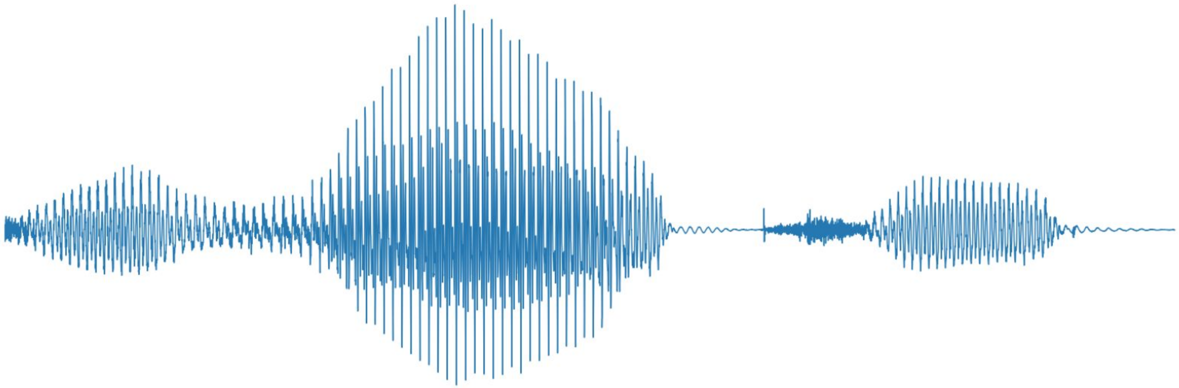
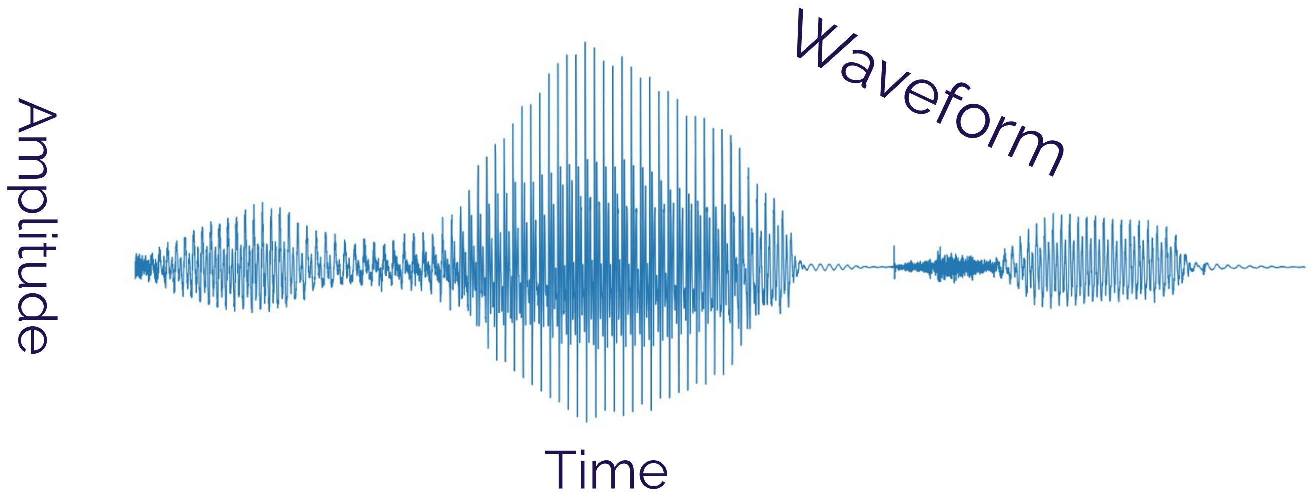
DEEP GENERATIVE MODELS FOR RAW AUDIO SYNTHESIS



DEEP GENERATIVE MODELS FOR RAW AUDIO SYNTHESIS



8 years in math, physics and computer science

For the last 4 years mostly work with machine learning

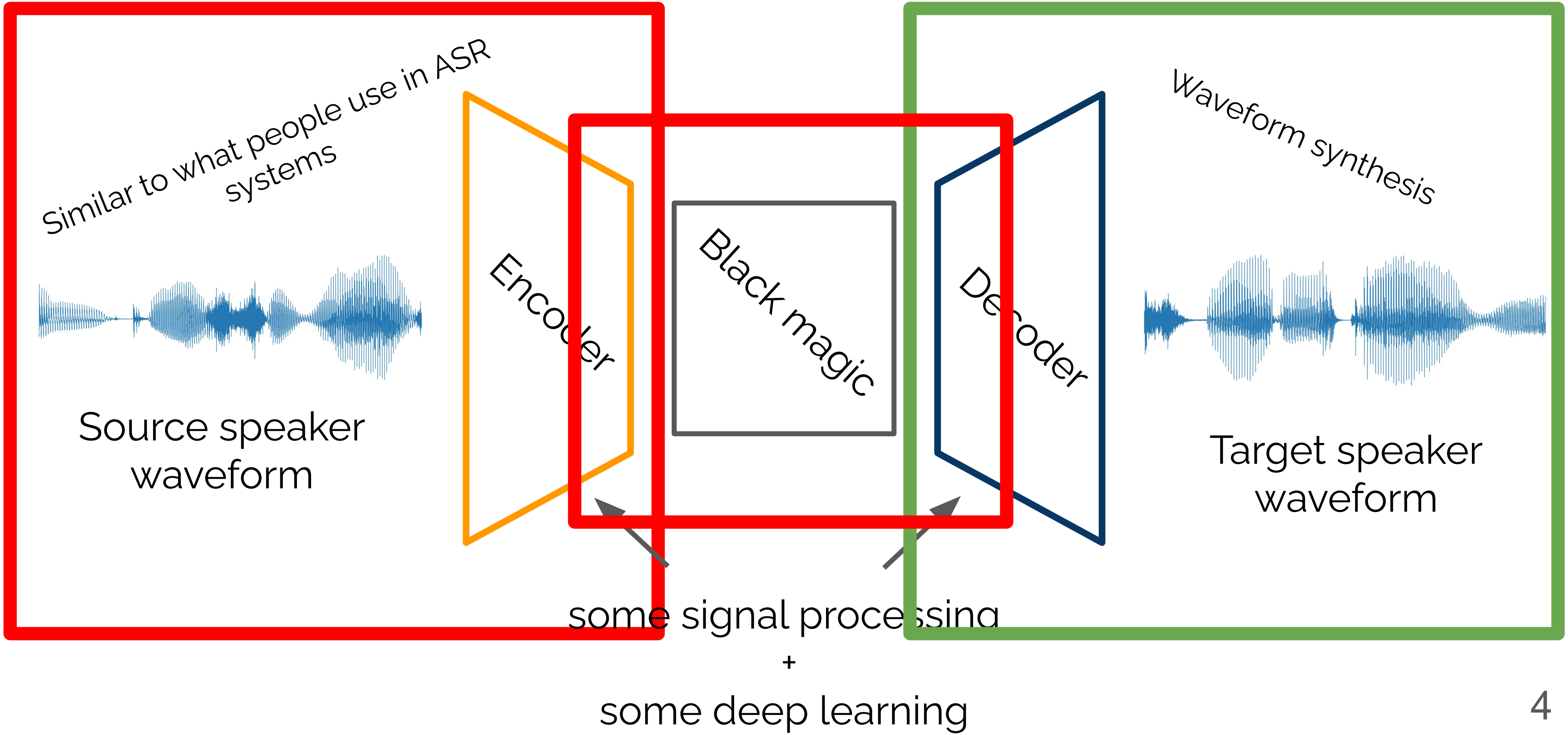
Currently do voice conversion in

respeecher

for applications in movie industry, audiobooks and games

Occasional Kaggler: currently top 1 in Ukraine and top 70 worldwide

VOICE CONVERSION IN A NUTSHELL



Text-to-speech

Hello AIUkraine!

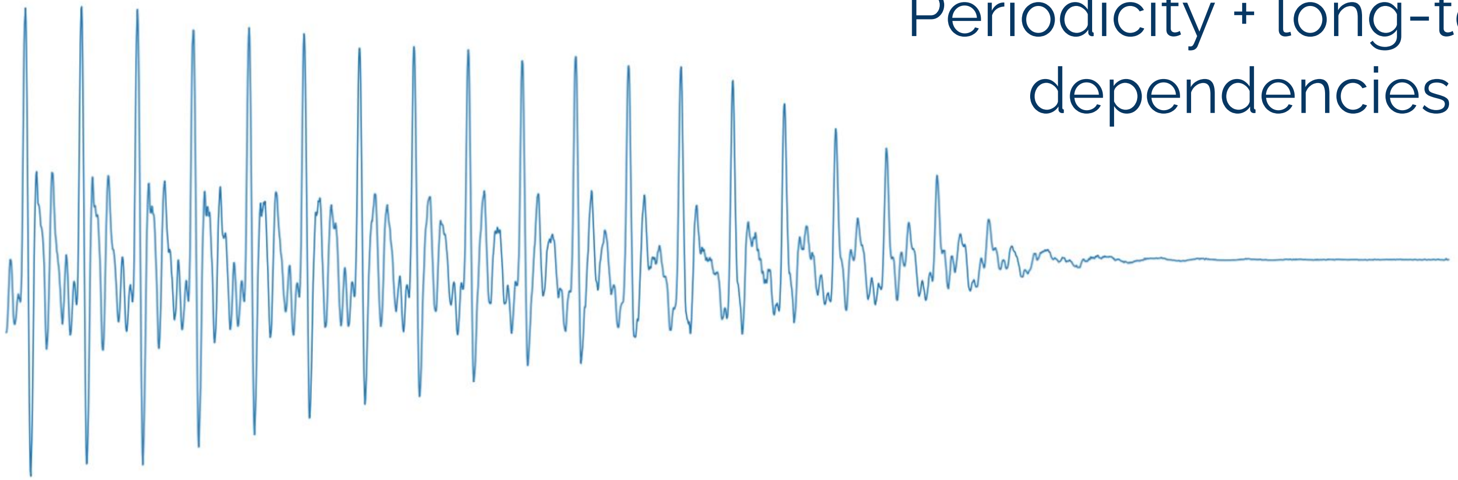
Very high dimensionality

Typical sample rate ranges from **16000** to **44000** samples per second

One second of 16 kHz speech

Samples are strongly correlated

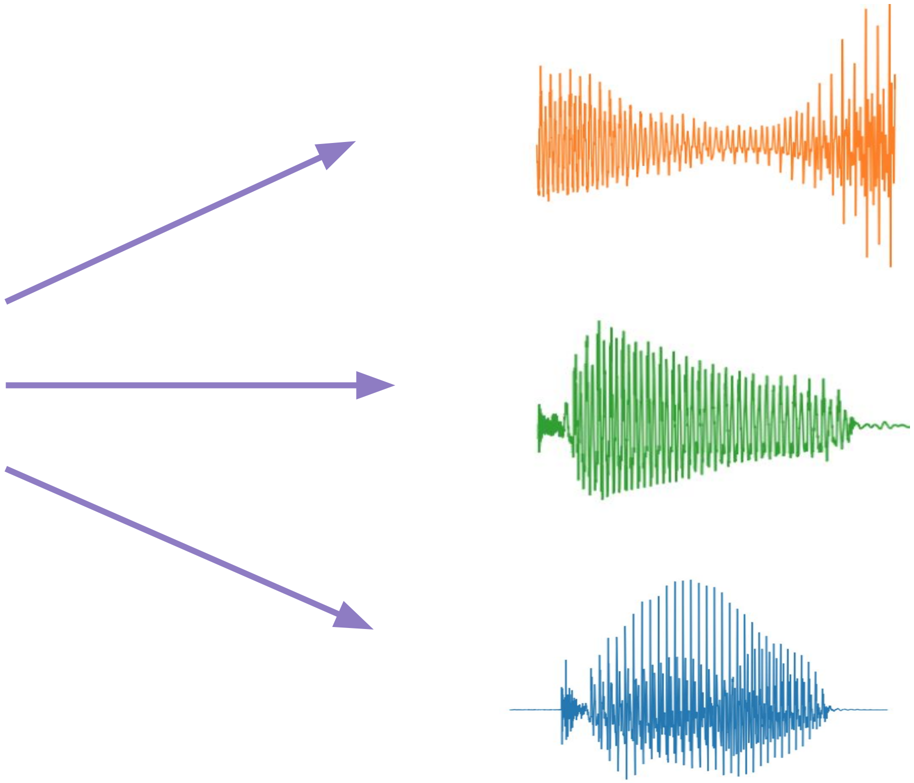
We need to **jointly** model **thousands** of random variables



Periodicity + long-term dependencies

The is no single answer

The same text

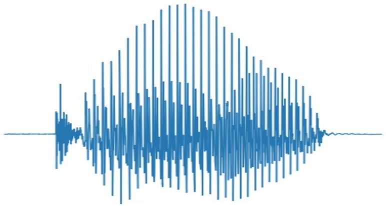


Issues with conventional methods

- Hard to control **prosody** (emotional content)
- Require a lot of **labeled** data
- **Inexpressive** models (such as HMM)
- Rely heavily on **domain knowledge**
- Hard to get **natural** sounding

Idea:

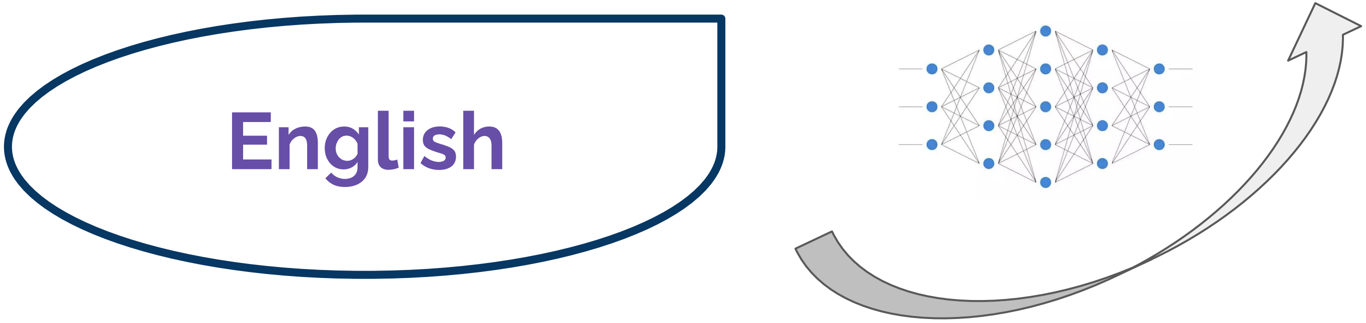
Reformulate the task as a **joint** probability function (or density) estimation:

$p($  $\mid$ text $)$

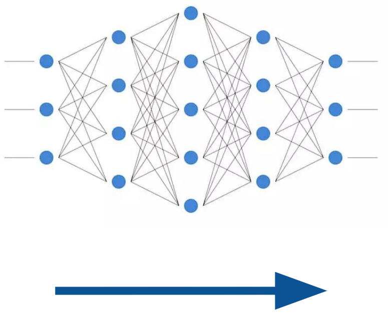
Which waveforms are likely to correspond to a given text?

Analogy to machine translation

- Multiple outcomes
- Joint distribution of words (language model)

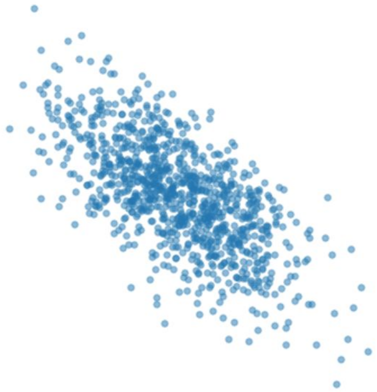


Text



Parameter estimation is typically performed via **maximum likelihood estimation**

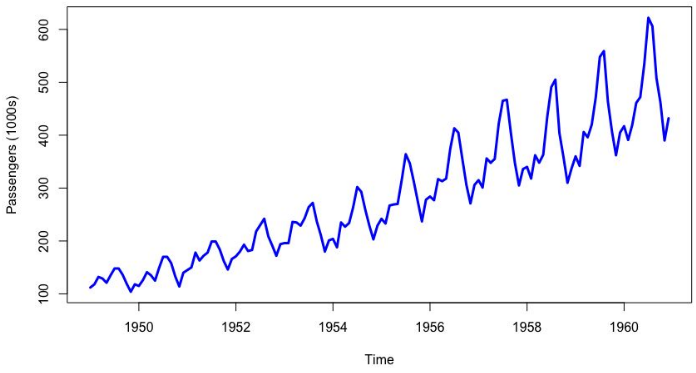
Recap: the maximum likelihood

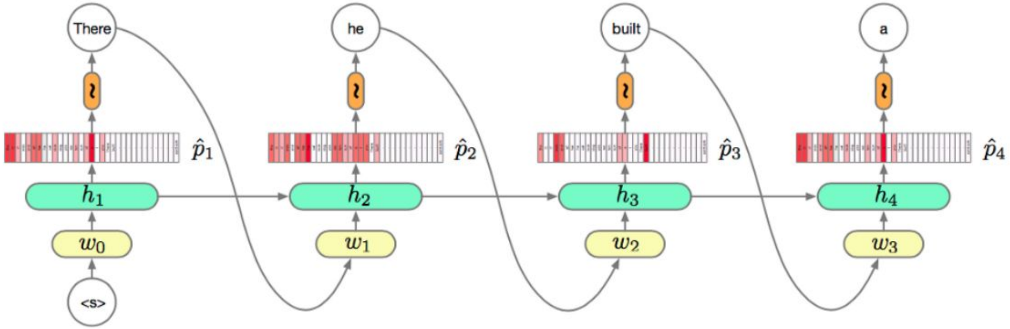


$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$$

Maximize the probability of observing the data

Autoregressive models



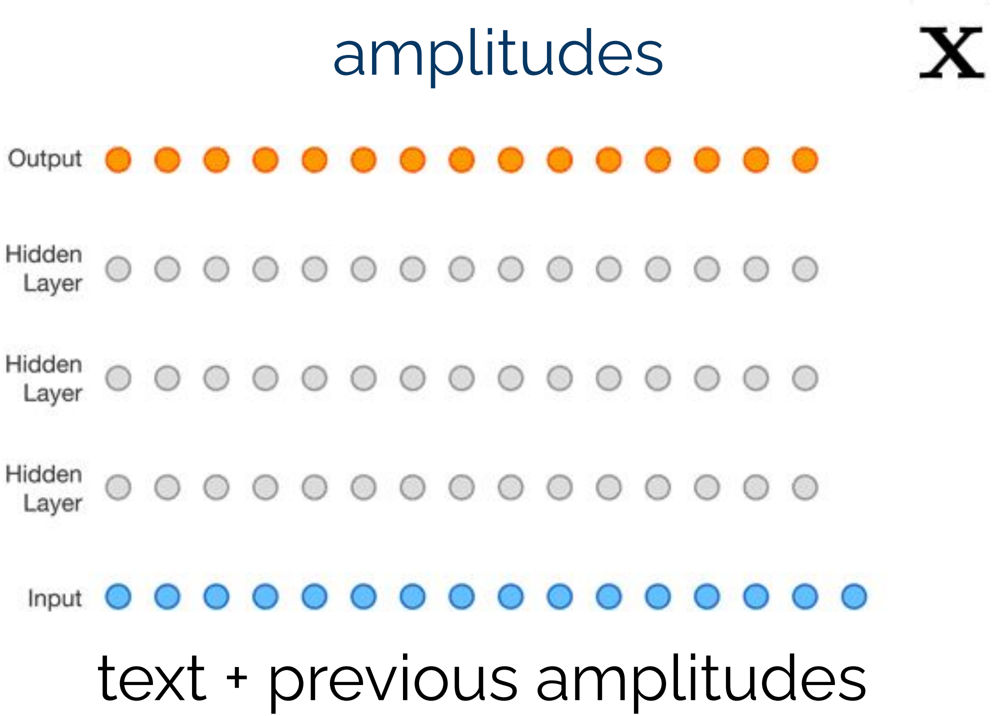


Time series forecasting (**AR**IMA, S**AR**IMA, F**AR**IMA)

Language models (typically with recurrent neural networks)

Basic idea: the next value can be represented as **a function of the previous values**

WaveNet



Waveform is modeled by a stack of dilated causal convolutions

Source: DeepMind blog

https://arxiv.org/abs/1609.03499

WaveNet

Training: maximize the probability estimated by the model according to the maximum likelihood principle. **Can be done in parallel for all time steps:**

$$p(x_t) = f(x_{t-1}, x_{t-2}, x_{t-3}, \ldots, x_1)$$

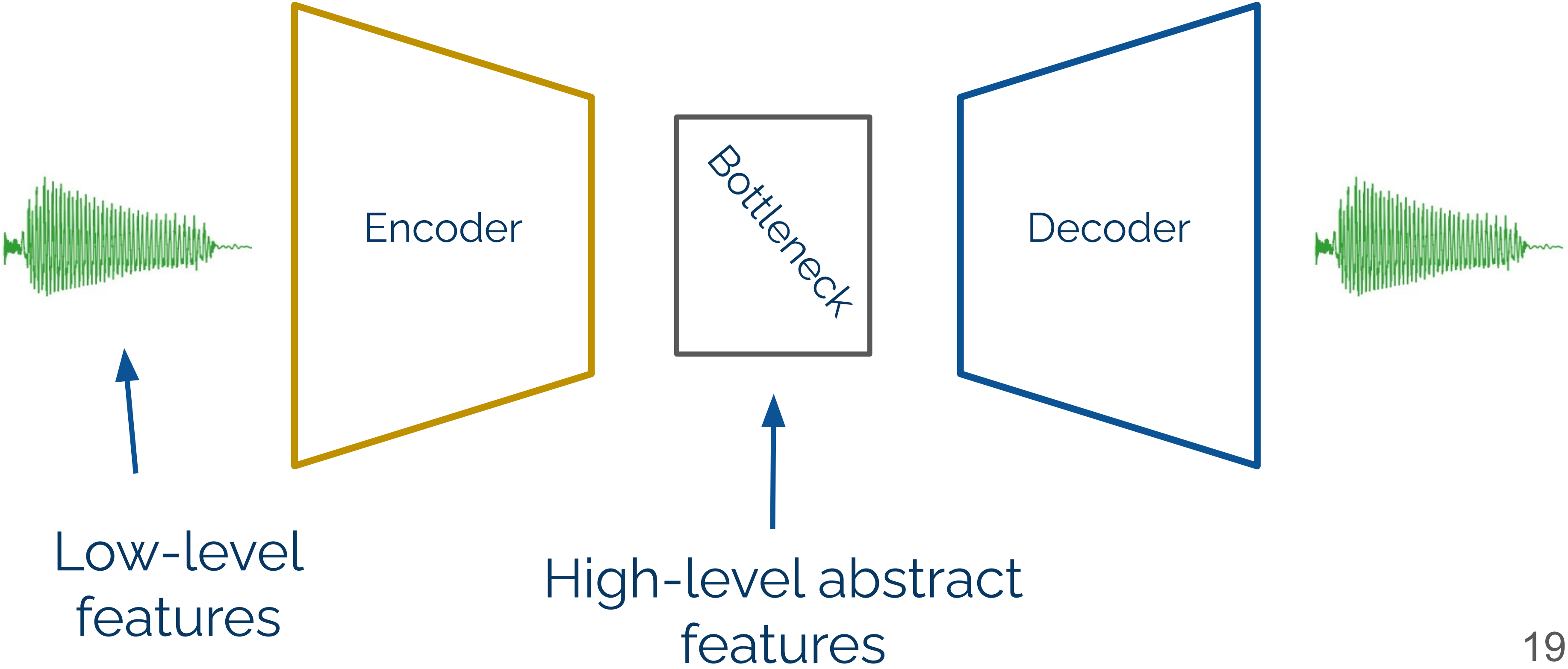
Generation: sequentially generate samples **one by one**, sampling from a predicted distribution on every time step

Data scientists when their model is training

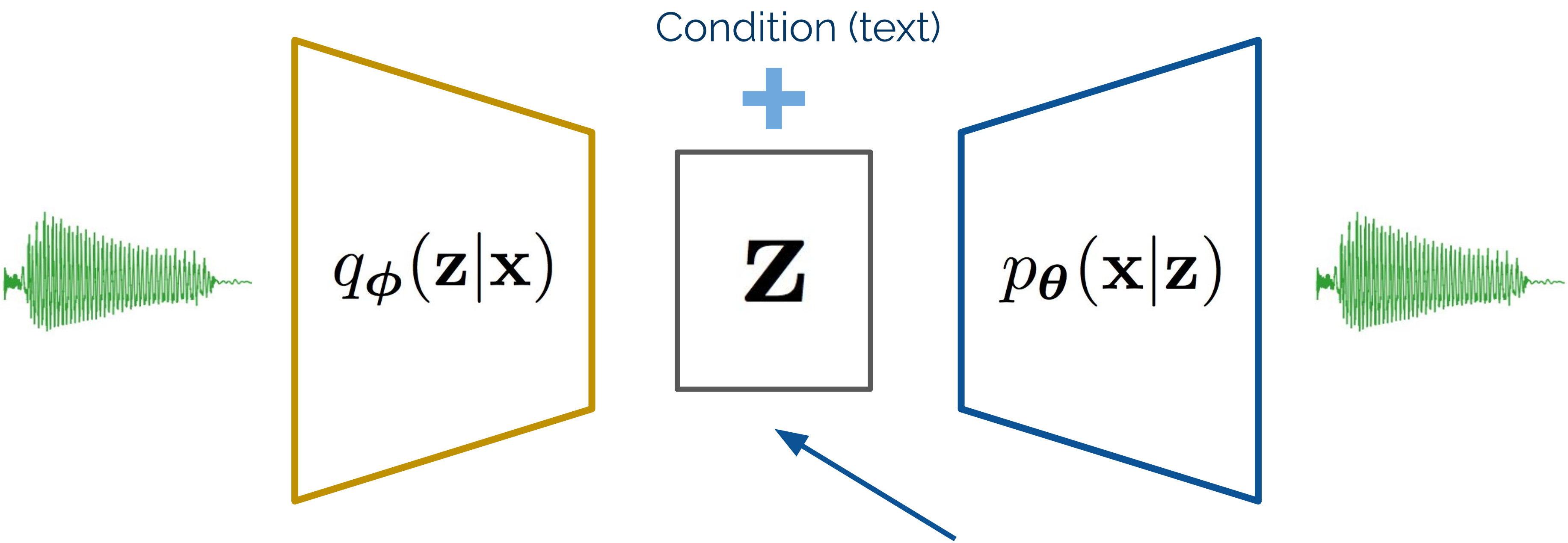
Deep learning engineers when their WaveNet is generating

Autoencoders

Goal: **reconstruct** the input



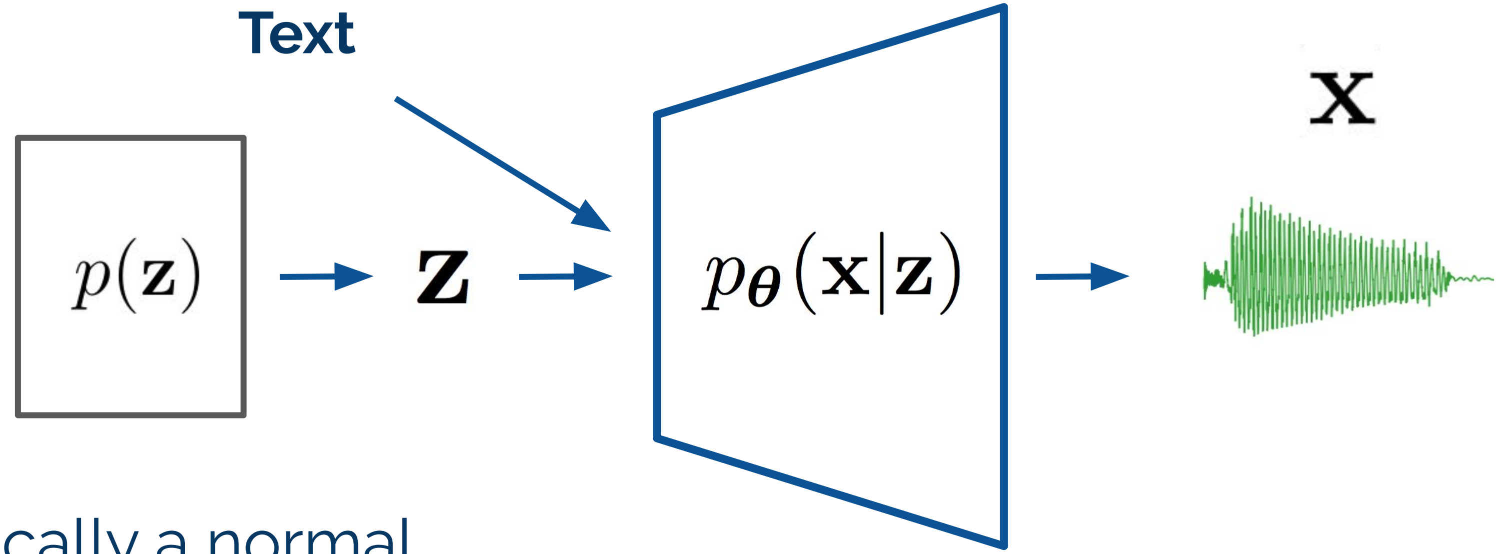
Variational autoencoder



Learns an approximation to $p_{\theta}(\mathbf{x})$

Latent space

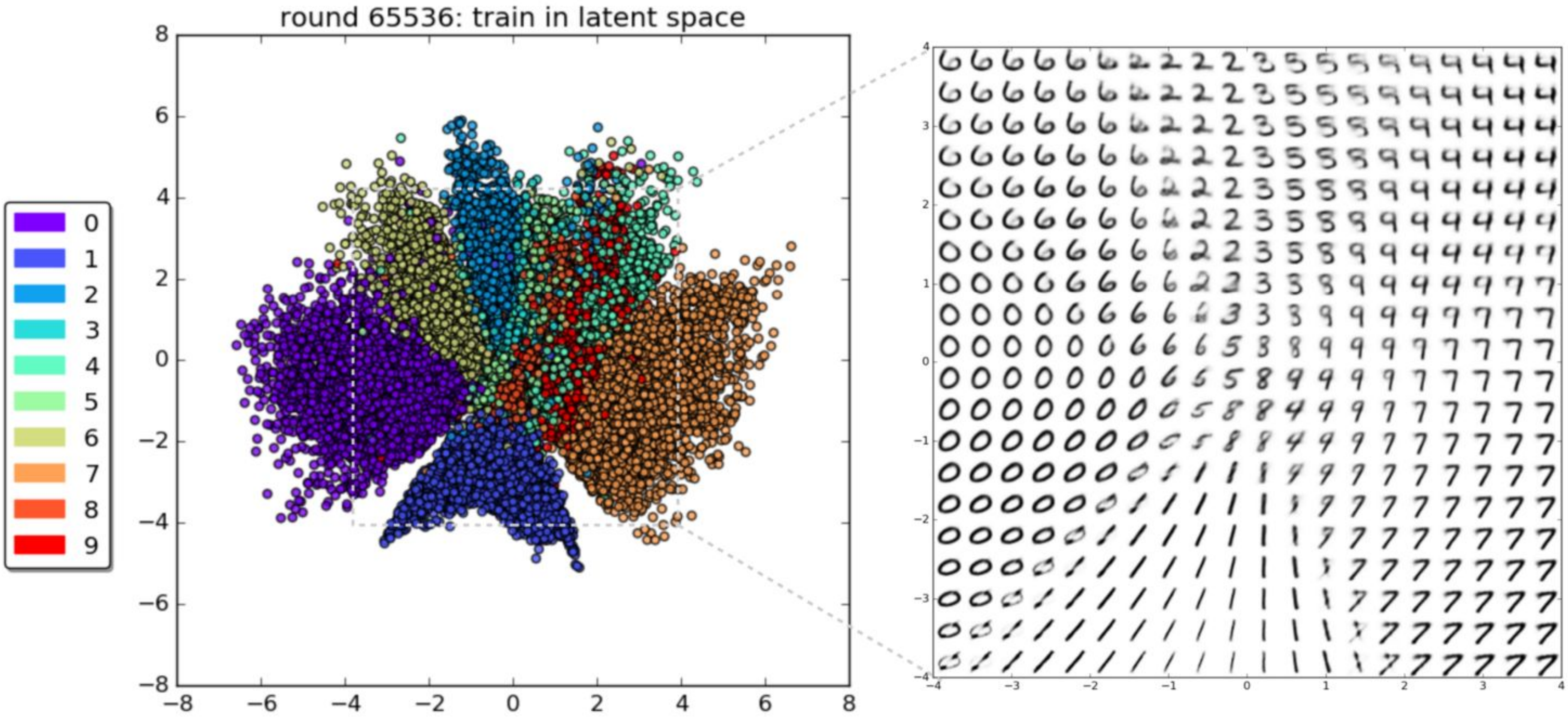
Variational autoencoder: sampling



Typically a normal distribution

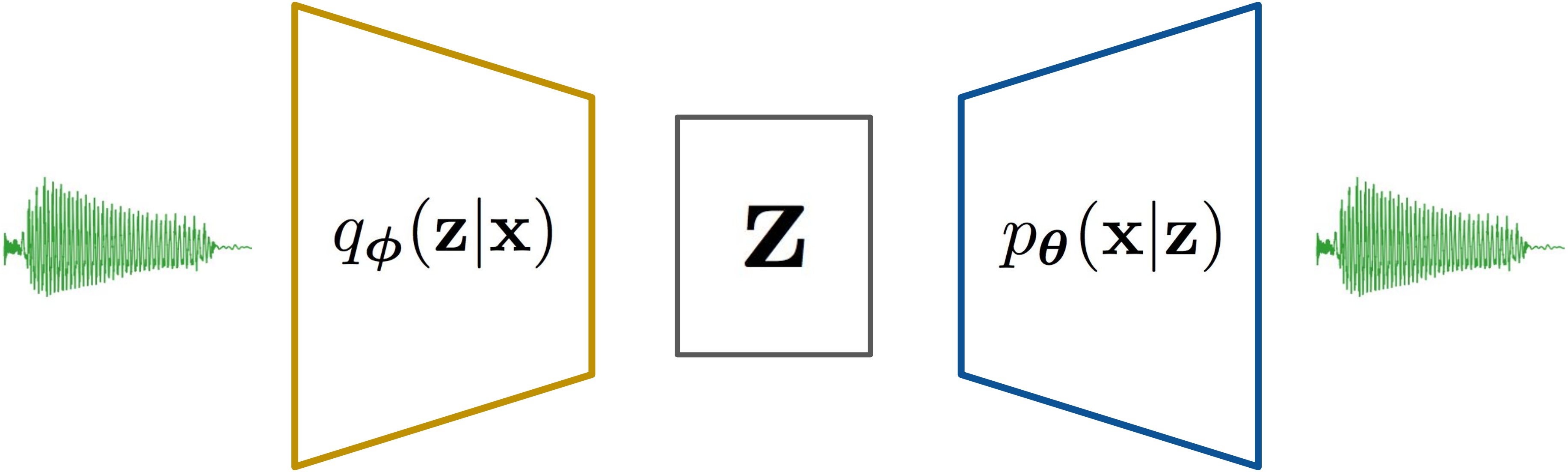
By tweaking the latent variables, we can control **prosody**, **tempo**, **accent** and much more

Variational autoencoder: latent space



Source: https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html

Upgrade: VQ-VAE

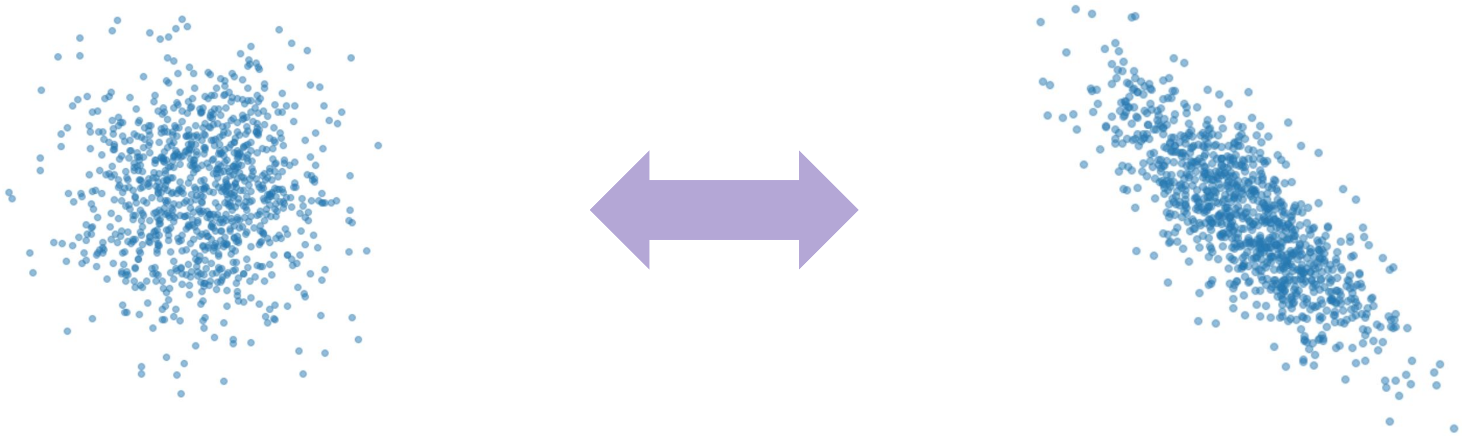


Now the latent space is **discrete** and represented by an **autoregressive** model

https://arxiv.org/abs/1711.00937

Normalizing flows

Take a random variable $\mathbf{z}$ with distribution $q(\mathbf{z})$, apply some **invertible** mapping: $\mathbf{z}' = f(\mathbf{z})$



Normalizing flows

Take a random variable $\mathbf{z}$ with distribution $q(\mathbf{z})$, apply some **invertible** mapping: $\mathbf{z}' = f(\mathbf{z})$

Recall the **change of variables** rule:

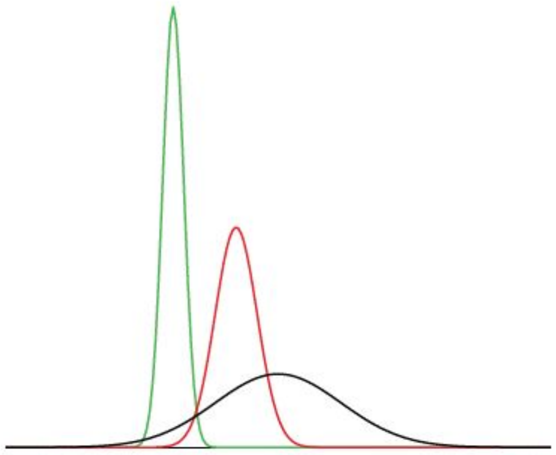
$$q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{z}'} \right| = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}$$

The change of variables rule

$$z \sim N(0,1) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} = f(z),$$

$$y = z\sigma + \mu \qquad \text{so that} \qquad z = (y-\mu)/\sigma$$

$$\begin{aligned} g(y) = f\{z(y)\}\left|\frac{dz}{dy}\right| &= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right\}\frac{1}{\sigma} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}\right\}. \end{aligned}$$



For multidimensional random variables, replace the derivative with the **Jacobian (a matrix of derivatives)**

General case (multiple transforms)

a flow

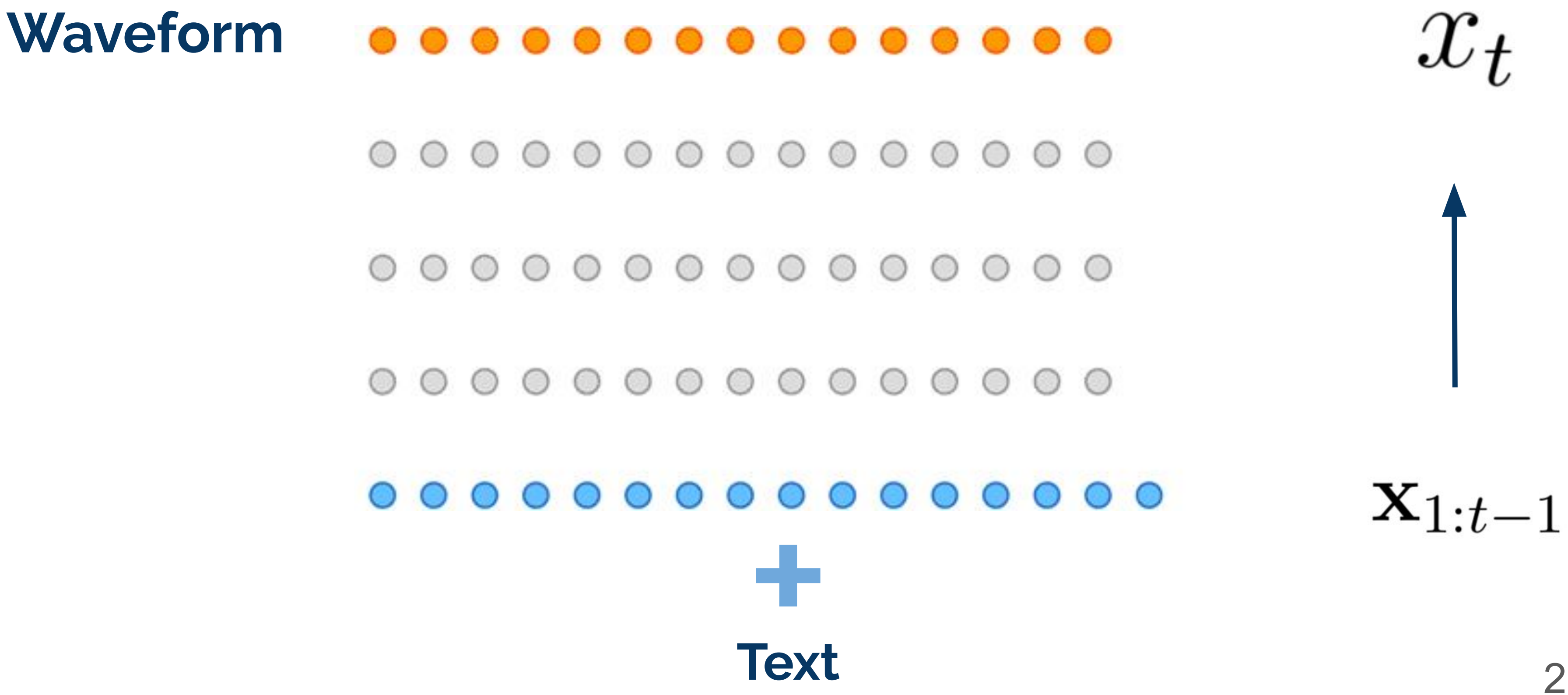
$$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}_0)$$

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

$p_\theta(\mathbf{x})$

Can be optimized directly, e.g. with a stochastic gradient ascent

$$p(x_t) = f(x_{t-1}, x_{t-2}, x_{t-3}, \ldots, x_1)$$



Key idea: represent WaveNet with a normalizing flow

This approach is called

Inverse Autoregressive Flow

$$p(x_t) = f(z_{t-1}, z_{t-2}, z_{t-3}, \dots, z_1)$$

Waveform

$$\mathbf{x}$$

$\uparrow$

White noise

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

\+

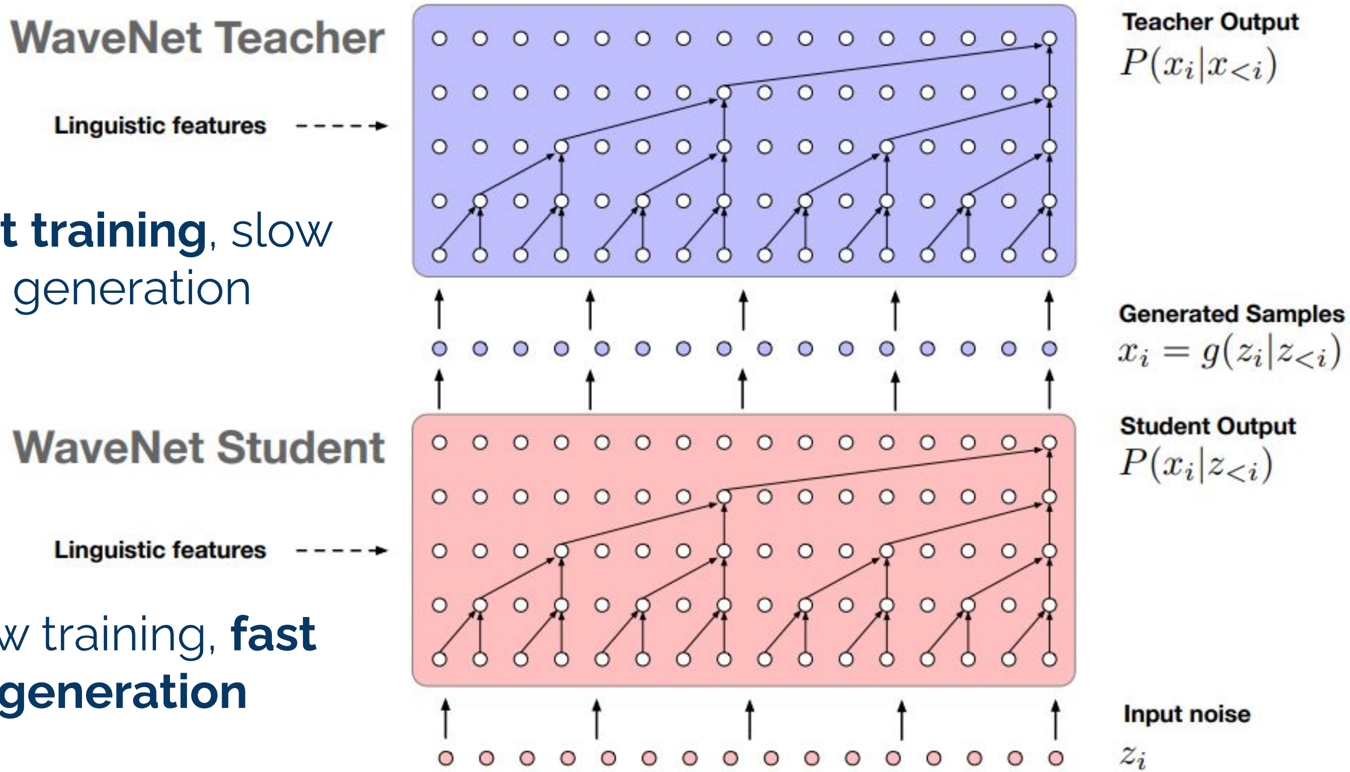
Text

https://deepmind.com/blog/article/high-fidelity-speech-synthesis-wavenet

Parallel WaveNet: the voice of Google Assistant

fast training, slow generation

slow training, **fast generation**



https://arxiv.org/abs/1711.10433

https://arxiv.org/abs/1609.03499 - WaveNet
https://arxiv.org/abs/1312.6114 - Variational Autoencoder
https://arxiv.org/abs/1711.00937 - VQ-VAE
https://arxiv.org/abs/1711.10433 - Parallel WaveNet
https://deepmind.com/blog/article/wavenet-generative-model-raw-audio - DeepMind's blogpost on WaveNet
https://deepmind.com/blog/article/high-fidelity-speech-synthesis-wavenet - DeepMind's blogbost on Parallel Wavenet
https://avdnoord.github.io/homepage/vqvae/ - VQ-VAE explanation from the author
https://deepgenerativemodels.github.io/notes/autoregressive/ - a good tutorial on deep autoregressive models
https://blog.evjang.com/2018/01/nf1.html - a nice intro to normalizing flows
https://medium.com/@kion.kim/wavenet-a-network-good-to-know-7caaae735435 - introductory blogpost on WaveNet
http://anotherdatum.com/vae.html - a good explanation of principles and math behind VAE

Q&A

dmitry-danevskiy

ddanevskyi