II Generative modeling with neural networks (in higher dimensions)

Relationship between generative modelling & compression

- compression $\quad X \to f(X) \to z \to g(z) \to \hat{X}$

  X: original data, f: encoder, z: compressed data, "codes", g: decoder, $\hat{X}$: reconstructed data

- lossless compression. $\hat{X} = X$, e.g. zip

  idea: - use short codes ($\|z\| \approx 0$) for frequent symbols
  longer codes ($\|z\| > 0$) for more rare symbols

- lossy compression. $\hat{X} \approx X$, e.g. jpeg

  idea: - decompose X into its important & unimportant components
  (jpeg: Fourier transform of 8×8 patches)

  - drop unimportant stuff ($\hat{=}$ set to 0, jpeg: wave length with low amplitude $\Rightarrow$ zero amplitude)
  - use lossless compression for important stuff (& the zeros)

- three conflicting goals:

(1) high compression $\quad d = \dim(z) \ll \dim(X) = D$

(2) accurate reconstruction $\quad \mathrm{dist}(x, \hat{x}) \approx 0$ for some distance fct.

(3) preserve data distribution: $\quad \mathrm{Dist}(p^*(x), p(\hat{x})) \approx 0$ for some Dist

(3) does not imply (2)

ex: consider a lossless compression $\quad \hat{x} = g(f(x)) = x \quad (\dim(z) = \dim(x))$

choose $f(x)$ such that $p_E(z) = f_\# p^*(x) = \mathcal{N}(z \mid 0, \mathbb{I})$

[push-forward of a distribution $f_\# p^*(x)$

- similar to function composition $\hat{x} = g(f(x)) = (g \circ f)(x)$
- for random variables $Z = f(X) \iff z \sim p(z) = f_\# p^*(x) \iff x \sim p^*(x), z = f(x)$]

$$p(\hat{x}) = g_\# p_E(z) = g_\# \mathcal{N}(z \mid 0, \mathbb{I}) = p^*(x) \text{ because lossless comp.}$$

due to symmetry of standard normal $\quad g(-z)_\# \mathcal{N}(z \mid 0, \mathbb{I}) = p^*(x)$

but $\quad \hat{x} = g(-f(x)) \neq x$