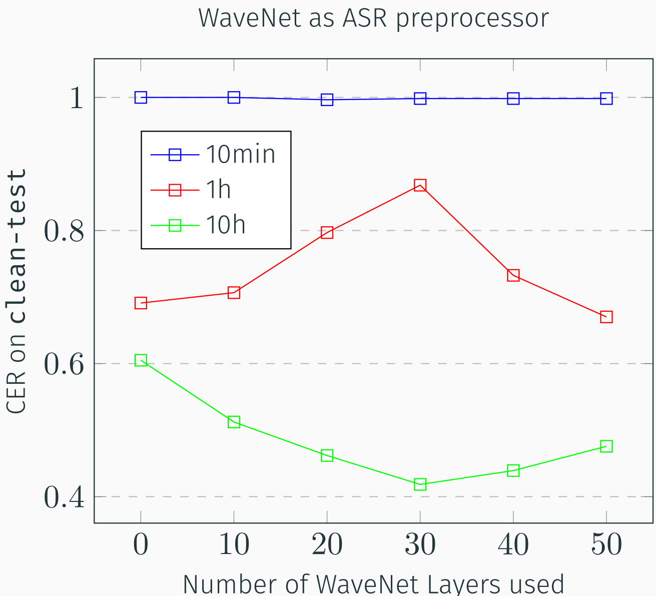
WaveNet as an ASR preprocessor - control setup



WaveNet as an ASR preprocessor - experiment setup



WaveNet as an ASR preprocessor - Results



WaveNet as an ASR preprocessor - Conclusions

- Using WaveNet as a preprocessor decreases the loss when trained on the 1 hour and 10-hour training subsets.
- The best performance occurs when using 30 layers of the WaveNet trained on 10 hours of training data.
- Notably, WaveNet's use as a preprocessor grows more competitive when increasing the training data size.

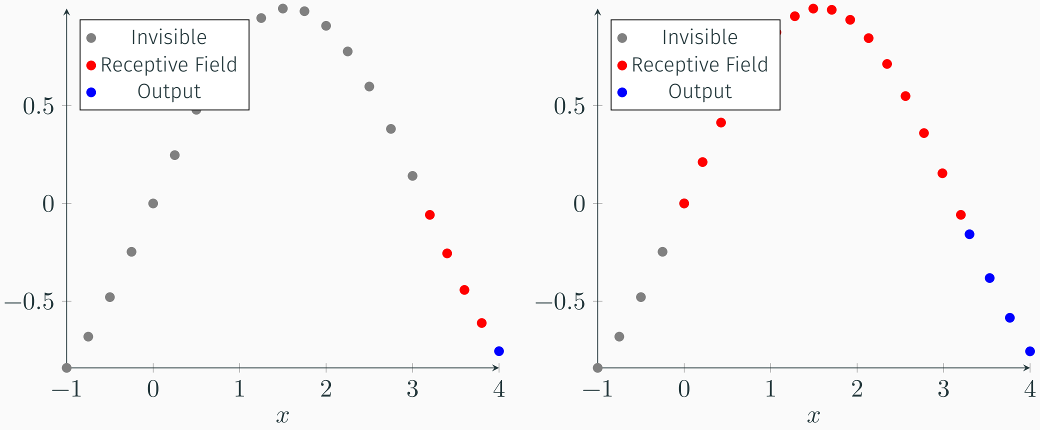
Conclusions

Conclusions

Some stuff here

References

Visualization of stacking on Sin curve



Notation

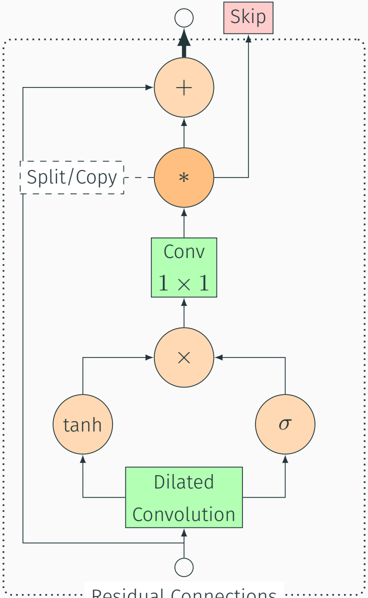
Symbol	Explanation
x_i,x_t	The ith index of $\mathbf{x}$, of size N. $x_i \in \mathbb{R}^N$. x_t is used when data is time-resolved.
$\mathbf{x}$	The data x, composed of vectors x_i. $\mathbf{x} \in \mathbb{R}^{T\times N}$
$p_\theta(\cdot), p(\cdot)$	Likelihood function over model parameters θ. Denoted $p(\cdot)$ for brevity
$\hat{x}_i$	Model prediction for x_i.
$\mathcal{L}_i$	Loss function for ith index.
R	Receptive field size.
S	Size of stack size used in stack transformations
d_i	Dilation of ith layer in a WaveNet architecture
C	Number of residual channels

Codebase

TODO: Implementations to mention:

- Residual Stack
- Categorical WaveNet
- DMoL WaveNet

Residual Block of WaveNet



DMoL vs. Categorical output distribution

Discretized Mixture of Logistics

With a mixture of K logistic distributions, for all discrete values of x except edge cases:

$$P(x|\pi, \mu, s) = CDF(x - 0.5, x + 0.5) = \sum_{i=1}^{K} \pi_i[\sigma(\frac{x + 0.5 - \mu_i}{s_i}) - \sigma(\frac{x - 0.5 - \mu_i}{s_i})]$$

Where $\sigma(\cdot)$ is the logistic sigmoid: $\sigma(x) = \frac{1}{1+e^x}$, π is the relative weight vector, μ is the location vector and s is the scale vector.

Softmax distribution

In a softmax distribution, the probability of the ith out of N discrete values is defined by:

$$\sigma(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}$$

Different tested embeddings for stacked WaveNet input

Embedding type	Dim	Number	Note
Lookup table embedding with input dimensionality $S \times C$	128	1024	Outputs collapsed to silence (suspect too sparse embeddings)
S embeddings convolved together	128	$S \cdot 256$	White noise output.
2 Layer perceptron with input size S and output size R	R	Continuous	Final used embedding

Overview of extra experiments

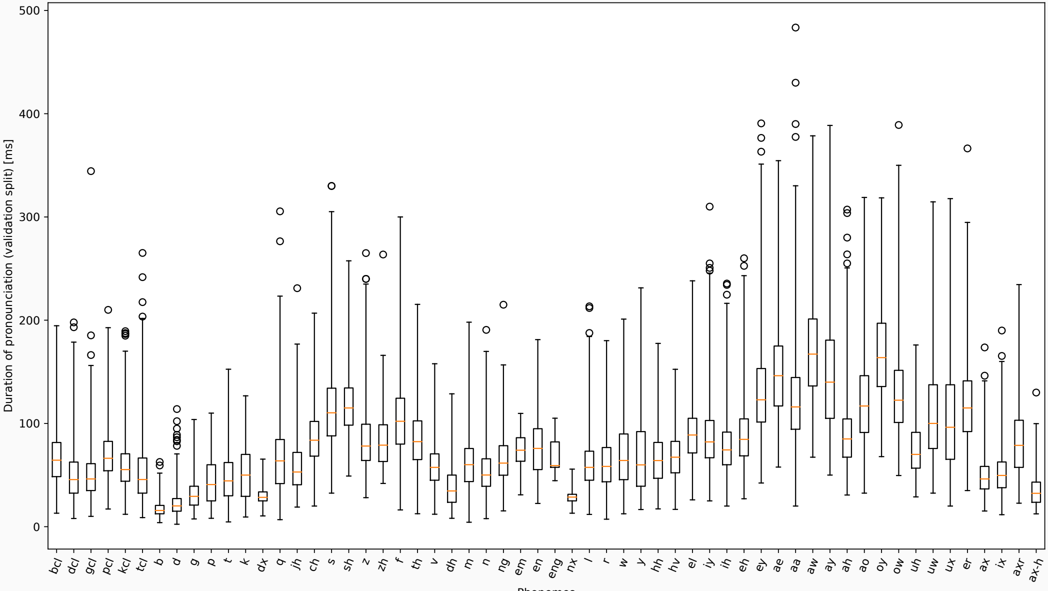
Model	Dataset(s)	Notes
Single-Timestep WaveNet (softmax output)	TIMIT	Slow convergence compared to later DMoL
Stacked WaveNet (softmax output)	TIMIT, Librispeech	Collapses to predict silence for all timesteps
Single Timestep WaveNet	Generated Sinusoids with periodically modulated pitch.	Fails to follow modulation in pitch

Phonemes in TIMIT

Group	Phonemes
Vowels	iy, ih, eh, ey, ae, aa, aw, ay, ah, ao,
	oy, ow, uh, uw, ux, er, ax, ix, axr, ax-h
Stops	b, d, g, p, t, k, dx, q
Closures	bcl, dcl, gcl, pcl, tck, kcl, tcl
Affricates	jh, ch
Fricatives	s, sh, z, zh, f, th, v, dh
Nasals	m, n, ng, em, en, eng, nx
Semivowels and Glides	l, r, w, y, hh, hv, el
Others	pau, epi, h#, 1, 2

Table 4: TIMIT phoneme groupings

Phoneme lengths in TIMIT



Explained

Run gradient evaluation over a trained WaveNet model and visualize the outputs.

Hypothesis tested

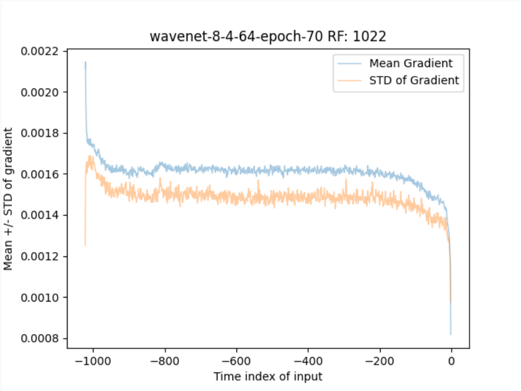
WaveNet uses the entirety of its receptive field for next-step prediction.

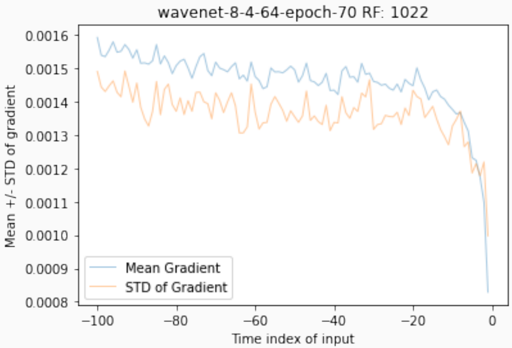
- Gradients in the end of the RF (close to output) are larger than the gradients in the rest of the RF.
- Gradients do NOT collapse to 0 around the beginning of the RF (furthest away from output).

Method

1. Calculate vector-Jacobian product with `torch.autograd`
2. Calculate norm with `torch.linalg.norm`

WaveNet Gradient analysis over input space - 2





Mu Law Distribution

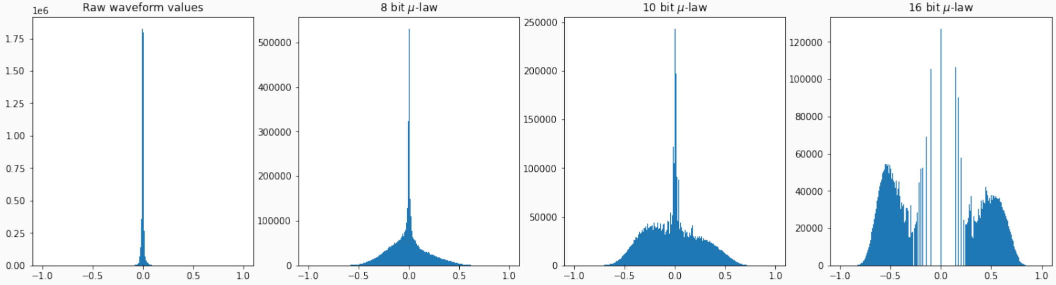


Figure 3: Distribution of raw PCM values from the TIMIT test set. Far left: PCM (16-bit integers). Others: corresponding distribution after μ-law encoding the waveform values with $\mu \in 8, 10, 16$.