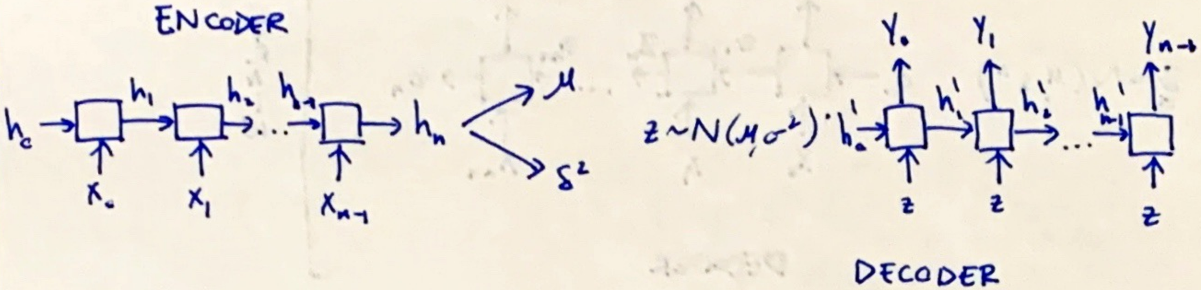
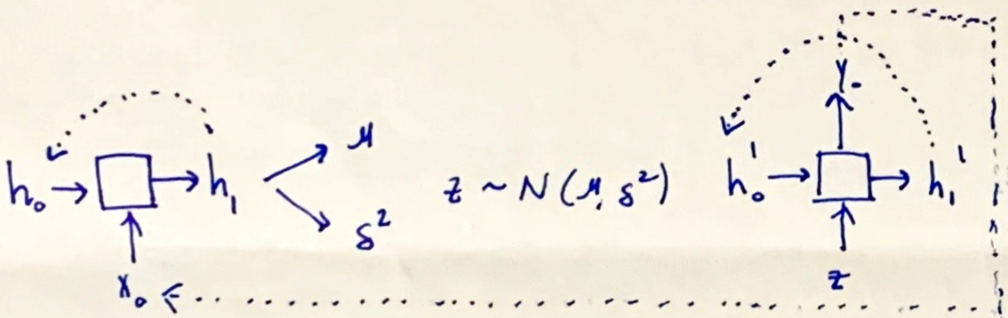
Variational RNN Architecture 1



TRAIN



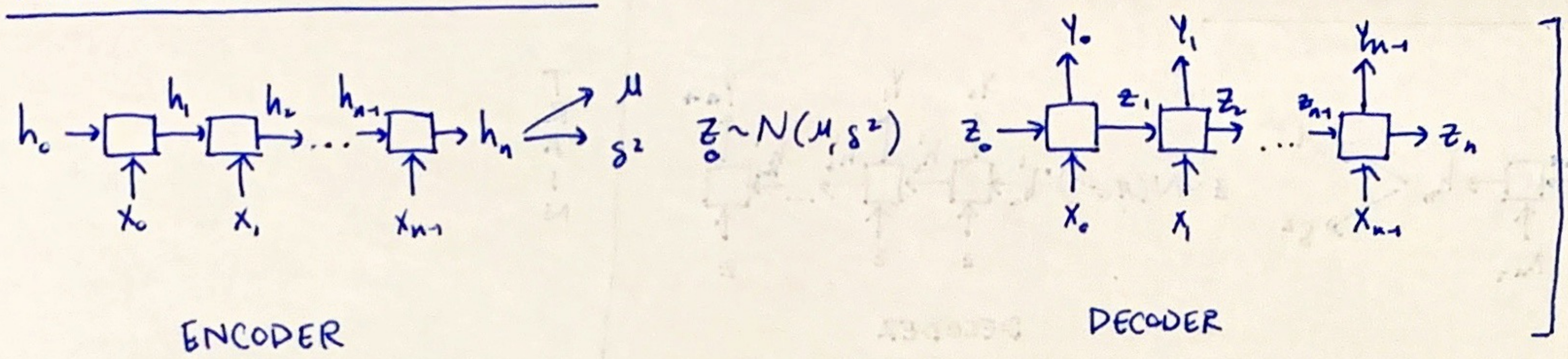
TEST

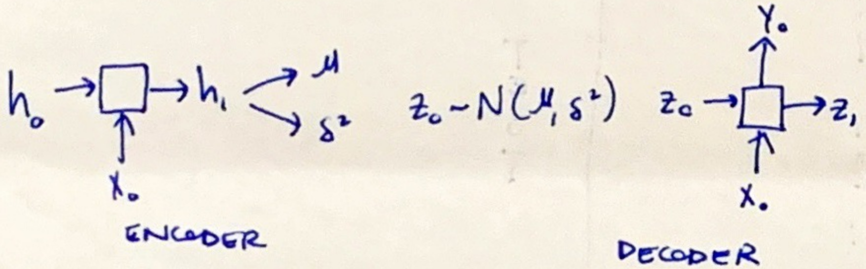
at next t, $h_0 = h_1$

$h'_0 = h'_1$

$x_0 = y_0$

Variational RNN Architecture 2





at next timestep,

$h_0 = h_1$

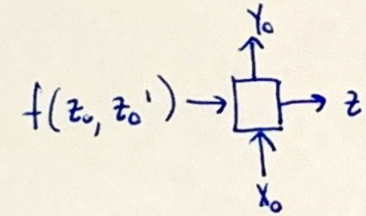
$z_0 = z_1$

$x_0 = y_0$

then, there is no need for the encoder after the 1st timestep → we could modify the decoder

$z_0' = \text{random}$

$z_0 \sim N(\mu, s^2)$

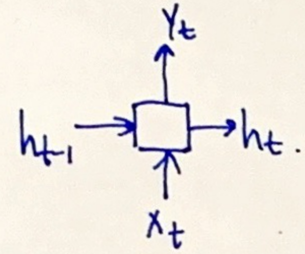


then, $z_0' = z_1$

$z_0 \sim N(\mu, s^2)$ from encoder.

TRUE VARIATIONAL RNN.

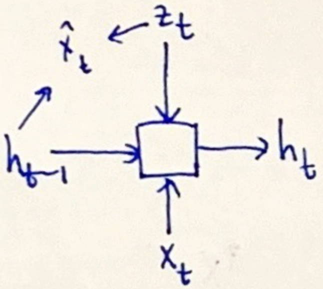
Let's look at a vanilla RNN.



$h_t = f(x_t, h_{t-1})$ f is the transition function.

$Y_t = g(h_t)$.

In a variational seq2seq Autoencoder, we would do this:



$h_t = f(x_t, z_t, h_{t-1})$.

encoder: $z_t \mid x_t$

$x_t, h_{t-1} \rightarrow \mu, \sigma^2$ $\quad z_t \sim N(\mu, \sigma^2)$

decoder: $x_t \mid z_t$

$z_t, h_{t-1} \rightarrow \mu, \sigma^2$ $\quad \hat{x}_t \sim N(\mu, \sigma^2)$

in the case of supervised learning.

$\hat{x}_t = \hat{Y}_t$