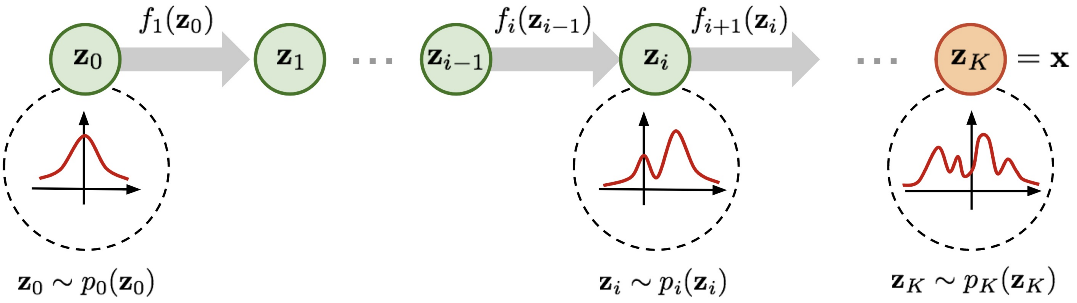
WAVEGLOW: A FLOW-BASED GENERATIVE NETWORK FOR SPEECH SYNTHESIS

date 16.02.2018, paper on arxiv (https://arxiv.org/pdf/1712.05884.pdf)

It is recommended to check tacotron2 review first, since some concepts were explained there.

Model overview

Waveglow is a normalizing flow model for generating war audio conditioned on mel spectograms. Flow consists from K bijection functions $f = f_1 \circ f_2 \circ \cdots \circ f_K$ such that the relationship between data x and random noise z_0 modeled as:


$f_1(\mathbf{z}_0)$
$\mathbf{z}_0$
$\mathbf{z}_1$
$f_i(\mathbf{z}_{i-1})$
$\mathbf{z}_{i-1}$
$\mathbf{z}_i$
$f_{i+1}(\mathbf{z}_i)$
$\mathbf{z}_K = \mathbf{x}$
$\mathbf{z}_0 \sim p_0(\mathbf{z}_0)$
$\mathbf{z}_i \sim p_i(\mathbf{z}_i)$
$\mathbf{z}_K \sim p_K(\mathbf{z}_K)$

Given an observed data variable $x \in X$, a simple prior probability distribution p_Z (zero mean spherical Gaussian) on a latent variable $z \in Z$, and a bijection $f: X \rightarrow Z$ (with $g = f^{-1}$), the change of variable formula defines a model distribution on X by

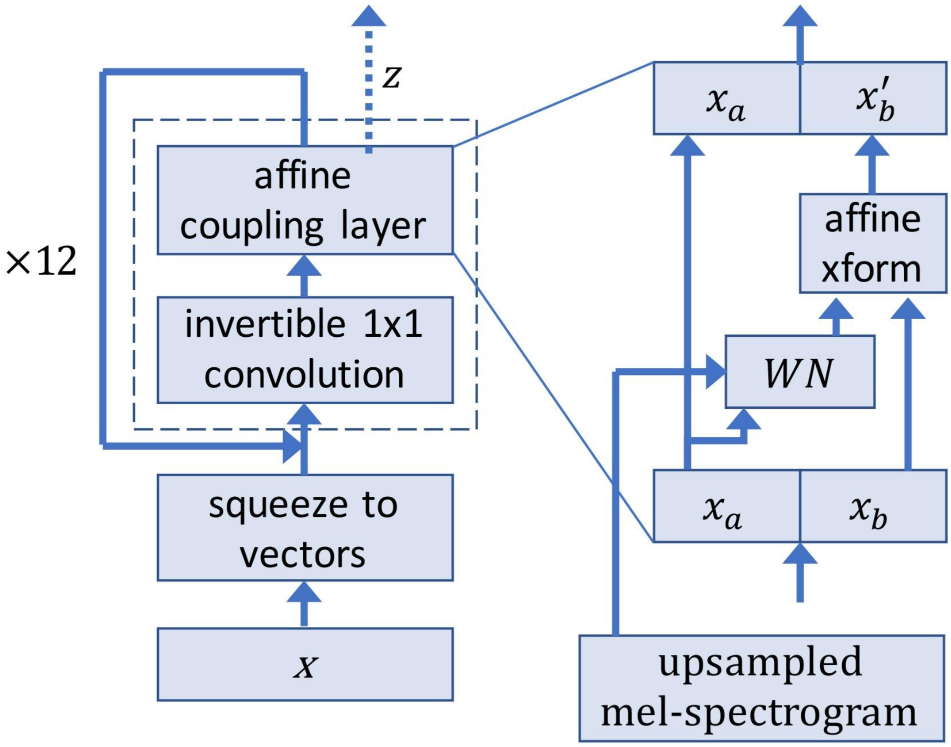
$$p_X(x) = p_Z(f(x)) \left| \det\left(\frac{\partial f(x)}{\partial x^T}\right) \right|$$

$$\log\left(p_X(x)\right) = \log\left(p_Z(f(x))\right) + \log\left(\left| \det\left(\frac{\partial f(x)}{\partial x^T}\right) \right|\right)$$

where $\frac{\partial f(x)}{\partial x^T}$ is the Jacobian of f at x.

Model consists of several blocks:

- squeeze to vectors;
- invertible 1x1 convolution;
- affine coupling layer with wavenet like NN.



Squeeze to vectors operation

X here is raw audio it could be interpreted as 1-dimensional vector. Authors proposed using special type of reshaping 1-dimensinal vector to 2-dim tensor via torch.unfold operation. It takes all slices from 1-dimensional tensor of shape 8 with step size 8 (NB! torch.unfold operation is not bijection is common case, since it changes the total numbers of elements in tensor, but in these case it is bijective).

After this operation we treat new dimesion (equal to 8) as channel dimension.

Invertible 1x1 convolution

This is simple 1-dimensional convolution done over channel dimension, it is initialized as orhtonormal martix, and no LUP decomposition is used (since small weight matrix shape e.g. 8x8). However, as far as i can see, in such case there is no guarantees about being invertible during training. Authors says that "it is guaranteed by loss", but no other guarantees are given. Since loss consist of the minus log determinant of weight matrix and matrix invertion criterion it seems to be enough.

Affine Coupling Layer with wavenet like NN.

It is usual affine coupling layer. The only two tricks are to upsample mel spectrogram in time domain (it is huge upsample from say ten to ten thousands) and adding skip and residual connections.

Early outputs

Authors proposed to output latent vector z not on last step, but emit it parts every $k = 4$ flows outputs. So final z is concatenation of parts after 4, 8 and 12 flow. After 4 and 8 flow only two first channels were taken, after 12 remaining 4 channels. This leads to change of shape of z along the flow between 1-4 flow we have 8 channels, between 5-8 6 channels, remaining flows working with 4 channels. Authors mention that it helps training and propogating gradients to early layers.

Training and inference

Training is done via directly maximizing log likelihood with respect to flow parameters. Inference could be done via inverse propogation through the flow with noise as input and corresponding mel spectrograms.

Results

Model achieved better mean opinion score than wavenet and Griffin-Lim alrotihm being much more faster than wavenet at inference.

```
In [ ]:
```