# TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2023

## Adjusting Generation

## Temperature and Guidance

## Temperature-Adjusted Generation

$$\text{Training:} \quad \Phi^* = \operatorname*{argmin}_{\Phi} E_{(x,y)\sim\mathrm{Pop}}[-\ln P_\Phi(y|x)]$$

$$P_\Phi(y|x) = \operatorname*{softmax}_{y} e^{s_\Phi(y|x)}$$

$$\text{Generation:} \quad P_\Phi^\beta(y|x) = \operatorname*{softmax}_{y} e^{\beta s_\Phi(y|x)} \propto P_\Phi(y)^\beta$$

In language translation we take $\beta = \infty$ (softmax $\Rightarrow$ argmax).

In language generation from an LLM we take $\beta > 1$.

## Temperature Adjusted Generation for Language

In practice we use

$$
\begin{aligned}
P_\Phi^\beta(y_{i+1} \mid y_1, \ldots, y_i) &= \operatorname*{softmax}_{y_{i+1}} \beta s_\Phi(y_{i+1} \mid y_1, \ldots, y_i) \\
&\propto P_\Phi(y_{i+1} \mid y_1, \ldots, y_i)^\beta
\end{aligned}
$$

This is different from

$$
P_\Phi^\beta(y_1, \ldots, y_N) \propto P_\Phi(y_1, \ldots, y_N)^\beta
$$

# Temperature-Adjusted Generation for Language

For language generation $\beta = 1$ tends to yield rambling and incoherent text.

On the other hand $\beta = \infty$ generates repetition.

We look for a Goldilocks $\beta$.

An alternative to temperature-adjusted generation is top-P sampling, also called nucleus sampling, which is similar in structure and performance.

There is a literature on generation adjustment for language.

# Temperature-Adjusted Reverse-Diffusion

$$z(t - \Delta t) = z(t) + \left(\frac{\hat{E}_\Phi[y|t, z(t)] - z(t)}{t}\right) \Delta t + \epsilon \sqrt{\frac{\Delta t}{\beta}}$$

$$t' = t/\beta$$

$$z(t' - \Delta t') = z(t') + \beta \left(\frac{\hat{E}_\Phi[y|t', z(t')] - z(t')}{t'}\right) \Delta t' + \epsilon \sqrt{\Delta t'}$$

As with language generation, this is not the same as $P_\Phi^\beta(y) \propto P_\Phi(y)^\beta$

## Classifier-Guidance

Diffusion Models Beat GANs on Image Synthesis
Dharwali and Nichol, May 2021

For imagenet class-conditional image generation $P_\Psi(y|x)$ they utilize an imagenet classification model $P_\Psi(x|y)$.

They train a diffusion model for unconditional imagenet generation $P_\Phi(y)$.

They note that

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)} \propto P(y)P(x|y)$$

## Classifier-Guidance

For generation they modify the reverse-diffusion process so as to intuitively approximate

$$P^{\gamma}_{\Phi,\Psi}(y|x) = \operatorname*{softmax}_{y} s_{\Phi}(y) + \gamma s_{\Psi}(x|y)$$

$\gamma$ is called the strength of the guidance.

$$z(t - \Delta t) = z(t) + \beta \left( \frac{\hat{E}_{\Phi}[y|t, z(t)] + -z(t)}{t} + \gamma s_{\Psi}(x|y) \right) \Delta t + \epsilon \sqrt{\Delta t}$$

I have included $\beta$ as a parameter because the relative size of the linear drift and noise is a natural parameter of reverse-diffusion.

# Classifier-Guidance

$$z(t-\Delta t) = z(t) + \beta\left(\frac{\hat{E}_\Phi[y|t, z(t)] - z(t)}{t} + \gamma s_\Psi(x|z(t))\right)\Delta t + \epsilon\sqrt{\Delta t}$$

Note that this uses an **unconditional** model $P_\Phi(y)$ implicitly defined by $\hat{E}_\Phi[y|t, z(t)]$.

This is different from, but motivated by,

$$P^{\beta,\gamma}_{\Phi,\Psi}(y|x) \propto P_\Phi(y)^\beta P_\Psi(x|y)^{\beta+\gamma}$$

# Conditional Diffusion Models

$$P_\Phi(y \mid \text{panda bear chemist})$$

panda mad scientist mixing sparkling chemicals, artstation

$$\text{Train } \hat{E}_\Phi[y|t, z(t), x]$$

# Classifier-Free Guidance (Self-Guidance)

Classifier-Free Diffusion Guidance
Ho and Salimans, December 2021 (NeurIPS workshop)

$$\text{Training:} \quad \Phi^* = \operatorname*{argmin}_{\Phi} E_{(x,y)\sim \mathrm{Pop}}[-\ln P_\Phi(y|x)]$$

$$P_\Phi(y|x) = \operatorname*{softmax}_{y} e^{s_\Phi(y|x)}$$

We introduce a special $x$-value $\emptyset$ and arrange that

$$\mathrm{Pop}(y|\emptyset) = \mathrm{Pop}(y).$$

# Guidance

They modify the reverse-diffusion process to intuitively approximate

$$P_\Phi^\beta(y|x) = \operatorname*{softmax}_y e^{\beta(s_\Phi(y|x) - (1-1/\beta)s_\Phi(y|\emptyset))}, \quad \beta \geq 1$$

For $\beta = 1$ we have no adjustment.

$$P_\Phi^1(y|x) = \operatorname*{softmax}_y e^{s_\Phi(y|x)}$$

For $\beta >> 1$ (used in practice) we have.

$$P_\Phi^\beta(y|x) \approx \operatorname*{softmax}_y e^{\beta(s_\Phi(y|x) - s_\Phi(y|\emptyset))}$$

# Guidance

$$P_\Phi^\beta(y|x) = \operatorname*{softmax}_y e^{\beta(s_\Phi(y|x) - s_\Phi(y|\emptyset))} \quad \propto \quad \left(\frac{P_\Phi(y|x)}{P_\Phi(y|\emptyset)}\right)^\beta$$

$$z(t-\Delta t) = z(t) + \left(\frac{\beta(\hat{E}_\Phi[y|t, z(t), x] - \hat{E}_\Phi[y|t, z(t), \emptyset]) - z_t}{t}\right) \Delta t + \epsilon\sqrt{\Delta t}$$

# Guidance

$$P_\Phi^\beta(y|x) \quad \propto \quad \left(\frac{P_\Phi(y|x)}{P_\Phi(y|\emptyset)}\right)^\beta$$

Ho and Salimans motivate this from Classifier Guidance and

$$P(x|y) \propto \frac{P(y|x)}{P(y)}$$

But this is false.

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)} \not\propto \frac{P(y|x)}{P(y)}$$

# Guidance

$$z(t-\Delta t) = z(t)+\beta\left(\frac{(\hat{E}_\Phi[y|t,z(t),x] - \hat{E}_\Phi[y|t,z(t),\textbf{blurry}]) - z_t}{t}\right)\Delta t+\epsilon\sqrt{\Delta t}$$

This will make the generated image sharper.

# A More General Formulation

Consider a Markovian VAE with deterministic encoder $z_{1,\mathrm{enc}}(y)$ and $z_{i+1,\mathrm{enc}}(z_i)$ and where $z_{N,\mathrm{enc}}(z_{N-1})$ is a constant $\emptyset$.

This holds for language models but also seems reasonable for a StyleGAN inverter (long story).

This is an enormous simplification (a good thing).

$$\mathrm{enc}^*, \mathrm{gen}^* = \operatorname*{argmin}_{\mathrm{enc},\mathrm{gen}} \; E_y[-\ln(P_{\mathrm{gen}}(y|z_1)P_{\mathrm{gen}}(z_1|z_2)\cdots P_{\mathrm{gen}}(z_{N-1}|\emptyset))]$$

# A More General Formulation

$$\text{enc}^*, \text{gen}^* = \underset{\text{enc,gen}}{\text{argmin}} \; E_y[-\ln(P_{\text{gen}}(y|z_1)P_{\text{gen}}(z_1|z_2)\cdots P_{\text{gen}}(z_{N-1}|\emptyset))]$$

In a language model we generate one word at a time.

But we can also consider the case where $z_i$ is a vector whose dimension is decreasing as $i$ increases.

In this case we can use

$$P_{\text{gen}}(z_{i-1}|z_i) = \hat{z}_{i-1}(z_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

## A More General Formulation

$$\mathrm{enc}^*, \mathrm{gen}^* = \underset{\mathrm{enc,gen}}{\mathrm{argmin}}\; E_y[-\ln(P_{\mathrm{gen}}(y|z_1)P_{\mathrm{gen}}(z_1|z_2)\cdots P_{\mathrm{gen}}(z_{N-1}|\emptyset))]$$

$$P_{\mathrm{gen}}(z_{i-1}|z_i) = \hat{z}_{i-1}(z_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

$$\mathrm{enc}^*, \mathrm{gen}^* = \underset{\mathrm{enc,gen}}{\mathrm{argmin}}\; E_y \, ||y - z_1||^2 + \sum_{i=1}^{N-1} ||z_i - \hat{z}_i(z_{i+1})||^2$$

## Conditional Generation

Training the encoder and the decoder conditioned on $x$ (as in a language translation model). This trains $\hat{z}_{i-1}(z_i, x)$.

For generation we then have

$$\begin{aligned}
\text{Unadjusted: } z_{i-1} &= \hat{z}_{i-1}(z_i, x) + \epsilon \\
\text{Temperature Adjusted: } z_{i-1} &= \hat{z}_{i-1}(z_i, x) + \frac{1}{\sqrt{\beta}}\,\epsilon \\
\text{Guidance Adjusted: } z_{i-1} &= \hat{z}_{i-1}(z_i, x_{\text{good}}) - \hat{z}_{i-1}(z_i, x_{\text{bad}}) + \frac{1}{\sqrt{\beta}}\,\epsilon
\end{aligned}$$

Output $z_1$

**END**