Autoregressive Models
(FVSBN,NADE,MADE)

Ayush Raina

Indian Institute of Science

October 6, 2024

Aim

We want to model the distribution $p(\mathbf{X}) = p(X_1, X_2, ..., X_D)$ where $X \in \mathbb{R}^D$ so that:

- **Generation:** If we sample $X_{new} \sim p(\mathbf{X})$, it should look like the data
- **Density Estimation:** If a training point X is similar to data, $p(X)$ should be high

Autoregressive Models

These models do not assume any conditional independence assumptions and they use chain rule factorization given by:

$$\boxed{p(X) = \prod_{i=1}^{D} p(X_i | X_{<i})}$$

where $X_{<i} = \{X_1, X_2, ..., X_{i-1}\}$ and $X \in \mathbb{R}^D$

Autoregressive Models

$$\boxed{p(X) = \prod_{i=1}^{D} p(X_i|X_{<i})}$$

But the number of parameters required to model the D factors of the above distribution are $1, 2, 4, 8, ..., 2^{D-1}$

hence total number of parameters required are

$$\boxed{1 + 2 + 4 + 8 + ... + 2^{D-1} = 2^D - 1}$$

which is exponential in D.
It is not feasible to learn exponential number of parameters.

Autoregressive Models

These models assume a functional form to approximate the factors of the distribution.

$$p(X_i|X_{<i}) = f(X_i, X_{<i}; \theta_i)$$

where f is a function which approximates the factor $p(X_i|X_{<i})$ and θ_i are the parameters of the function f.

Autoregressive Models

We cannot use fully connected neural network here because while predicting the $p(X_i)$, the inputs used are $X_{<i}$

In the fully connected neural network, all the inputs are used to predict the output.

Fully Visible Sigmoid Belief Net(FVSBN)

1. In this model,for $X = (X_1, X_2, ..., X_D) \in \mathbb{R}^{\mathbb{D}}$ each $X_i \sim Bernoulli(p_i)$ where $p_i = p(X_i = 1|X_{<i})$.
2. Functional form of $p(X_i|X_{<i})$ is given by Sigmoid function:

$$\boxed{p(X_i = 1|X_{<i}) = \sigma(\sum_{j=1}^{i-1} w_j^{(i)} X_j + b_i)}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, $w_j^{(i)}$ are the weights and b_i is the bias.

3. $\sum_{j=1}^{i-1} w_j^{(i)} X_j + b_i$ is the linear combination of inputs
4. $\Sigma = \{w_j^{(i)}, b_i\}$ are the parameters of the model.

For predicting the $p(X_i)$ only inputs used are $X_{<i}$

Fully Visible Sigmoid Belief Net(FVSBN)

$$\boxed{p(X_i = 1 | X_{<i}) = \sigma(\sum_{j=1}^{i-1} w_j^{(i)} X_j + b_i)}$$

How many parameters are required now ?

1. for each i, we have i weights i.e $w_0^{(i)}, w_1^{(i)}, ..., w_{i-1}^{(i)}$ and 1 bias b_i
2. Hence total number of parameters required are

$$\boxed{(1 + 2 + 3 + ... + D) + D = \frac{D(D+1)}{2} + D = \frac{D(D+3)}{2}}$$

Better than exponential number of parameters.

Experimental Results

Neural Autoregressive Density Estimator(NADE)

Here we add a neural network layer to approximate the factors $p(X_i|X_{<i})$

Output: n dimensional vector $p(X_i|X_{<i})$ for $i = 1, 2, ..., D$

But the kth output should see inputs $X_{<k}$ only.

Adding a neural network layer

Consider $X \in \mathbb{R}^N$ and $h_k \in \mathbb{R}^D$ be the hidden representation for the kth output.

For the kth output, the hidden representation will be computed using previous $k-1$ inputs.

$$\boxed{h_k = \sigma(W_{.,<k}X_{<k} + b)}$$

where W is the weight matrix which is shared during the computation of h_k for all k.

Here $W_{.,<k}$ represents first $k-1$ columns of W.

Computing output from hidden representation

We now compute the output $p(X_k|X_{<k})$ using the hidden representation h_k as follows:

$$y_k = p(X_k|h_k) = \sigma(V_k h_k + c_k)$$

Final Equations

So here is our model:

$$h_k = \sigma(W_{.,<k} X_{<k} + b)$$

$$y_k = p(X_k | h_k) = \sigma(V_k h_k + c_k)$$

$\Sigma = \{W, V, b, c\}$ are the parameters of the model.

Calculating the number of parameters

But how many parameters are there:

1. $W \in \mathbb{R}^{D \times N} \implies DN$ parameters
2. $b = \{b_1, b_2, ..., b_D\} \implies D$ parameters
3. $V_k \in \mathbb{R}^D$ for $k = 1, 2, ..., N \implies DN$ parameters
4. $c = \{c_1, c_2, ..., c_N\} \implies N$ parameters
5. $h_1 \in \mathbb{R}^D$ is also a parameter $\implies D$ parameters

Hence total number of parameters are $2DN + 2D + N \sim O(DN)$

Generation: Binary Random Variable

1. Compute $p(X_1 = 1) = \sigma(V_1 h_1 + c_1) = t_1$ where $t_1 \in [0, 1]$
2. Sample $m_1 \sim Unif[0, 1]$, if $m_1 < t_1$ then $X_1 = 1$ else $X_1 = 0$
3. Compute $p(X_2 = 1|X_1) = \sigma(V_2 h_2 + c_2) = t_2$ where $t_2 \in [0, 1]$
4. Sample $m_2 \sim Unif[0, 1]$, if $m_2 < t_2$ then $X_2 = 1$ else $X_2 = 0$

Repeat this process for $X_3, X_4, ..., X_D$ to generate a sample from the distribution $p(X)$

Experimental Results

Masked Autoencoder Density Estimator(MADE)

Thank You!

Thank You!