Parallel WaveNet: Fast High-Fidelity Speech Synthesis

Aaron van den Oord [1] Yazhe Li [1] Igor Babuschkin [1] Karen Simonyan [1] Oriol Vinyals [1] Koray Kavukcuoglu [1]
George van den Driessche [1] Edward Lockhart [1] Luis C. Cobo [1] Florian Stimberg [1] Norman Casagrande [1]
Dominik Grewe [1] Seb Noury [1] Sander Dieleman [1] Erich Elsen [1] Nal Kalchbrenner [1] Heiga Zen [1] Alex Graves [1]
Helen King [1] Tom Walters [1] Dan Belov [1] Demis Hassabis [1]

Abstract

The recently-developed *WaveNet* architecture (van den Oord et al., 2016a) is the current state of the art in realistic speech synthesis, consistently rated as more natural sounding for many different languages than any previous system. However, because WaveNet relies on sequential generation of one audio sample at a time, it is poorly suited to today's massively parallel computers, and therefore hard to deploy in a real-time production setting. This paper introduces *Probability Density Distillation*, a new method for training a parallel feed-forward network from a trained WaveNet with no significant difference in quality. The resulting system is capable of generating high-fidelity speech samples at more than 20 times faster than real-time, a 1000x speed up relative to the original WaveNet, and capable of serving multiple English and Japanese voices in a production setting.

1. Introduction

Recent successes of deep learning go beyond achieving state-of-the-art results in research benchmarks, and push the frontiers in some of the most challenging real world applications such as speech recognition (Hinton et al., 2012), image recognition (Krizhevsky et al., 2012; Szegedy et al., 2015), and machine translation (Wu et al., 2016). The recently published WaveNet (van den Oord et al., 2016a) model achieves state-of-the-art results in speech synthesis, and significantly closes the gap with natural human speech. However, it is not well suited for real world deployment due to its prohibitive generation speed. In this paper, we present a new algorithm for distilling WaveNet into a feed-forward neural network which can synthesise equally high quality speech much more efficiently, and is deployed to millions of users.

WaveNet is one of a family of autoregressive deep generative models that have been applied with great success to data as diverse as text (Mikolov et al., 2010), images (Larochelle & Murray, 2011; Theis & Bethge, 2015; van den Oord et al., 2016c;b), video (Kalchbrenner et al., 2016), handwriting (Graves, 2013) as well as human speech and music. Modelling raw audio signals, as WaveNet does, represents a particularly extreme form of autoregression, with up to 24,000 samples predicted per second. Operating at such a high temporal resolution is not problematic during network training, where the complete sequence of input samples is already available and—thanks to the convolutional structure of the network—can be processed in parallel. When generating samples, however, each input sample must be drawn from the output distribution before it can be passed in as input at the next time step, making parallel processing impossible.

Inverse autoregressive flows (IAFs) (Kingma et al., 2016) represent a kind of dual formulation of deep autoregressive modelling, in which sampling can be performed in parallel, while the inference procedure required for likelihood estimation is sequential and slow. The goal of this paper is to marry the best features of both models: the efficient training of WaveNet and the efficient sampling of IAF networks. The bridge between them is a new form of neural network distillation (Hinton et al., 2015), which we refer to as *Probability Density Distillation*, where a trained WaveNet model is used as a teacher for a feedforward IAF model.

The next section describes the original WaveNet model, while Sections 3 and 4 define in detail the new, parallel version of WaveNet and the distillation process used to transfer knowledge between them. Section 5 then presents experimental results showing no loss in perceived quality for parallel versus original WaveNet, and continued superiority over previous benchmarks. We also present timings for sample generation, demonstrating more than 1000× speed-up relative to original WaveNet.

[1] DeepMind Technologies, London, United Kingdom. Correspondence to: Aaron van den Oord <avdnoord@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

2. WaveNet

Autoregressive networks model the joint distribution of high-dimensional data as a product of conditional distributions using the probabilistic chain-rule:

$$p(\boldsymbol{x}) = \prod_t p(x_t | x_{<t}, \boldsymbol{\theta}),$$

where x_t is the t-th variable of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ are the parameters of the autoregressive model. The conditional distributions are usually modelled with a neural network that receives $x_{<t}$ as input and outputs a distribution over possible x_t.

WaveNet (van den Oord et al., 2016a) is a convolutional autoregressive model which produces all $p(x_t|x_{<t})$ in one forward pass, by making use of *causal*—or *masked*—convolutions (van den Oord et al., 2016c; Germain et al., 2015). Every causal convolutional layer can process its input in parallel, making these architectures very fast to train compared to RNNs (van den Oord et al., 2016b), which can only be updated sequentially. At generation time, however, the waveform has to be synthesised in a sequential fashion as x_t must be sampled first in order to obtain $x_{>t}$. Due to this nature, real time (or faster) synthesis with a fully autoregressive system is challenging. While sampling speed is not a significant issue for offline generation, it is essential for real-word applications. A version of WaveNet that generates in real-time has been developed (Paine et al., 2016), but it required the use of a much smaller network, resulting in severely degraded quality.

Raw audio data is typically very high-dimensional (e.g. 16,000 samples per second for 16kHz audio), and contains complex, hierarchical structures spanning many thousands of time steps, such as words in speech or melodies in music. Modelling such long-term dependencies with standard causal convolution layers would require a very deep network to ensure a sufficiently broad receptive field. WaveNet avoids this constraint by using *dilated* causal convolutions, which allow the receptive field to grow exponentially with depth.

WaveNet uses gated activation functions, together with a simple mechanism introduced in (van den Oord et al., 2016c) to condition on extra information such as class labels or linguistic features:

$$\boldsymbol{h}_i = \sigma\left(W_{g,i} * \boldsymbol{x}_i + V_{g,i}^T \boldsymbol{c}\right) \odot \tanh\left(W_{f,i} * \boldsymbol{x}_i + V_{f,i}^T \boldsymbol{c}\right), \tag{1}$$

where $*$ denotes a convolution operator, and $\odot$ denotes an element-wise multiplication operator. $\sigma(\cdot)$ is a logistic sigmoid function. $\boldsymbol{c}$ represents extra conditioning data. i is the layer index. f and g denote filter and gate, respectively. W and V are learnable weights. In cases where $\boldsymbol{c}$ encodes spatial or sequential information (such as a sequence of linguistic features), the matrix products ($V_{f,i}^T \boldsymbol{c}$ and $V_{g,i}^T \boldsymbol{c}$) are replaced by convolutions ($V_{f,i} * \boldsymbol{c}$ and $V_{g,i} * \boldsymbol{c}$).

2.1. Higher Fidelity WaveNet

For this work we made two improvements to the basic WaveNet model to enhance its audio quality for production use. Unlike previous versions of WaveNet (van den Oord et al., 2016a), where 8-bit (μ-law or PCM) audio was modelled with a 256-way categorical distribution, we increased the fidelity by modelling 16-bit audio. Since training a 65,536-way categorical distribution would be prohibitively costly, we instead modelled the samples with the discretized mixture of logistics distribution introduced in (Salimans et al., 2017). We further improved fidelity by increasing the audio sampling rate from 16kHz to 24kHz. This required a WaveNet with a wider receptive field, which we achieved by increasing the dilated convolution filter size from 2 to 3. An alternative strategy would be to increase the number of layers or add more dilation stages.

3. Parallel WaveNet

While the convolutional structure of WaveNet allows for rapid parallel training, sample generation remains inherently sequential and therefore slow, as it is for all autoregressive models which use ancestral sampling. We therefore seek an alternative architecture that will allow for rapid, parallel generation.

Inverse-autoregressive flows (IAFs) (Kingma et al., 2016) are stochastic generative models whose latent variables are arranged so that all elements of a high dimensional observable sample can be generated in parallel. IAFs are a special type of normalising flow (Dinh et al., 2014; Rezende & Mohamed, 2015; Dinh et al., 2016) which model a multivariate distribution $p_X(\boldsymbol{x})$ as an explicit invertible non-linear transformation $\boldsymbol{x} = f(\boldsymbol{z})$ of a simple tractable distribution $p_Z(\boldsymbol{z})$ (such as an isotropic Gaussian distribution). Using the change of variables formula the resulting distribution can be written as:

$$\log p_X(\boldsymbol{x}) = \log p_Z(\boldsymbol{z}) - \log\left|\frac{d\boldsymbol{x}}{d\boldsymbol{z}}\right|,$$

where $\left|\frac{d\boldsymbol{x}}{d\boldsymbol{z}}\right|$ is the determinant of the Jacobian of f. For all normalizing flows the transformation f is chosen so that it is invertible and its Jacobian determinant is easy to compute. In the case of an IAF, the output is modelled by $x_t = f(\boldsymbol{z}_{\leq t})$. Because of this strict dependency structure, the transformation has a triangular Jacobian matrix which makes the determinant equal to the product of the diagonal entries:

$$\log\left|\frac{d\boldsymbol{x}}{d\boldsymbol{z}}\right| = \sum_t \log \frac{\partial f(\boldsymbol{z}_{\leq t})}{\partial z_t}.$$

To sample from an IAF, a random sample is first drawn from $\boldsymbol{z} \sim p_Z(\boldsymbol{z})$ (we use the $\mathrm{Logistic}(0, I)$ distribution) which

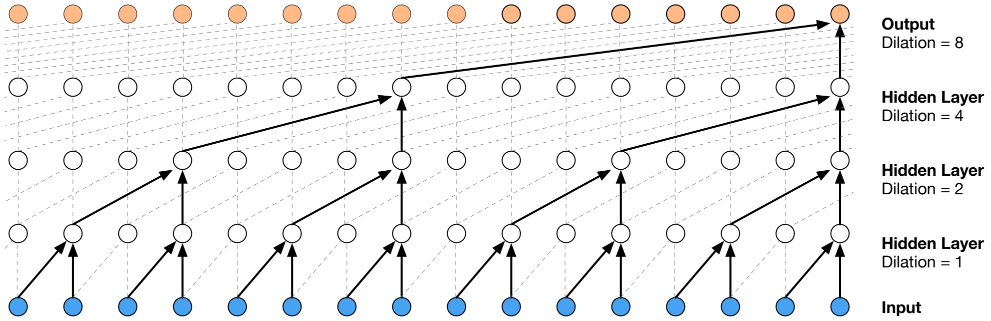

Figure 1. Visualisation of a WaveNet stack and its receptive field (van den Oord et al., 2016a). Starting from the inputs at the bottom, the WaveNet architecture has increasing levels of dilation by a factor of 2, so that each output unit—shown at the top row of the figure— can combine dependency from a large range of inputs.

is then transformed as follows:

$$x_t = z_t \cdot s(\boldsymbol{z}_{<t}, \boldsymbol{\theta}) + \mu(\boldsymbol{z}_{<t}, \boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{s}$ are outputs by the network. Therefore, $p(x_t|\boldsymbol{z}_{<t})$ follows a logistic distribution parameterised by μ_t and s_t.

$$p(x_t|\boldsymbol{z}_{<t}, \boldsymbol{\theta}) = \mathbb{L}\left(x_t \middle| \mu(\boldsymbol{z}_{<t}, \boldsymbol{\theta}), s(\boldsymbol{z}_{<t}, \boldsymbol{\theta})\right),$$

While $\mu(\boldsymbol{z}_{<t}, \boldsymbol{\theta})$ and $s(\boldsymbol{z}_{<t}, \boldsymbol{\theta})$ can be any model, we use the same convolutional network structure as the original WaveNet (van den Oord et al., 2016a).

Autoregressive models (or flows (Papamakarios et al., 2017)) model the data as $p(x_t|x_{<t})$ and IAFs as $p(x_t|z_{<t})$. If these models share the same output distribution class (e.g., mixture of logistics or categorical) then mathematically they should be able to model the same multivariate distributions. However, in practice there are some differences (see Section 3.1). To output the correct distribution for timestep x_t, the inverse autoregressive flow can implicitly infer what it would have output at previous timesteps $x_1, \ldots, x_{t-1}$ based on the noise inputs $z_1, \ldots, z_{t-1}$, which allows it to output all x_t in parallel given z_t.

In general, normalising flows might require repeated iterations to transform uncorrelated noise into structured samples, with the output generated by the flow at each iteration passed in as input at the next (Rezende & Mohamed, 2015). This is less crucial for IAFs, as the autoregressive latents can induce significant structure in a single pass. Nonetheless we observed that having up to 4 flow iterations did improve the quality (the weights are not shared between the flows).

The first (bottom) network takes as input the white unconditional logistic noise: $\boldsymbol{z}^0$. Thereafter the output of each network i is passed as input to the next network $i+1$, which again transforms it.

$$\boldsymbol{z}^i = \boldsymbol{z}^{\boldsymbol{i-1}} \cdot \boldsymbol{s}^i + \boldsymbol{\mu}^i \tag{3}$$

Because we use the same ordering in all the flows, the final distribution $p(x_t|z_{<t}, \boldsymbol{\theta})$ is still logistic with location μ_{tot} and scale s_{tot}:

$$\boldsymbol{\mu}_{\text{tot}} = \sum_i^N \boldsymbol{\mu}^i \left(\prod_{j>i}^N \boldsymbol{s}^j \right) \tag{4}$$

$$\boldsymbol{s}_{\text{tot}} = \prod_i^N \boldsymbol{s}_i \tag{5}$$

where N is the number of flows and the dependencies on t and z are omitted for simplicity.

3.1. Autoregressive Models and Inverse-autoregressive Flows

Although inverse-autoregressive flows (IAFs) and autoregressive models can in principle model the same distributions (Chen et al., 2016), they have different inductive biases and may vary greatly in their capacity to model certain processes. As a simple example consider the Fibonacci series $(1, 1, 2, 3, 5, 8, 13, \ldots)$. For an autoregressive model this is easy to model with a receptive field of two: $f(k) = f(k-1) + f(k-2)$. For an IAF, however, the receptive field needs to be at least size k to correctly model k terms, leading to a larger model that is less able to generalise.

4. Probability Density Distillation

Training the parallel WaveNet model directly with maximum likelihood would be impractical, as the inference procedure required to estimate the log-likelihoods is sequential and slow[1]. We therefore introduce a novel form of neural network distillation (Hinton et al., 2015) that uses an al-

[1] In this sense the two architectures are dual to one another: slow training and fast generation with parallel WaveNet versus fast training and slow generation with WaveNet.

ready trained WaveNet as a 'teacher' from which a parallel WaveNet 'student' can efficiently learn. To stress the fact that we are dealing with normalised density models, we refer to this process as *Probability Density Distillation* (in contrast to Probability Density Estimation). The basic idea is for the student to attempt to match the probability of its own samples under the distribution learned by the teacher.

Given a parallel WaveNet student $p_S(\boldsymbol{x})$ and WaveNet teacher $p_T(\boldsymbol{x})$ which has been trained on a dataset of audio, we define the *Probability Density Distillation* loss as follows:

$$D_{\text{KL}}\left(P_S \| P_T\right) = H(P_S, P_T) - H(P_S) \tag{6}$$

where D_{KL} is the Kullback–Leibler divergence, and $H(P_S, P_T)$ is the cross-entropy between the student P_S and teacher P_T, and $H(P_S)$ is the entropy of the student distribution. When the KL divergence becomes zero, the student distribution has fully recovered the teacher's distribution. The entropy term (which is not present in previous distillation objectives (Hinton et al., 2015)) is vital in that it prevents the student's distribution from collapsing to the mode of the teacher (which, counter-intuitively, does not yield a good sample—see Section 4.1). Crucially, all the operations required to estimate derivatives for this loss (sampling from $p_S(\boldsymbol{x})$, evaluating $p_T(\boldsymbol{x})$, and evaluating $H(P_S)$) can be performed efficiently, as we will see.

It is worth noting the parallels to Generative Adversarial Networks (GANs (Goodfellow et al., 2014)), with the student playing the role of generator, and the teacher playing the role of discriminator. As opposed to GANs, however, the student is not attempting to fool the teacher in an adversarial manner; rather it co-operates by attempting to match the teacher's probabilities. Furthermore the teacher is held constant, rather than being trained in tandem with the student, and both models yield tractable normalised distributions.

Recently (Gu et al., 2017) has presented a related idea to train feed-forward networks for neural machine translation. Their method is based on conditioning the feedforward decoder on fertility values, which require supervision by an external alignment system. The training procedure also involves the creation of an additional dataset as well as fine-tuning. During inference, their model relies on re-scoring by an auto-regressive model.

First, observe that the entropy term $H(P_S)$ in Equation 6 can be rewritten as follows:

$$H(P_S) = \mathop{\mathbb{E}}_{z\sim L(0,1)}\left[\sum_{t=1}^{T} -\ln p_S(x_t|\boldsymbol{z}_{<t})\right] \tag{7}$$

$$= \mathop{\mathbb{E}}_{z\sim L(0,1)}\left[\sum_{t=1}^{T} \ln s(\boldsymbol{z}_{<t},\boldsymbol{\theta})\right] + 2T, \tag{8}$$

where $\boldsymbol{x} = g(\boldsymbol{z})$ and z_t are independent samples drawn from the logistic distribution. The second equality in Equation 8 follows because the entropy of a logistic distribution $\mathbb{L}(\mu, s)$ is $\ln s + 2$. We can therefore compute this term without having to explicitly generate $\boldsymbol{x}$.

The cross-entropy term $H(P_S, P_T)$ however explicitly depends on $\boldsymbol{x} = g(\boldsymbol{z})$, and therefore requires sampling from the student to estimate.

$$H(P_S, P_T) = \int_{\boldsymbol{x}} p_S(\boldsymbol{x}) \ln p_T(\boldsymbol{x}) \tag{9}$$

$$= \sum_{t=1}^{T} \int_{\boldsymbol{x}} p_S(\boldsymbol{x}) \ln p_T(x_t|\boldsymbol{x}_{<t}) \tag{10}$$

$$= \sum_{t=1}^{T} \int_{\boldsymbol{x}} p_S(\boldsymbol{x}_{<t}) p_S(\boldsymbol{x}_{\geq t}|\boldsymbol{x}_{<t}) \ln p_T(x_t|\boldsymbol{x}_{<t}) \tag{11}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_S(\boldsymbol{x}_{<t})} \Bigg[\int_{x_t} p_S(x_t|\boldsymbol{x}_{<t}) \ln p_T(x_t|\boldsymbol{x}_{<t}) \tag{12}$$

$$\int_{\boldsymbol{x}_{>t}} p_S(\boldsymbol{x}_{>t}|\boldsymbol{x}_{\leq t}) \Bigg] \tag{13}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_S(\boldsymbol{x}_{<t})} H\Big(p_S(x_t|\boldsymbol{x}_{<t}), p_T(x_t|\boldsymbol{x}_{<t})\Big). \tag{14}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{\boldsymbol{z}\sim L \\ \boldsymbol{x}=g(\boldsymbol{z})}} H\Big(p_S(x_t|\boldsymbol{z}_{<t}), p_T(x_t|x_{<t})\Big). \tag{15}$$

In Equation 11 we apply the chain rule to mathematicaly decompose $P_S(\boldsymbol{x})$ into conditional distributions but they are only explicitly constructed with a neural network to depend on $\boldsymbol{z}$ as in Equation 15.

For every sample $\boldsymbol{x}$ we draw from the student p_S we can compute all $p_T(x_t|\boldsymbol{x}_{<t})$ in parallel with the teacher and then evaluate $H(p_S(x_t|\boldsymbol{z}_{<t}), p_T(x_t|\boldsymbol{x}_{<t}))$ very efficiently by drawing multiple different samples x_t from $p_S(x_t|\boldsymbol{z}_{<t})$ for each timestep. This unbiased estimator has a much lower variance than naively evaluating the sample under the teacher with Equation 9.

We parameterise the teacher's output distribution $p_T(x_t|\boldsymbol{x}_{<t})$ as a mixture of logistics distribution (Salimans et al., 2017), which allows the loss term $\ln p_T(x_t|\boldsymbol{x}_{<t})$ to be differentiable with respect to both x_t and $x_{<t}$. A categorical distribution, on the other hand, would only be differentiable w.r.t. $\boldsymbol{x}_{<t}$.

4.1. Argument against MAP estimation

In this section we make an argument against *maximum a posteriori* (MAP) estimation for distillation; similar arguments have been made by previous authors in a different setting (Sønderby et al., 2016).

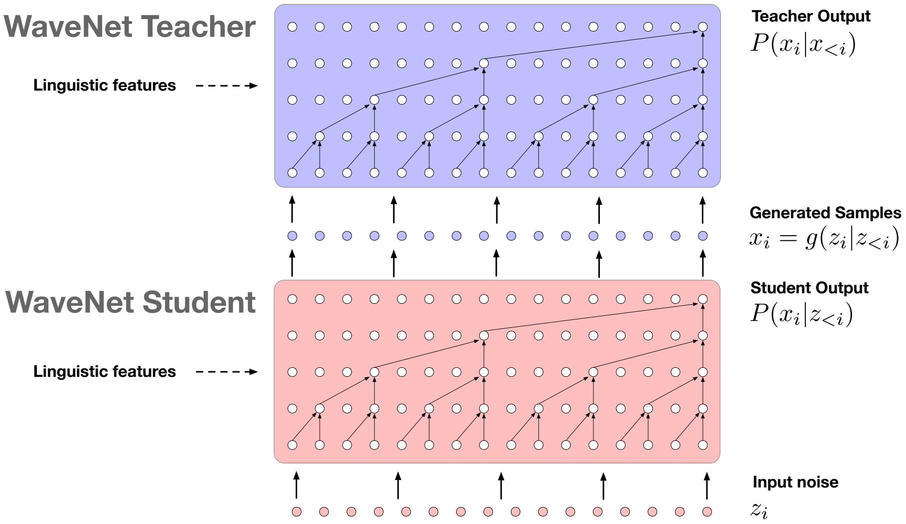

Figure 2. **Overview of Probability Density Distillation**. A pre-trained WaveNet teacher is used to score the samples $\boldsymbol{x}$ output by the student. The student is trained to minimise the KL-divergence between its distribution and that of the teacher by maximising the log-likelihood of its samples under the teacher and maximising its own entropy at the same time.

The distillation loss defined in Section 4 minimises the KL divergence between the teacher and generator. We could instead have minimised only the cross-entropy between the teacher and generator (the standard distillation loss term (Hinton et al., 2015)), so that the samples by the generator are as likely as possible according to the teacher. Doing so would give rise to MAP estimation. Counter-intuitively, audio samples obtained through MAP estimation do not sound as good as typical examples from the teacher: in fact they are almost completely silent, even if using conditional information such as linguistic features. This effect is not due to adversarial behaviour on the part of the teacher, but rather is a fundamental property of the data distribution which the teacher has approximated.

As an example consider the simple case where we have audio from a white random noise source: the distribution at every timestep is $\mathcal{N}(0, 1)$, regardless of the samples at previous timesteps. White noise has a very specific and perceptually recognizable sound: a continual hiss. The MAP estimate of this data distribution, and thus of any generative model that matches it well, recovers the distribution mode, which is 0 at every timestep: i.e. complete silence. More generally, any highly stochastic process is liable to have a 'noiseless' and therefore atypical mode. For the KL divergence the optimum is to recover the full teacher distribution. This is clearly different from any random sample from the distribution. Furthermore, if one changes the representation of the data (e.g., by nonlinearly pre-processing the audio signal), then the MAP estimate changes, unlike the KL-divergence in Equation 6, which is invariant to the coordinate system.

4.2. Additional loss terms

Training with *Probability Density Distillation* alone might not sufficiently constrain the student to generate high quality audio streams. Therefore, we also introduce additional loss functions to guide the student distribution towards the desired output space.

Power loss

The first additional loss we propose is the *power loss*, which ensures that the power in different frequency bands of the speech are on average used as much as in human speech. The power loss helps to avoid the student from collapsing to a high-entropy WaveNet-mode, such as whispering.

The power-loss is defined as:

$$||\phi(g(\boldsymbol{z}, \boldsymbol{c})) - \phi(\boldsymbol{y})||^2, \tag{16}$$

where $(\boldsymbol{y}, \boldsymbol{c})$ is an example with conditioning from the training set, $\phi(\boldsymbol{x}) = |\text{STFT}(\boldsymbol{x})|^2$ and STFT stands for the Short-Term Fourier Transform. We found that $\phi(\boldsymbol{x})$ can be averaged over time before taking the Euclidean distance with little difference in effect, which means it is the *average* power for various frequencies that is important.

Perceptual loss

In the power loss formulation given in equation 16, one can also use a neural network instead of the STFT to conserve a perceptual property of the signal rather than total energy. In our case we have used a WaveNet-like classifier trained to predict the phones from raw audio. Because such a classifier naturally extracts high-level features that are relevant for

Method	**Subjective 5-scale MOS**
16kHz, 8-bit μ-law, 25h data	
LSTM-RNN parametric[1]	3.67 ± 0.098
HMM-driven concatenative[1]	3.86 ± 0.137
WaveNet[1]	4.21 ± 0.081
24kHz, 16-bit lin. PCM, 65h data	
HMM-driven concatenative	4.19 ± 0.097
Autoregressive WaveNet	4.41 ± 0.069
Distilled WaveNet	4.41 ± 0.078

Table 1. Comparison of WaveNet distillation with the autoregressive teacher WaveNet, unit-selection (concatenative), and previous results ([1]) from (van den Oord et al., 2016a). MOS stands for Mean Opinion Score.

recognising the phones, this loss term penalises bad pronunciations. A similar principle has been used in computer vision for artistic style transfer (Gatys et al., 2015), or to get better perceptual reconstruction losses, e.g., in super-resolution (Johnson et al., 2016).

We have experimented with two different ways of using the perceptual loss, the feature reconstruction loss (the Euclidean distance between feature maps in the classifier) and the style loss (the Euclidean distance between the Gram matrices (Johnson et al., 2016)). The latter produced better results in our experiments.

Contrastive loss

Finally, we also introduce a contrastive distillation loss as follows:

$$D_{\mathrm{KL}}\Big(P_S(\boldsymbol{c}_1)\Big|\Big|P_T(\boldsymbol{c}_1)\Big) - \gamma D_{\mathrm{KL}}\Big(P_S(\boldsymbol{c}_1)\Big|\Big|P_T\boldsymbol{c}_2)\Big), \tag{17}$$

which minimises the KL-divergence between the teacher and student when both are conditioned on the same information $\boldsymbol{c}_1$ (e.g., linguistic features, speaker ID, ...), but also maximises it for different conditioning pairs $\boldsymbol{c}1 \neq \boldsymbol{c}2$. In order to implement this loss, we use the output of the student $\boldsymbol{x} = g(\boldsymbol{z}, \boldsymbol{c}_1)$ and evaluate the waveform twice under the teacher: once with the same conditioning $P_T(\boldsymbol{x}|\boldsymbol{c}_1)$ and once with a randomly sampled conditioning input: $P_T(\boldsymbol{x}|\boldsymbol{c}_2)$. The weight for the contrastive term γ was set to 0.3 in our experiments. The contrastive loss penalises waveforms that have high likelihood regardless of the conditioning vector.

5. Experiments

In all our experiments we used text-to-speech models that were conditioned on linguistic features (similar to (van den Oord et al., 2016a)), providing phonetic and duration information to the network. We also conditioned the models on pitch information (logarithm of f_0, the fundamental frequency) predicted by a different model. We never used ground-truth information (such as pitch or duration) extracted from human speech for generating audio samples and the test sentences were not present (or similar to those) in the training set.

The teacher WaveNet network was trained for 1,000,000 steps with the ADAM optimiser (Kingma & Ba, 2014) with a minibatch size of 32 audio clips, each containing 7,680 timesteps (roughly 320ms). Remarkably, a relatively short snippet of time is sufficient to train the parallel WaveNet to produce long term coherent waveforms. The learning rate was held constant at 2×10^{-4}, and Polyak averaging (Polyak & Juditsky, 1992) was applied over the parameters. The model consists of 30 layers, grouped into 3 dilated residual block stacks of 10 layers. In every stack, the dilation rate increases by a factor of 2 in every layer, starting with rate 1 (no dilation) and reaching the maximum dilation of 512 in the last layer. The filter size of causal dilated convolutions is 3. The number of hidden units in the gating layers is 512 (split into two groups of 256 for the two parts of the activation function (1)). The number of hidden units in the residual connection is 512, and in the skip connection and the 1×1 convolutions before the output layer is also 256. We used 10 mixture components for the mixture of logistics output distribution.

The student network consisted of the same WaveNet architecture layout, except with different inputs and outputs and no skip connections. The student was also trained for 1,000,000 steps with the same optimisation settings. The student typically consisted of 4 flows with 10, 10, 10, 30 layers respectively, with 64 hidden units for the residual and gating layers.

Audio Generation Speed

We have benchmarked the sampling speed of autoregressive and distilled WaveNets on an NVIDIA P100 GPU. Both models were implemented in Tensorflow (Abadi et al., 2016) and compiled with XLA. The hidden layer activations from previous timesteps in the autoregressive model were cached with circular buffers (Paine et al., 2016). The resulting sampling speed with this implementation is **172 timesteps/second** for a minibatch of size 1. The distilled model, which is more parallelizable, achieves **over 500,000 timesteps/second** with same batch size of 1, resulting in three orders of magnitude speed-up.

Audio Fidelity

In our first set of experiments, we looked at the quality of WaveNet distillation compared to the autoregressive WaveNet teacher and other baselines on data from a professional female speaker (van den Oord et al., 2016a). Table

	Parametric	Concatenative	Distilled WaveNet
English speaker 1 (female - 65h data)	3.88	4.19	4.41
English speaker 2 (male - 21h data)	3.96	4.09	4.34
English speaker 3 (male - 10h data)	3.77	3.65	4.47
English speaker 4 (female - 9h data)	3.42	3.40	3.97
Japanese speaker (female - 28h data)	4.07	3.47	4.23

Table 2. Comparison of MOS scores on English and Japanese with multi-speaker distilled WaveNets. Note that some speakers sounded less appealing to people and always get lower MOS, however distilled parallel WaveNet always achieved significantly better results.

1 gives a comparison of autoregressive WaveNet, distilled WaveNet and current production systems in terms of mean opinion score (MOS). There is no difference between MOS scores of the distilled WaveNet (4.41 ± 0.08) and autoregressive WaveNet (4.41 ± 0.07), and both are significantly better than the concatenative unit-selection baseline (4.19 ± 0.1).

It is also important to note that the difference in MOS scores of our WaveNet baseline result 4.41 compared to the previous reported result 4.21 (van den Oord et al., 2016a) is due to the improvement in audio fidelity as explained in Section 2.1: modelling a sample rate of 24kHz instead of 16kHz and bit-depth of 16-bit PCM instead of 8-bit μ-law.

Multi-speaker Generation

By conditioning on the speaker-ids we can construct a single parallel WaveNet model that is able to generate multiple speakers' voices and their accents. These networks require slightly more capacity than single speaker models and thus had 30 layers in each flow. In Table 2 we show a comparison of such a distilled parallel WaveNet model with two main baselines: a parametric and a concatenative system. In the comparison, we use a number of English speakers from a single model (one of them, English speaker 1, is the same speaker as in Table 1) and a Japanese speaker from another model. For some speakers, the concatenative system gets better results than the parametric system, while for other speakers it is the opposite. The parallel WaveNet model, on the other hand, significantly outperforms both baselines for all the speakers.

Ablation Studies

To analyse the importance of the loss functions introduced in Section 4.2 we show how the quality of the distilled WaveNet changes with different loss functions in Table 3 (top). We found that MOS scores of these models tend to be very similar to each other (and similar to the result in Table 1). Therefore, we report subjective preference scores from a paired comparison test ("A/B test"), which we found to be more reliable for noticing small (sometimes qualitative) differences. In these tests, the subjects were asked to listen to a pair of samples and choose which they preferred, though they could choose "neutral" if they did not have any preference.

Losses used	Win-Lose-Neutral
KL + Power	60% - 15% - 25%
KL + Power + Percept	66% - 10% - 24%
KL + Power + Percept + Contrast *	65% - 9% - 26%

Table 3. Performance with respect to different combinations of loss terms. We report preference comparison scores since their mean opinion scores tend to be very close and inconclusive. Last row (combination of KL + Power + Perceptual + Contrastive losses)is the default model used.

As mentioned before, the KL loss alone does not constrain the distillation process enough to obtain natural sounding speech (e.g., low-volume audio suffices for the KL), therefore we do not report preference scores with only this term. The KL loss (section 4) combined with power-loss is enough to generate quite natural speech. Adding the perceptual loss gives a small but noticeable improvement. Adding the contrastive loss does not improve the preference scores any further, but makes the generated speech less noisy, which is something most raters do not pay attention to, but is important for production quality speech.

As explained in Section 3, we use multiple inverse-autoregressive flows in the parallel WaveNet architecture: A model with a single flow gets a MOS score of 4.21, compared to a MOS score of 4.41 for models with multiple flows.

6. Conclusion

In this paper we have introduced a novel method for high-fidelity speech synthesis based on WaveNet (van den Oord et al., 2016a) using *Probability Density Distillation*. The proposed model achieved several orders of magnitude speed-up compared to the original WaveNet with no significant difference in quality. Moreover, we have successfully transferred this algorithm to new languages and multiple speakers.

As a result, we have been able to run a real-time speech synthesis system, opening the door to many exciting future developments thanks to the flexibility of deep learning models. We believe that the same method presented here can be used in many different domains to achieve similar speed improvements whilst maintaining output accuracy.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 881–889, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711. Springer, 2016.

Kalchbrenner, N., van den Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 29–37, 2011.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.

Paine, T. L., Khorrami, P., Chang, S., Zhang, Y., Ramachandran, P., Hasegawa-Johnson, M. A., and Huang, T. S. Fast wavenet generation algorithm. *CoRR*, abs/1611.09482, 2016. URL http://arxiv.org/abs/1611.09482.

Papamakarios, G., Murray, I., and Pavlakou, T. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2335–2344, 2017.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Theis, L. and Bethge, M. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems*, pp. 1927–1935, 2015.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016b.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016c.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.