Lecture 22

Variational Autoencoders (2013)

Last time: Vanilla AE not suitable for generative modeling
- latent space doesn't have a fixed depth, so how to sample?
- latent space has gaps

VAE fixes these problems and can be used for generation.

Idea: add term to loss for that enforces latent space regularity & known depth.

Idea: Vanilla AE  \[ x \rightarrow z \]  VAE  \[ x \rightarrow \mathcal{N}(\mu, \sigma) \]  every pt in latent is smoothed out and nearby pts are similar in x.
Given \( x \) from data, we want to infer \( z \) statistically.

Statistical Treatment

\[
\frac{\text{Statistical Treatment}}{\text{latent variables}}
\]

\[
\text{Width of Gaussian can encode how "distinct" an input is.}
\]

\[
\text{Instead: try to learn tractable } q(z|x) \text { to be as close as possible to } p(z|x). \text{ "Variational inference"}
\]

\[
\text{Wat: } p(z|x) = \frac{p(x,z)p(z)}{p(x)} (\text{Bayes' The})
\]

\[
\begin{align*}
\text{need to know full joint, dirn of data } & \text{ difficult, often intractable.} \\
\text{close in } z & \text{ as measured by } o
\end{align*}
\]
\[
\minKL(p(z|x)||p(z|x)) \quad \Rightarrow \quad \int dz \frac{q(z|x)}{p(z|x)} \log \frac{q(z|x)}{p(z|x)} = \int dz \frac{q(z|x)}{p(z|x)} \log \frac{q(z|x)}{p(z|x)} = \int dz q(z|x) \log \frac{q(z|x)}{p(z|x)} = \int dz q(z|x) \log \frac{p(z|x)}{\mathbb{E}_{q(z|x)} [p(z|x)]} - KL(q(z|x)||p(z|x))
\]

So \( \min KL \to \)

\[
= \max \left[ \text{reconstruction likelihood} \right] - KL
\]

"reconstruction likelihood"

promotes \( p(z|x) \) to be similar to prior distribution on latent space \( \mathbb{P}(z) \)

for every \( x \).

promotes smooth, regular latent space
Implementation:

Assume: \( \rho(x) = N(0, 1)(x) \)

\[ q(x|z) = N(\mu(x), \sigma(x)) \]

(assume unrelated latent space)

2nd term (KL) becomes tractable!

\[
KL (N(\mu(x), \sigma(x)) \parallel N(0, 1)) = \frac{1}{2} \left[ -1 + \mu(x)^2 + \sigma(x)^2 + \ln \sigma(x)^2 \right]
\]

Assume: \( \rho(x|z) \sim e^{-\|x - h(z)\|^2 / 2\beta} \)

Then 1st term is tractable

decoder!

Separate for to minimize over.
\[
\begin{align*}
\text{max} \left( E_{p(x)} \log p(x|z) - KL \right) &= \text{encoder: } \mu(x), \sigma(x) \text{ for all NNs} \\
\text{mean} \left( E_q(z|x) \left[ \|x - f(z)\|^2 \right] + \beta \left[ -1 + \mu(x)^2 + \sigma(x)^2 \log \sigma(x)^2 \right] \right) &= \text{decoder: } f(z) \text{ in practice.}
\end{align*}
\]

\[
\begin{align*}
\text{min} \left( E_q(z|x) \left[ \|x - f(z)\|^2 \right] \right) &= \text{(latent loss)} \\
\text{need to sample from } q(z|x) \sim N(\mu(x), \sigma(x)) \\
\text{but also need to be able to differentiate } \| &\text{ wrt to parameters of } \mu(x), \sigma(x).
\end{align*}
\]

\[
\begin{align*}
\text{"reparameterization trick"} \\
\text{instead of } z \sim N(\mu(x), \sigma(x)) \Rightarrow \text{backpropable wrt. } &\text{par. of } \mu(x), \sigma(x).
\end{align*}
\]

"higher } \beta \rightarrow \text{smoother latent space!} \\
\text{often have } \beta \text{ to get good performance.}
Example VAE - MNIST from Keras blog.