

for conventional VAE.

Assume x_i for data, $P(x)$ is the data distribution

$$\text{find } z \rightarrow P(x) = \int f(x, \theta) P(z) dz$$

{ Some similar → increase variance → Use $P(x|z; \theta)$ to
dissimilar substitute $f(z, \theta)$

$$P(x|z; \theta) = N(x|f(z, \theta), b^2 I)$$

$$\begin{aligned} u &= f(z, \theta) \\ b &= \sqrt{b^2 I} \end{aligned}$$

For JT VAE $\rightarrow x$ to be represented by graph matrix

$$\therefore G(V, E) = X$$

$$V_i = \begin{bmatrix} \text{node} \\ \text{Info} \end{bmatrix} \underbrace{\begin{bmatrix} \text{Node} \\ \text{features} \end{bmatrix}}_{\rightarrow x_u}$$

X_{uv} to be the edge feature matrix \rightarrow bond type

$$X_{uv}^{(t)} = \tau(W_1^g w_u + W_2^g x_{uv} + W_3^g \sum_{w \in N(v) \setminus v} x_{wy}^{(t-1)})$$

τ is the activation function.

$N(v) \setminus v$ Neighbors of node v except v

After T steps \rightarrow Aggregate all messages as latent vector of each vertex v .

$$h_u = T(U_1^g X_u + \sum_{v \in N(u)} U_2^g v^{(T)})$$

The final graph representation $h_G = \frac{\sum h_i}{|V|}$

$$h_G = W_u * h_G + b_u$$

$$log b_G = W_b * h_G + b_b$$

Sample Σ from standard distribution $\mathcal{N}(0, I)$
 $\Sigma \sim \mathcal{N}(0, I)$.

Sample z_G from Latent space $\mathcal{N}(U_G, G_G)$

$$z_G = U_G + G_G \odot \Sigma, G_G = \exp\left(\frac{\log b_G}{2}\right)$$

element wise multiplication

For Tree encoder

GRU \rightarrow Gated Recurrent Unit

$$S_{i,j} = \sum_{k \in N(i) \setminus j} M_{kj} \rightarrow \text{Aggregated messages}$$

Updated Gates:

$$Z_{i,j} = g(W^z x_i + U^z S_{i,j} + b^z)$$

Determine how much information to be incorporate.

Reset Gates:

$$R_{i,j} = g(W^r x_i + U^r M_{kj} + b^r)$$

control how much information is contributed by each incoming message M_{kj} from neighboring nodes $N(i) \setminus j$

So candidate message:

$$\tilde{m}_{i,j} = \tanh(CW x_i + U \sum_{k \in N(i) \setminus j} R_{i,j} \odot M_{kj})$$

So, Final message.

$$m_{ij} = \frac{(1-z_{ij}) \theta S_{ij} + z_{ij} \theta \tilde{m}_{ki}}{\rho}$$

retained info ↓ origin info ↗ weighted new info