07 Nov 2024

Let the data be $D = {(x_i, y_i)}_{i=1}^N$ iid $\sim p(x, y)$.

here $x_i = { x_i^1, x_i^2, \ldots, x_i^k }$, where $x_j^i \in \mathbb{R}^d$ represents a sequence of k vectors of dimension d.They are called tokens in usual NLP models.

here $y_i = { y_i^1, y_i^2, \ldots, y_i^m }$, where $y_j^i \in \mathbb{R}^{d'}$ represents a sequence of m vectors of dimension d'. Note that $d' \neq d$. It represents a softmax distribution over a vocabulary of size d' in NLP models.

The models that map x_i to y_i are called seq2seq models.

We will stufy transformers as regularizers just as we studied CNNs as regularizers.

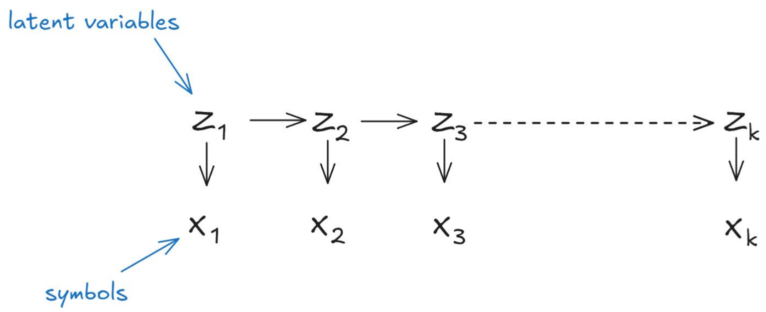
Historical models - Hidden Markov Models(HMMs)

In 1980s, Hidden Markov Models were used to model sequences before RNNs were popular.

We assume a sequence of latent variables $Z_1 \rightarrow Z_2 \rightarrow \ldots \rightarrow Z_k$.

This follows a Markovian assumption on the latent variables, meaning that the future state depends only on the current state.

At every transition, the latent variable emits a symbol x^j_i.



We model the joint distribution of the sequence as:

$$p(x) = p(x | z) p(z)$$

We model $p(x | z)$ as a Gaussian mixture model and use EM algorithm to estimate the parameters. We model $p(z)$ as a Markov chain and try to estimate the transition probabilities.

The intution behind this model is:

1. The model was usually used for speech modeling.
2. Humans also think something in their brain which is not observable just like the latent variables and emit sounds which are observable just like the symbols.

Recurrent Neural Networks

The problem with HMMs was that the length of the input sequence was always fixed.