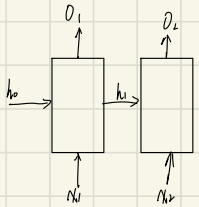
· RNN Autoregressive Model (Pixel RNN)

Model $p(x^t | x_{1:t-1}; \alpha^t)$

keep a summary of the history and recursively update it



summary update rule: $h_{t+1} = \tanh(W_{hh} h_t + W_{hx} x_{t+1})$

prediction: $O_{t+1} = W_{oy} h_{t+1}$ then pass through a non-linearity

summary initialization: $h_0 = b_0$

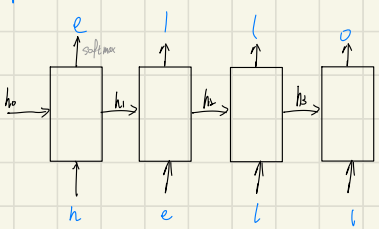
hidden state h_t is a summary of input seen till t

O_{t+1} specifies parameters for conditional $p(x_t | x_{1:t-1})$

Parameters: b_0, W_{hh}, W_{hx}, W_{oy} constant number of params w.r.t n

Autoregressive: $p(x = \text{hello}) = p(x_0 = h) \cdot p(x_1 = e | x_1 = h)$

$p(x_3 = l | x_1 = h, x_2 = e) \cdots$



pros:

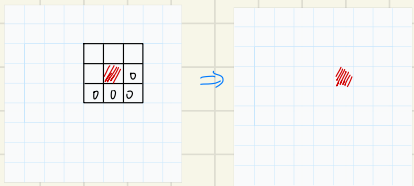
- arbitrary length

cons:

- Still requires ordering
- sequential likelihood function (slow for training)
- sequential generation (unavoidable for autoregressive model)

· CNN Autoregressive Models (Pixel CNN)

use convolutional architecture to predict next pixel given context (a neighborhood of pixels)



mask out the lower-right part of filter

(a pixel is not conditioned on the pixels "behind" it)

assume that the domain is governed by some underlying distribution P_{data}

given a dataset D of m samples from P_{data}

assume samples in D is iid

given a family of models $\mathcal{M}$. want to learn a "good" model $\hat{M} \in \mathcal{M}$ that defines a distribution $P_{\hat{M}}$

the definition of "best model" depends on what we want to do:

density estimation: want the full distribution

prediction: want $P(Y|X)$

structure or knowledge discovery: interested in the model itself

· KL - Divergence

$D(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} = E_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right]$ $\quad \left(\int_x P(x) \log \frac{P(x)}{Q(x)} dx\right)$

$D(P||Q) \geq 0$, $=0$ iff $P=Q$

$E_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = -E_{x\sim P}\left[\log \frac{Q(x)}{P(x)}\right] \geq -\log E_{x\sim P}\left[\frac{Q(x)}{P(x)}\right] = -\log 1 = 0$

KL is asymmetric $\quad D(P||Q) \neq D(Q||P)$

$D(P_{data} \,||\, P_\theta) = E_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{P_\theta(x)}\right] = \sum_x P_{data}(x) \log \frac{P_{data}(x)}{P_\theta(x)} = \sum_x P_{data}(x) \log P_{data}(x) - P_{data}(x) \log P_\theta(x)$

$\arg\min_\theta - \sum_x P_{data}(x) \log P_\theta(x) = \arg\max_\theta E_{x\sim P_{data}}[\log P_\theta(x)]$ minimizing the KL = maximizing the Expected log-likelihood

Although we can compare models, since we ignore $H(P_{data}) = \sum_x P_{data}(x) \log P_{data}(x)$,
we do not explicitly compute $KL(P||Q)$, thus we don't know how far we are from optimal (we approximate $E_{x\sim P_{data}}[\log P_\theta(x)]$

we approximate $E_{x\sim P_{data}}[\log P_\theta(x)]$ with empirical log-likelihood $E_D[\log P_\theta(x)] \quad \frac{1}{|D|} \sum_{x\in D} \log P_\theta(x)$

· Monte Carlo Estimation

$E_{x\sim P}[g(x)] \approx E_D[g(x)] = \frac{1}{|D|} \sum_{x\in D} g(x)$ where $x \in D$ are iid sampled from P

MC estimation is unbias:

$$E_D\left[\frac{1}{|D|} \sum_{x\in D} g(x)\right] = \frac{1}{|D|} \cdot E_P\left[\sum_{x\in D} g(x)\right] = E_P[g(x)]$$

convergence: by law of large number

$\frac{1}{|D|} \sum_{x\in D} g(x) \to E_P[g(x)]$ as $|D| \to +\infty$

variance

$$\mathrm{Var}_p\left(\frac{1}{|D|}\sum_{x\in D} g(x)\right) = \frac{1}{|D|}\mathrm{Var}\left(g(x)\right)$$

larger $|D|$, smaller variance