Parallel WaveNet:Fast High-Fidelity Speech Synthesis

date 28.11.2017, paper on arxiv (https://arxiv.org/pdf/1711.10433.pdf)

Is highly recommended to read wavenet review (https://gitlab.com/dasha.ai/ml-team/ai-docs/blob/7-tts-literature-review/papers/TTS/wavenet.md) first

Model overview

Modification of base wavenet (more fidelity)

In this article wavenet model was used for training parallel wavenet model. Wavenet model was modified to predict not 8 bit output, but 16 bit output via changing output from softmax to a sample from mix of logistic distribution. Model predicts instead of vector of probabitiles $p(x_t) \in R^{256}$ parameters of mix of logistic distribution: $\pi^i(x_t) = \pi^i_t,\ \mu^i(x_t) = \mu^i_t,\ s^i(x_t) = s^i_t$ and x_t calculated via sampling:

$$x_t \sim \sum_{i=1}^{K} \pi^i_t logistic(\mu^i_t, s^i_t)$$

Inverse Autoregressive Flows

Inverse Autoregressive Flows is a type of Normalizing Flows (NF) - bayesian generative models with explicit likelihood, ability to calculate aposterior density and effective at sampling (inference).

Fast recap about Normalizing Flows

Normalizing flow is a sequence of bijective transformations $f_k, k \in \{1, \ldots, K\}$ mapping random sample z_0 from some explicit distribution (e.g. Gaussian) to point from unknown data distribution using samples from this distributions for learning. The main formula for learing such transofmations is change of variable formula (if $z' = f(z)$, and z has distribution $q(z)$):

$$q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{z}'} \right| = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}$$

Using this formula we can stack multiple transformation to model complex data distributions:

$$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}_0)$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|$$

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

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

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

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

It could be trained via directly optimizing $\log\big(p_X(x)\big)$ using stohastic gradient methods.

Exact samples from the resulting distribution can be generated by using the inverse transform sampling rule. A sample $z \sim p_Z$ is drawn in the latent space, and its inverse image $x = f^{-1}(z) = g(z)$ generates a sample in the original space. Computing the density on a point x is accomplished by computing the density of its image $f(x)$ and multiplying by the associated Jacobian determinant

$\det\left(\frac{\partial f(x)}{\partial x^T}\right).$

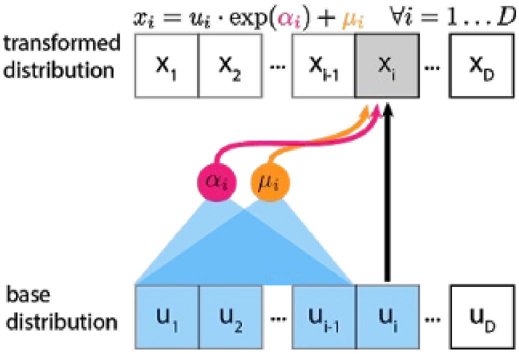
Inverse Autoregressive Flows

IAF uses following type of transformation random sample $z_t \sim Logistic(0,1)$ to a data point x_t (assuming $x_0 = z_0$):

$$x_t = z_t \cdot s(z_{<t}, \theta) + \mu(z_{<t}, \theta)$$

where $s(z_{<t}, \theta)$ and $\mu(z_{<t}, \theta)$ can be any autoregressive model, in article the same convolutional autoregressive network structure as the original WaveNet was used (actually there were 4 stacked transformations).

To make things clear lets check it more detailed: To output the correct distribution for time-step x_t, the inverse autoregressive flow can implicitly infer what it would have output at previous time-steps $x_1, \ldots, x_{t-1}$ based on the noise inputs $z_1, \ldots, z_{t-1}$, which allows it to output all x_t in parallel given z_t. The image below would make things lot clearer (assume $\alpha_i = s(z_{<i}, \theta)$, $\mu_i = \mu(z_{<i}, \theta)$ - outputs of neural networks).



As it was mentioned earlier any NF could be trained via directly optimizing likelihood, nevertheless, this way is time consuming, so authors proposed new method of training model called Probability Density Distillation.

Condtitioning (TTS)

Condition vector (e.g., linguistic features, speaker ID, …) could be processed like in wavenet model, since $s(z_{<t}, \theta)$ and $\mu(z_{<t}, \theta)$ are wavenet like model.

Training (Probability Density Distillation).

The base idea is to take trained WaveNet model (called teacher) and to minimize KL divergence between distribution given by teacher network and parallel WaveNet model (called student).

$$D_{\mathrm{KL}}\left(P_S || P_T\right) = H(P_S, P_T) - H(P_S),$$

where P_S and P_T are student and teacher distributions.

In article following formulas were derived:

$$H(P_S) = \mathop{\mathrm{E}}_{z \sim L(0,1)} \left[\sum_{t=1}^{T} \ln s(z_{<t}, \theta)\right] + 2T,$$

$$H(P_S, P_T) = \sum_{t=1}^{T} \mathop{\mathrm{E}}_{p_S(x_{<t})} H\left(p_S(x_t | x_{<t}), p_T(x_t | x_{<t})\right).$$

For every sample x we draw from the student p_S we can compute all $p_T(x_t | x_{<t})$ in parallel with the teacher and then evaluate $H(p_S(x_t | x_{<t}), p_T(x_t | x_{<t}))$ very efficiently by drawing multiple different samples x_t from $p_S(x_t | x_{<t})$ for each timestep. Because the teacher's output distribution $p_T(x_t | x_{<t})$ is parameterised as a mixture of logistics distribution, the loss term $\ln p_T(x_t | x_{<t})$ is differentiable with respect to both x_t and $x_{<t}$. A categorical distribution, on the other hand, would only be differentiable w.r.t. $x_{<t}$.

Beside main loss (KL divergence) there were used Power loss (l2 norm between Short Term Fourier Trasform (STFT) of generated sample conditioned on linguistic features and target), Perceptual loss (instead of STFT feauteres from CNN classifier was used, like content loss in image style transfer), Contrastive loss (which minimises the KL-divergence between the teacher and student when both are conditioned on the same information c_1 (e.g., linguistic features, speaker ID, …), but also maximises it for different conditioning pairs $c1 \neq c2$)

Inference

Inference could be done in parallel via forward passing each random sample z_t and condition vector h_t (like batch in CV model).

Experiements

In all text to speech experiements model was conditioned on linguistic features, providing phonetic and duration information to the network, and pitch information (logarithm of the fundamental frequency) predicted by a different model. Model received same score as WaveNet model with 16 bit output, being much 1000 times faster and both Parallel WaveNet with 16 bit output and WaveNet with 16 bit output highly outperforms WaveNet with 8 bit output in terms of MOS score.