Speech Synthesis

Replicating MelNet: A Generative Model for Audio in the Frequency Domain

Jorge García Pueyo (jgarciapueyo@gmail.com)

Who Am I?

- I am studying a Computer Science degree.
- I am from Zaragoza, a city in the northeast of Spain located between Madrid and Barcelona.
- I have participated in the Erasmus+ programme at KTH last academic year, taking courses of the Master in Software Engineering for Distributed Systems and Master in Machine Learning.

Summary

- We have implemented:
  - *MelNet (S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," arXiv preprint arXiv:1906.01083, 2019)*
- We have applied this model to the task of unconditional speech training.
- This model has been applied to new datasets producing insightful results on the task of unconditional speech.

- The code for the project can be found:
  - *https://github.com/jgarciapueyo/MelNet-SpeechGeneration*
- The audio files for the examples can be found in:
  - *https://github.com/jgarciapueyo/MelNet-SpeechGeneration/tree/master/results*

Speech Synthesis

- It consists on artificially creating human speech.
- It can be categorised into:
  - Unconditional Speech: generating random babbling
  - Conditional Speech (also known as Text-to-Speech)
- There are different approaches:
  - Concatenative Speech Synthesis
  - Parametric Speech Synthesis

Parametric Speech Synthesis

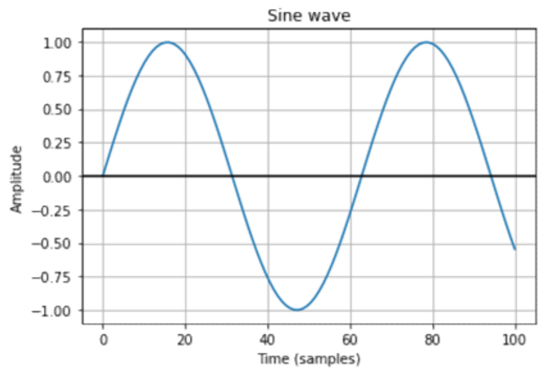
- Examples of these models are
  - Hidden Markov Models
  - Deep Learning Models

- These models work on varying audio representations:
  - Some of them use waveforms, which are one dimensional representation *(i.e. WaveNet)*
  - Other models use spectrograms, which are two dimensional representations *(i.e MelNet)*

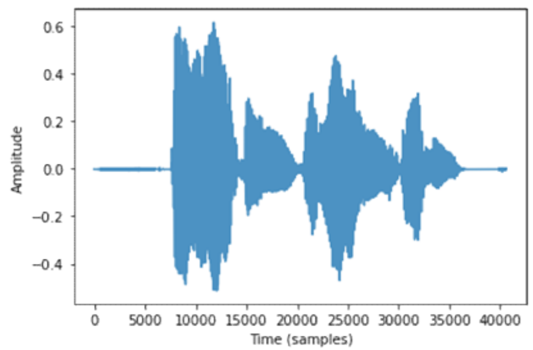
Audio Representations

- Sound is transmitted through air as pressure oscillations.
- This can be represented by a pressure-time plot showing the deviation of air pressure from normal state, also known as a **waveform**.
- A **waveform** is represented digitally as one-dimensional discrete-time signal

$$y = (y_1, \ldots, y_n)$$



Sine waveform

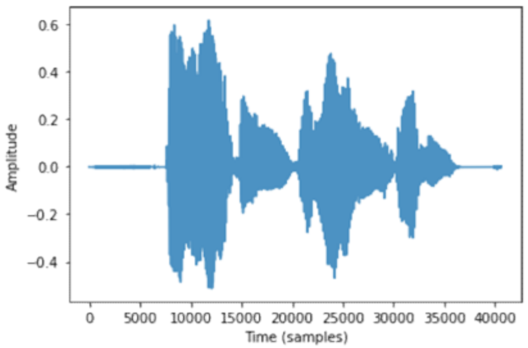


Real audio waveform

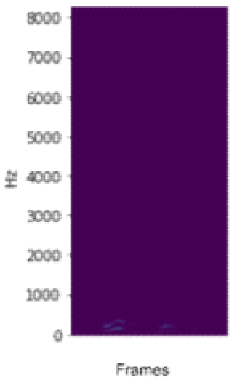
Audio Representations

- Applying the Short-Time Fourier Transform to the waveform we obtain a two dimensional time-frequency representation known as a **spectrogram**.
- In this case, we use the **energy (or amplitude) spectrogram.**

$$x_{ij} = \|STFT(y)\|^2$$



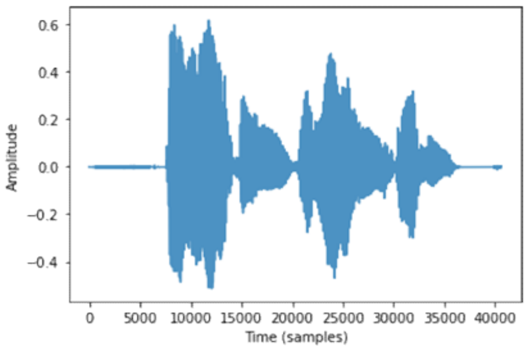
Real audio waveform



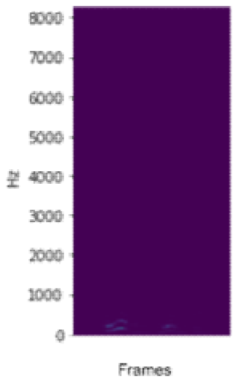
Spectrogram

Audio Representations

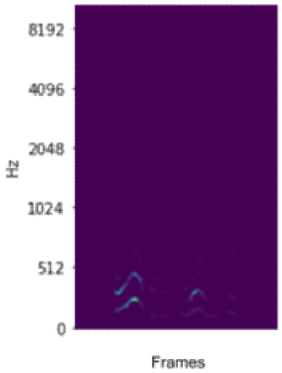
- Basic spectrograms are not aligned with how humans perceive audio.
- We transform:
  - the frequency axis to the Mel scale, creating a **Melspectrogram.**
  - the amplitude values to the decibel scale.



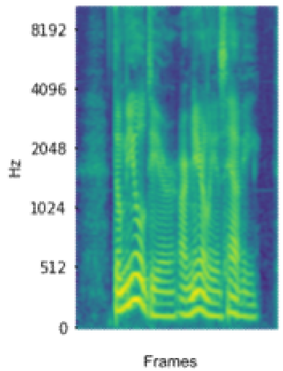
Real audio waveform



Spectrogram of a real audio waveform



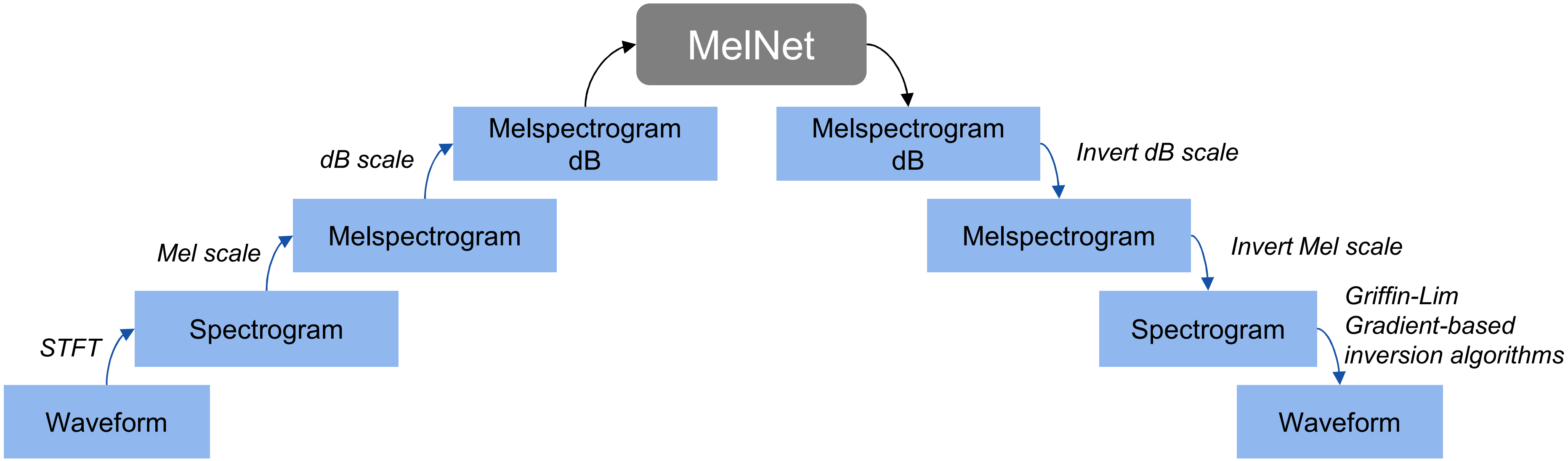
MelSpectrogram



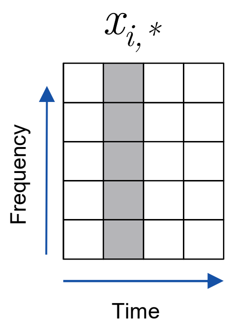
MelSpectrogram in decibel scale

Audio Representations

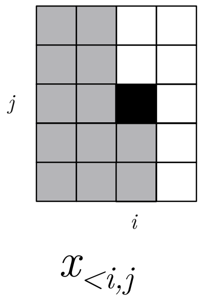
- MelNet uses the melspectrogram as input and output.



MelNet: Definitions



- We will consider the spectrogram as a sequence of frames $x_{i,*}$
- Inside each frame we go from low to high frequencies
- The context of a value $x_{i,j}$, named $x_{<i,j}$,consists of all previous frames $x_{<i,*}$ and the lower frequencies in the same frame $x_{i,<j}$.



MelNet: Single Tier

- MelNet is an autoregressive model which factorizes the joint distribution over a spectrogram x as a product of conditional distributions.

$$p(x;\theta) = \prod_i \prod_j p(x_{ij} | x_{<ij}; \theta_{ij})$$

- Each factor is modelled as a Gaussian Mixture Model (GMM) with K components $\theta_{ij} = \{\mu_{ijk}, \sigma_{ijk}, \pi_{ijk}\}_{k=1}^{K}$.

$$p(x_{ij} | x_{<ij}; \theta_{ij}) = \sum_{k=1}^{K} \pi_{ijk} N(x_{ij}; \mu_{ijk}, \sigma_{ijk})$$

MelNet: Single Tier

- MelNet makes the parameters θ_{ij} to be governed by the output of a neural network f with weights ψ as a function of the context $x_{<ij}$.

$$\hat{\theta}_{ij} = f(x_{<ij}, \psi)$$

- The output of the neural network is assumed to be unconstrained parameters and they are constrained to ensure that the output parameterizes a valid Gaussian Mixture Model.

MelNet: Single Tier

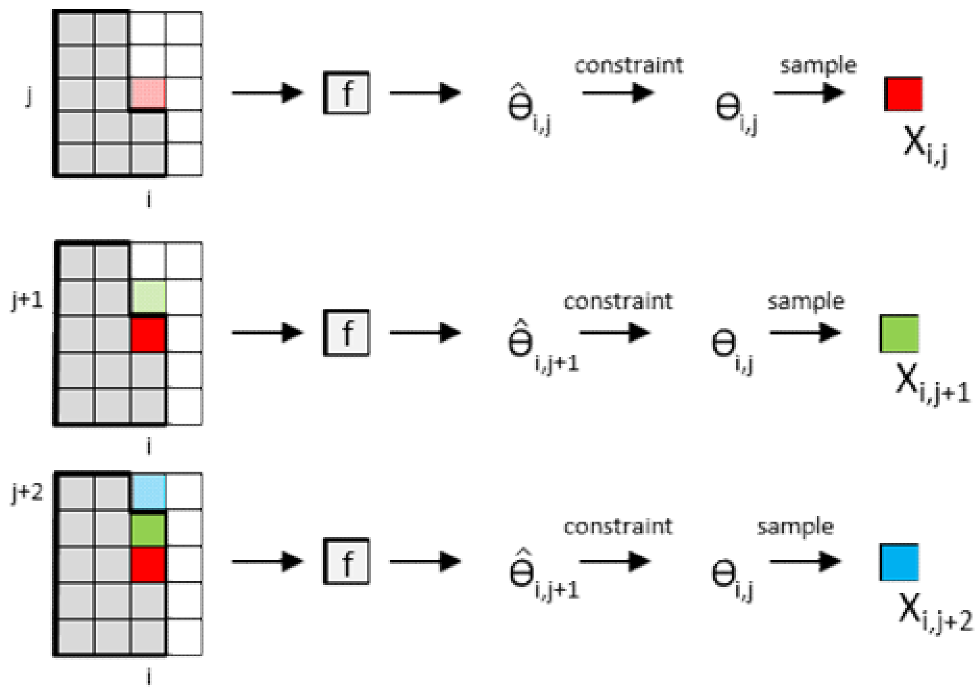
- MelNet uses maximum-likelihood estimation for the parameters $\theta = \{\theta_{00}, ..., \theta_{frames,freq}\}$ by minimizing the negative log-likelihood via gradient descent.

$$loss(x, \theta) = -log\ \mathcal{L}(\theta|x) = -log \prod_i \prod_j p(x_{ij}|x_{<ij}; \theta_{ij}) = \sum_i \sum_j -log\ p(x_{ij}|x_{<ij}; \theta_{ij})$$

- The negative log-likelihood of an individual value x_{ij} is

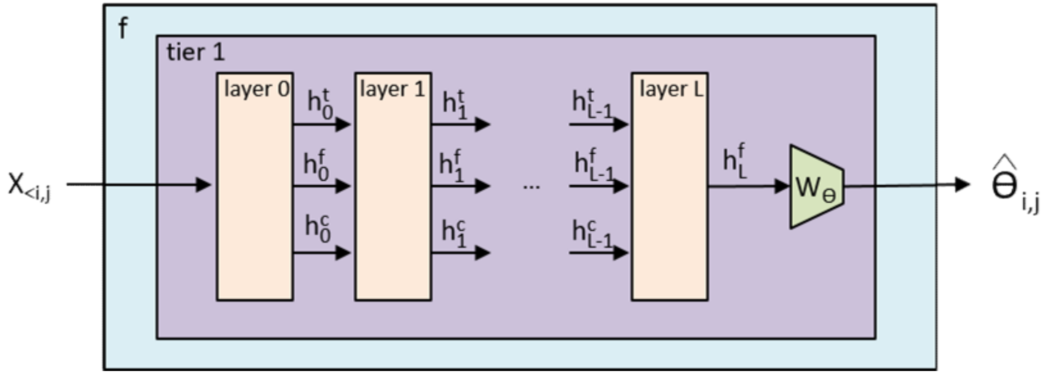
$$-log\ p(x_{ij}|x_{<ij}; \theta_{ij}) = -log \sum_{k=1}^{K} \pi_{ijk} N(x_{ij}; \mu_{ijk}, \sigma_{ijk})$$

MelNet: Single Tier



MelNet: Single Tier

- The network f is composed of tiers and every tier is composed of layers.
- Every layer is composed of stacks of computation whose objective is to extract features from different segments of the input to summarize the full context $x_{<ij}$.



Architecture of a single-tier model

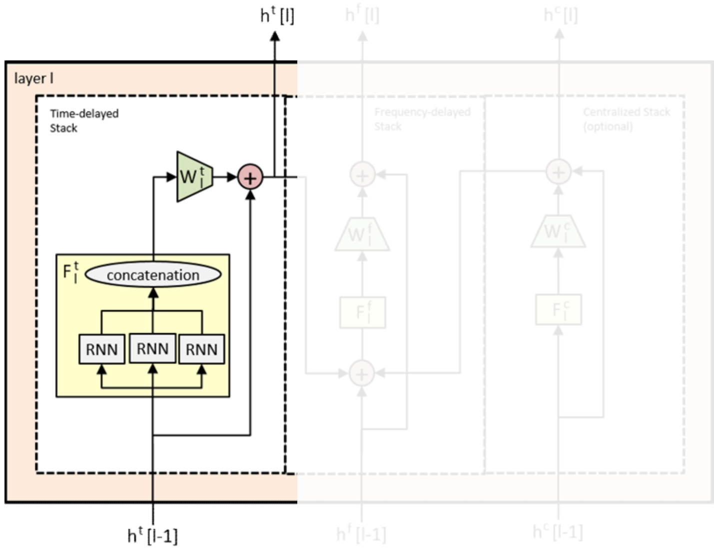
MelNet: Single Tier

- There are three stacks in each layer:
  - **Time-delayed**: computes features from the previous frames $x_{<i,*}$

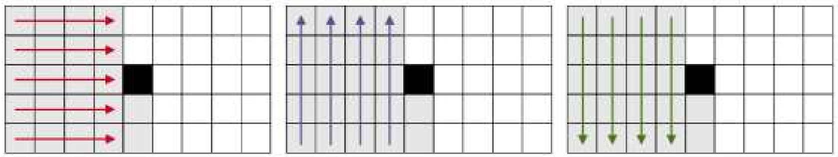
$$h^t_{ij}[0] = W^t_0 x_{i-1,j}$$

$$h^t_{ij}[l] = W^t_l \mathcal{F}^t_l(h^t[l-1])_{ij} + h^t_{ij}[l-1]$$

  - **Frequency-delayed**
  - **Centralized**



Computation graph of a single layer of the network



RNNs used in the Time-delayed stack

Source: *Melnet: A generative model for audio in the frequency domain*

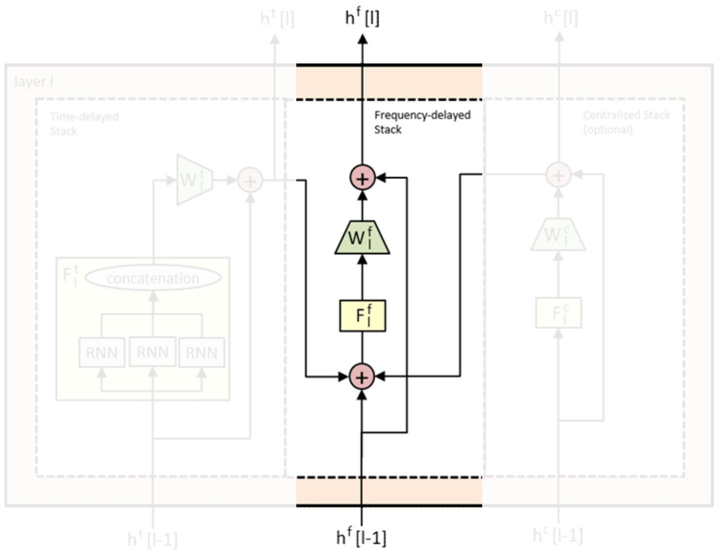
MelNet: Single Tier

- There are three stacks:
  - **Time-delayed**
  - **Frequency-delayed**: computes features from the elements within a frame $x_{i,<j}$ and the features from the time-delayed and centralized stack.

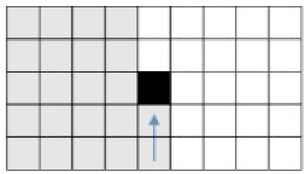
$$h_{ij}^f[0] = W_0^f x_{i,j-1}$$

$$h^f[l] = W_l^f \mathcal{F}_l^f(h^f[l-1], h^t[l], h^c[l]) + h^f[l-1]$$

  - **Centralized**



Computation graph of a single layer of the network



RNN used in the Frequency-delayed stack

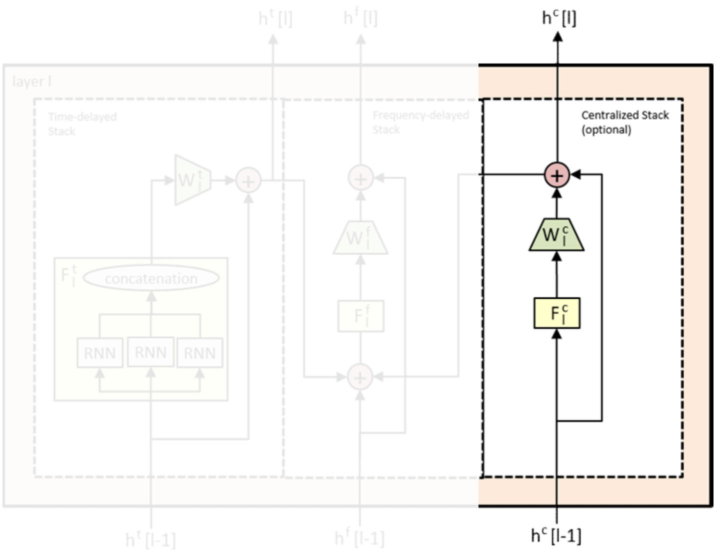
Source: *Melnet: A generative model for audio in the frequency domain*

MelNet: Single Tier

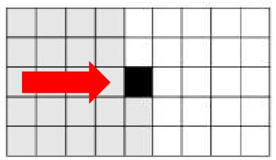
- There are three stacks:
  - **Time-delayed**
  - **Frequency-delayed**
  - **Centralized**: computes information from the previous frames $x_{<i,*}$, but taking the entire frame.

$$h_i^c[0] = W_0^c x_{i-1,*}$$

$$h_i^c[l] = W_l^c \mathcal{F}_l^c(h^c[l-1])_i + h_i^c[l-1]$$



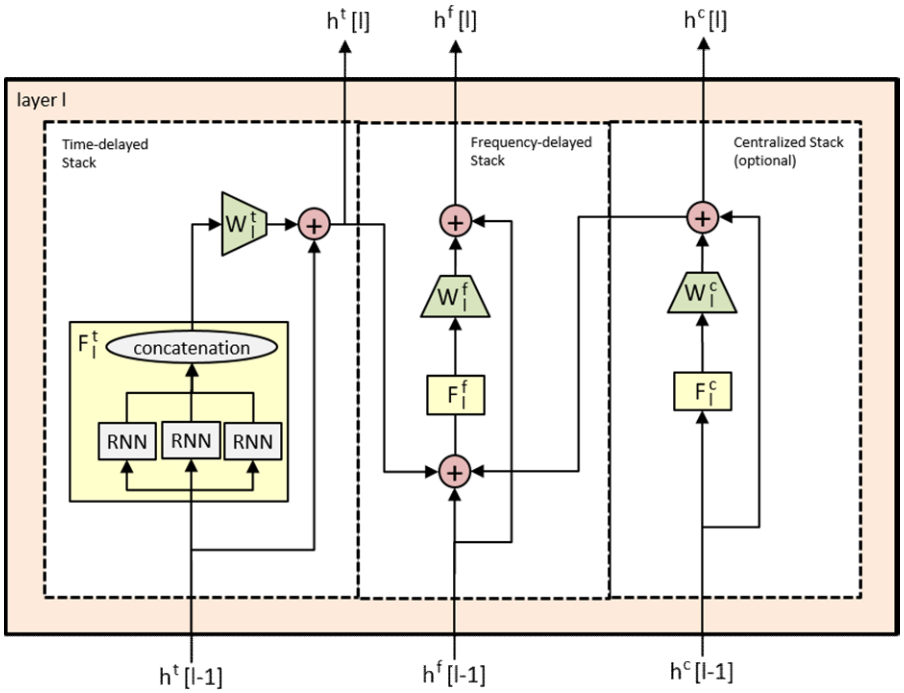
Computation graph of a single layer of the network



RNN used in the Centralized-delayed stack

Source: *Melnet: A generative model for audio in the frequency domain*

MelNet: Single Tier



Computation graph of a single layer of the network

MelNet: Multi-Tier

- In the Single Tier approach the autoregressive ordering is a simple time-major ordering.
- Spectrograms have a high number of dimensions which hinders learning the global structure of spectrograms (because autoregressive models tend to learn the local structure).
- To solve this, MelNet uses a multiscale approach generating spectrograms in a coarse-to-fine order.

Autoregressive generation of spectrograms using a time-major ordering (left) and a multiscale ordering (right).

Source*: https://sjvasquez.github.io/blog/melnet/*

MelNet: Multi-Tier

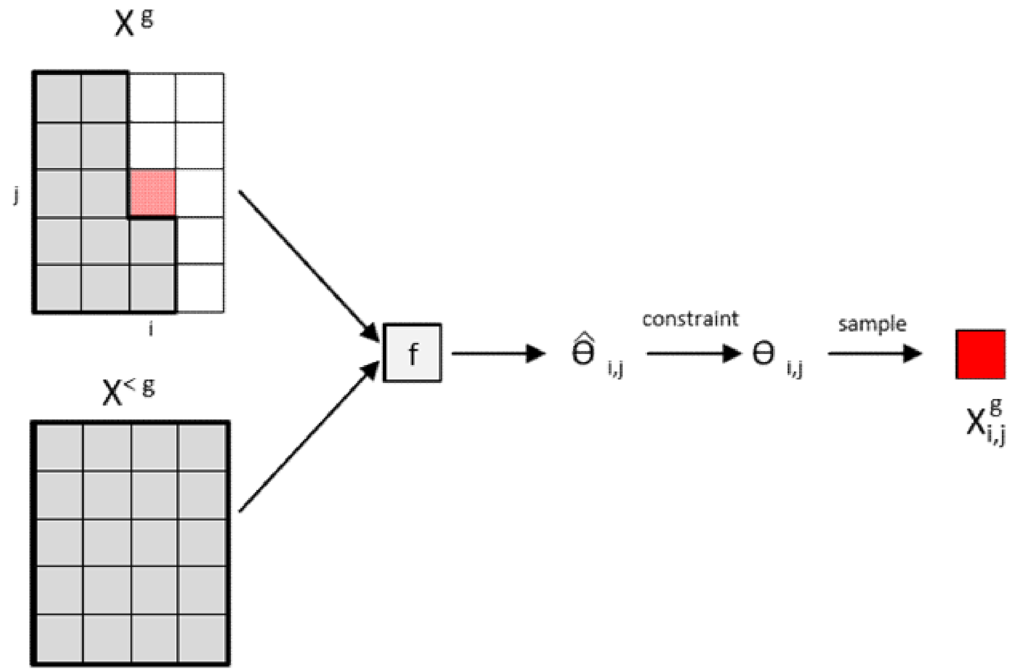
- The elements of a spectrogram x are partitioned into G tiers $x = (x^1, ..., x^G)$.
- We define $x^{<g}$ as the union of all tiers preceding x^g, i.e. $x^{<g} = (x^1, ..., x^{g-1})$.
- The joint distribution of a spectrogram is now factorized over the tiers:

$$p(x;\psi) = \prod_g p(x^g | x^{<g}; \theta^g = f(x^{<g}; \psi^g))$$

- The distribution of each tier is factorized as

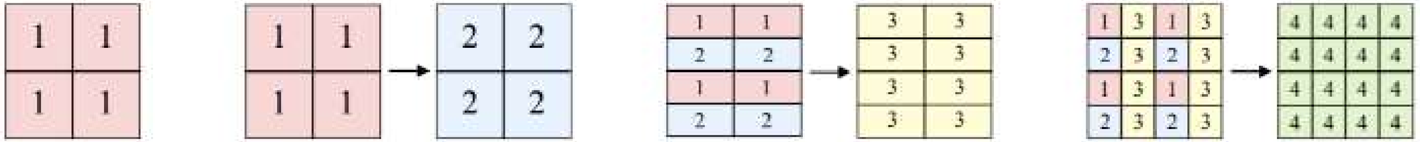
$$p(x^g | x^{<g}; \theta^g = f(x^{<g}; \psi^g)) = \prod_i \prod_j p(x^g_{ij} | x^g_{<ij}, x^{<g}; \theta^g_{ij} = f(x^g_{ij}, x^{<g}; \psi^g))$$

MelNet: Multi-Tier



Process to generate the spectrogram corresponding to x^g value by value.

MelNet: Multi-Tier



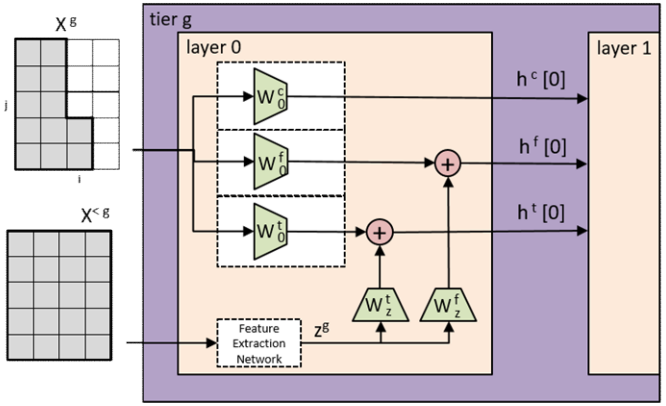
(a) $p(x^1;\ \psi^1)$ (b) $p(x^2 \mid x^1;\ \psi^2)$ (c) $p(x^3 \mid x^1, x^2;\ \psi^3)$ (d) $p(x^4 \mid x^1, x^2, x^3;\ \psi^4)$

Schematic showing how tiers of the multiscale model are interleaved and used to condition the distribution for the subsequent tier.

Source: *Melnet: A generative model for audio in the frequency domain*

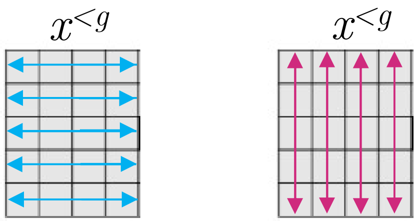
MelNet: Multi-Tier

- The first tier network has the same structure as the single tier network.
  - Now instead of generating a full spectrogram x, it generates x^1 from $p(x^1; \theta^1 = f(\psi^1))$.
- The other tiers have a similar architecture to the first tier, but they need a mechanism to add the information from preceding tiers known as feature extraction network.



Computation graph for the layer 0 of a tier g (g>1)

- The Feature Extraction Network is a multidimensional RNN composed of two one-dimensional RNN running bidirectionally along slices of both axes in x^g.



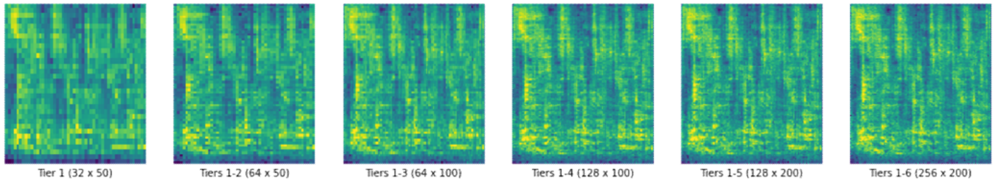
RNNs used in the Feature Extraction Network

MelNet: Implementation and Training

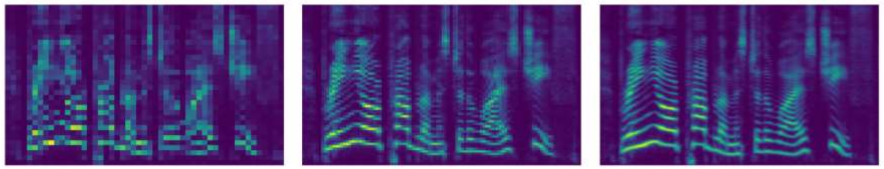
- Implemented using PyTorch.
- Each tier has been trained separately
  - From a real spectrogram y, we separate $y^{<g}$ and y^{g} .
  - We use $y^{<g}$ as the condition to generate the parameters θ^{g}.
  - We compute the loss $loss(y^{g}, \theta^{g})$.
- Tiers were trained on Redsofa-1 (GTX 2080 with 8GB of VRAM)
  - The architecture size is defined by #tiers, #layers and hidden size (RNN hidden state size)
  - Hidden size had to be reduced from 512 (MelNet paper) to 16 to fit in memory
  - Usage of PyTorch checkpointing helps to increase hidden size from 16 to 200
  - To increase batch size, we use gradient accumulation

MelNet: Initial Results

- Initially trained in the Podcast dataset (provided by Éva Székely)



Spectrogram viewed at different stages generated by the initial architecture (from the project).
Architecture: dpodcast_t6_l12.5.4.3.2.2_hd200_gmm10



(a) Tier 1 (32 × 50) (b) Tiers 1–3 (64 × 100) (c) Tiers 1–6 (256 × 200)

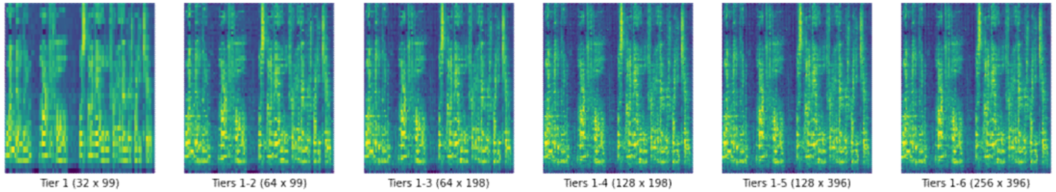
Spectrogram viewed at different stages generated (from the original paper).

Source: *Melnet: A generative model for audio in the frequency domain*

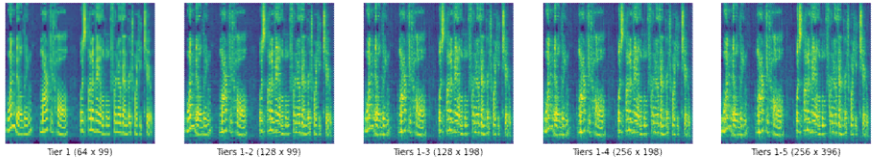
- We trained other models varying the architecture parameters.
- It appears that upsampling layers add detail, but the initial tier is the most important tier because it dictates the high-level structure.

MelNet: Results with Upsampling Layer Only

- Normal synthesis algorithm: first tier generates unconditionally a low-resolution spectrogram and upsampling layers add detail.
- Modified synthesis algorithm: the first tier is an item from the dataset and we use only upsampling layers to add detail.



Architecture:
dljspeech_t6_l0.7.6.5.4.4_hd200_gmm10

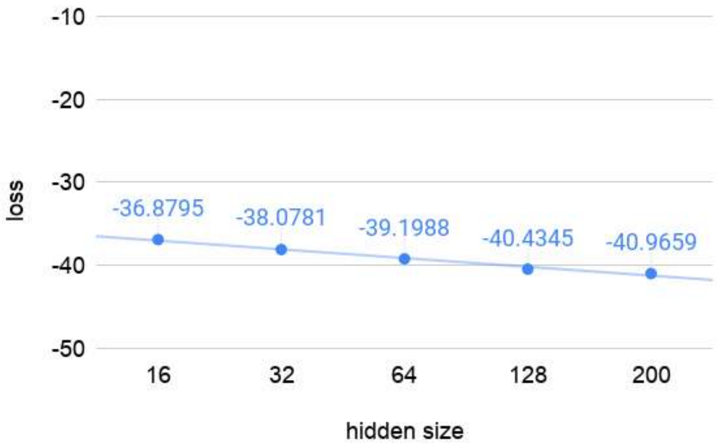


Architecture:
dljspeech_t5_l0.6.5.4.4_hd200_gmm10

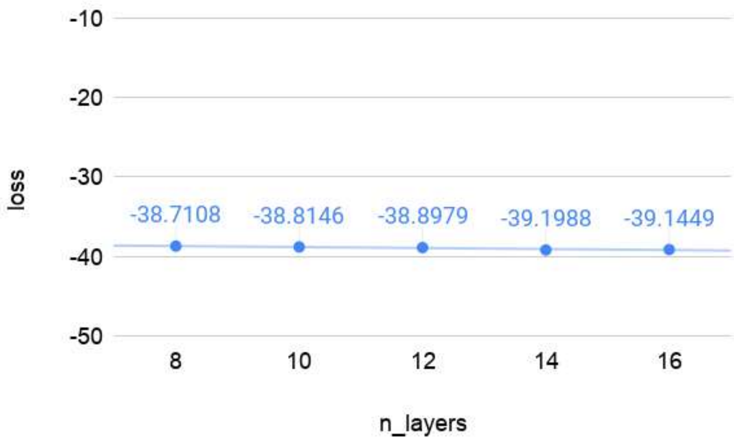
Spectrogram viewed at different stages generated using a real low resolution spectrogram as the output of the first tier.

MelNet: Results with First Tier

- We know First Tier is the most important because it dictates the high-level structure.

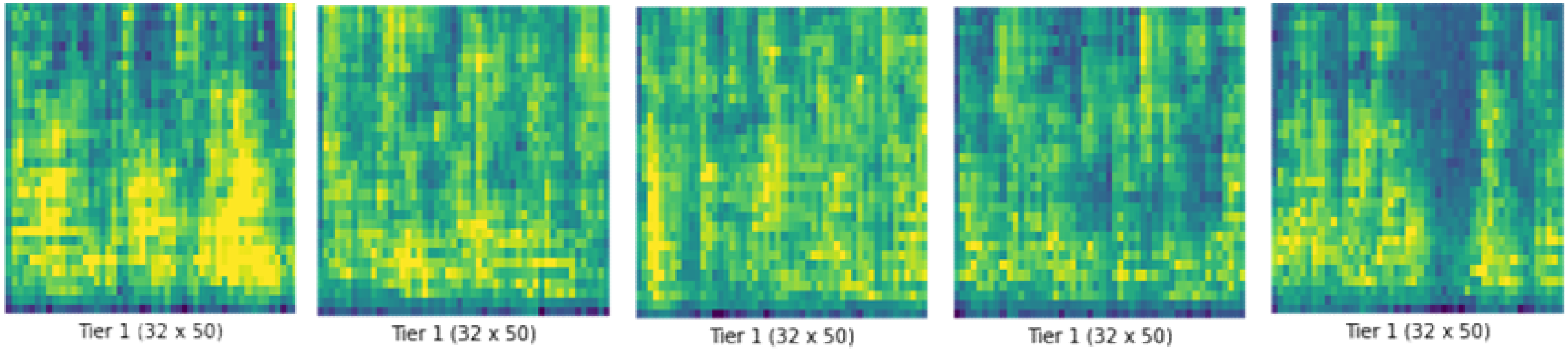


First tier: Hidden size vs. loss.
Architecture: dljspeech_t6_l14.5.4.3.2.2_hdX_gmm10



First tier: number of layers vs. loss.
Architecture: dljspeech_t6_lX.5.4.3.2.2_hd64_gmm10.

MelNet: Results with First Tier

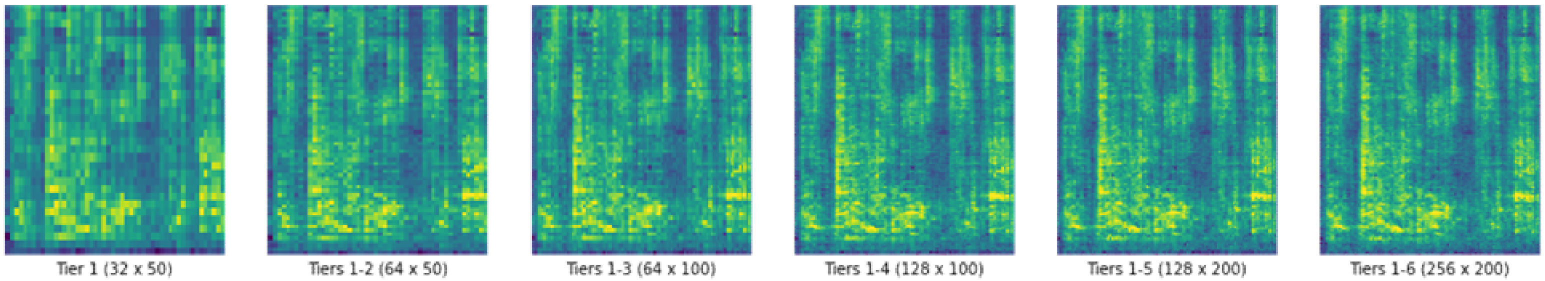


(a) hd = 16 (b) hd = 32 (c) hd = 64 (d) hd = 128 (e) hd = 200

Spectrogram generated by the first tier with different hidden size.
Architecture: dljspeech_t6_l14.5.4.3.2.2_hdX_gmm10.

MelNet: Final Results

- First tier is the most important tier to generate realistic spectrograms because it dictates the high-level structure.
- Bigger models, especially with bigger hidden sizes, produce better spectrograms.
- We trained the biggest model possible on LJSpeech dataset:



Spectrogram viewed at different stages.
Architecture: dljspeech_t6_l12.7.6.5.4.4_hd200_gmm10