MADE:
Masked Autoencoder for Distribution Estimation

CS 598: Deep Generative and Dynamical Models

Instructor: Arindam Banerjee

September 7, 2021

Introduction, Motivation

- Autoregressive models: Product of conditional distributions

$$p(x_1, \ldots, x_D) = \prod_{d=1}^{D} p(x_d | x_{<d})$$

- Sequential training and prediction
- Consider the binary case
  - Conditional probability tables, 2^D complexity
  - Conditional probability models, still sequential $O(D)$
- Autoencoders: Parallel training and prediction
  - However, do not satisfy autoregressive property
  - Not valid probabilistic models
- Motivation: Can the two be combined?

Background: Autoencoders

- Autoencoders learn latent representations
  - Latent representation $x \mapsto h(x)$
  - Reconstruct original $h(x) \mapsto \hat{x}$
- Example: $x \in \{0,1\}^D, \hat{x} \in [0,1]^d$

$$h(x) = g(Wx + b)$$

$$\hat{x} = \sigma(Vh(x) + c)$$

- Training using cross-entropy loss

$$\ell(x) = \sum_{d=1}^{D} -x_d \log \hat{x}_d - (1 - x_d) \log(1 - \hat{x}_d)$$

- Not proper likelihood, not normalized

$$q(x) = \prod_{d=1}^{D} \hat{x}_d^{x_d} (1 - \hat{x}_d)^{1 - x_d} \ , \quad \sum_x q(x) \neq 1$$

AR models for Distribution Estmation

- Chain rule (product rule) for joint distributions

$$p(x_1, \ldots, x_D) = \prod_{d=1}^{D} p(x_d | x_{<d})$$

- Valid likelihood based on

$$\hat{x}_d = p(x_d = 1 | x_{<d}) \qquad 1 - \hat{x}_d = p(x_d = 0 | x_{<}d)$$

  - $\hat{x}_d$ only depends on $x_{<d}$
  - Autoregressive property

- Minimize negative log-likelihood

$$\ell(x) = -\log p(\mathrm{x}) = \sum_{d=1}^{D} -\log p(x_d | x_{<d})$$

$$= \sum_{d=1}^{D} -x_d \log \hat{x}_d - (1 - x_d) \log(1 - \hat{x}_d)$$

Masked Autoencoders

- Binary mask matrices to ensure autoregressive property
- Single layer autoencoder

$$h(x) = g((W \odot M^W)x + b)$$

$$\hat{x} = \sigma((V \odot M^V)h(x) + c)$$

- Ensuring autoregressive property
  - Each hidden layer gets a $m \in \{1, \ldots, D-1\}$
  - $m(k), k = 1, \ldots, K$: number of input units it can connect to
  - Mask matrix constructed based on $m(k)$

$$M^W_{k,d} = 1_{m(k) \geq d} = \begin{cases} 1\ , & \text{if } m(k) \geq d\ , \\ 0\ , & \text{otherwise} \end{cases}$$

  - Output unit $x_{d'}$ can only connect to $k : d' > m(k)$

$$M^V_{d',k} = 1_{d' > m(k)} = \begin{cases} 1\ , & \text{if } d' > m(k)\ , \\ 0\ , & \text{otherwise} \end{cases}$$

Properties, Variations

- Output-input connectivity $M^{V,W} = M^V M^W$
  - $M^{V,W}_{d,d'}$: # paths between output $x_{d'}$ and input x_d
- $M^{V,W}$ is strictly lower diagonal
  - For $d' \leq d$,

$$M^{V,W}_{d',d} = \sum_{k=1}^{K} M^V_{d',k} M^W_{k,d} = \sum_{k=1}^{K} 1_{d'>m(k)} 1_{m(k)\geq d} = 0$$

- Only need to assign $m(k)$ to each hidden unit k
  - Sample i.i.d. from a uniform distribution over $\{1, \ldots, D-1\}$
- Residual connections from input to output
  - Output $x_{d'}$ can connect to input $x_d, d < d'$
  - Mask matrix M^A is lower triangular

$$\hat{\mathrm{x}} = \sigma((V \odot M^V)\mathrm{h}(\mathrm{x}) + (A \odot M^A)\mathrm{x} + \mathrm{c})$$

Deep MADE

- Extend to L hidden layers, weight matrices $W^1, \ldots, W^L$
- For each unit, assign maximum number of connected inputs
  - $m^l(k)$: max # connected inputs for k^{th} unit in l^{th} layer
  - As before, sample uniformly
  - Avoid unconnected units, $m^l(k) \geq \min_{\tilde{k}} m^{l-1}(\tilde{k})$
- Mask connecting any layer l to previous layer

$$M^{W^l}_{k',k} = 1_{m^l(k') \geq m^{l-1}(k)} = \begin{cases} 1\ , & \text{if } m^l(k') \geq m^{l-1}(k), \\ 0\ , & \text{otherwise} \end{cases}$$

  - Input layer: $l = 0$, $m^0(d) = d$, W^1: Weights of first layer
- Mask connecting output to last layer L

$$M^V_{d,k} = 1_{d > m^L(k)} = \begin{cases} 1\ , & \text{if } d > m^L(k)\ , \\ 0\ , & \text{otherwise} \end{cases}$$

Deep MADE (Contd.)

- For layers $l = 1, \ldots, L$, parameters (W^l, b^l)

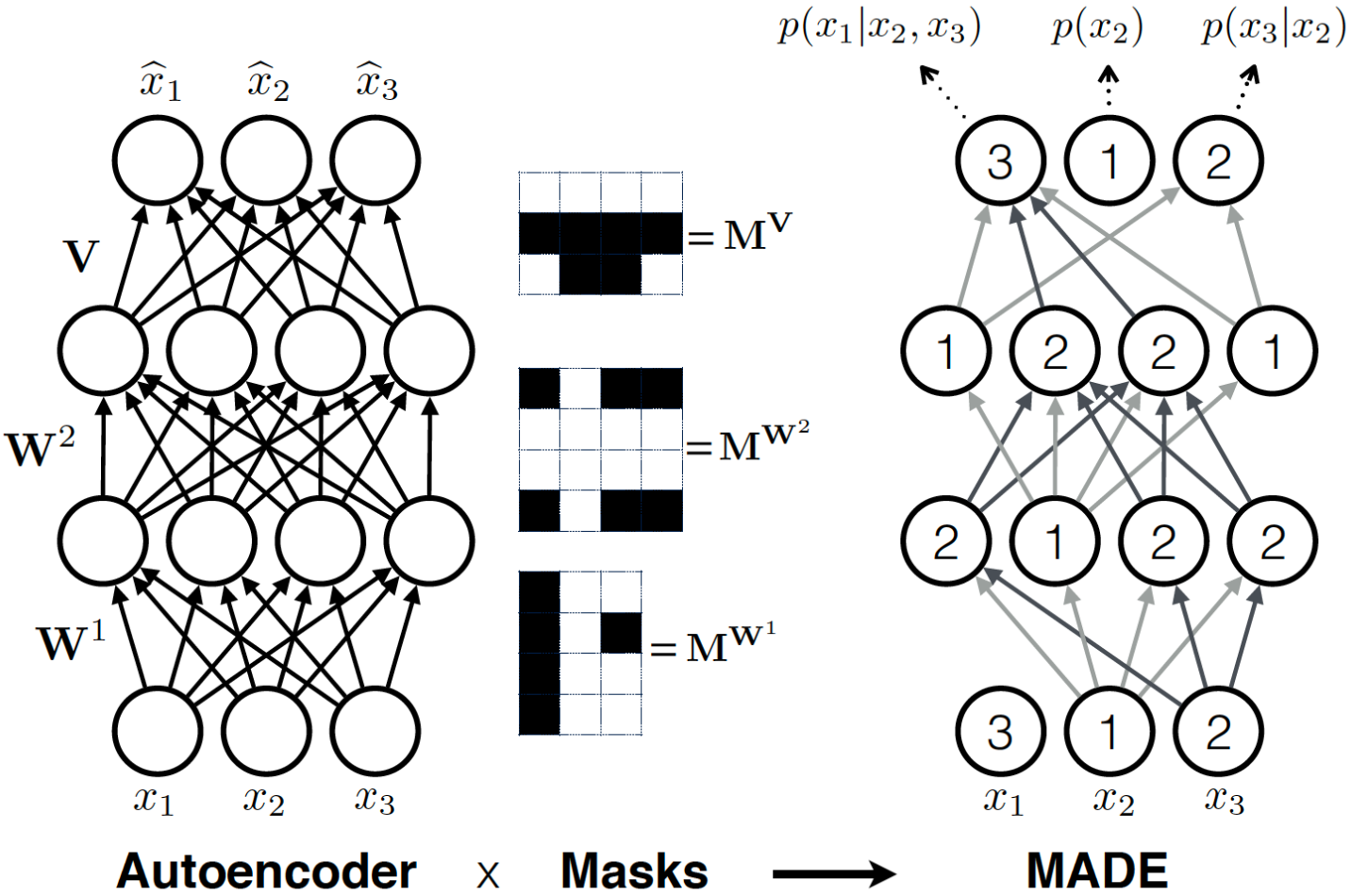
$$\mathrm{h}^l(\mathrm{x}) = \mathrm{g}((W^l \odot M^{W^l})\mathrm{h}^{l-1}(\mathrm{x}) + \mathrm{b}^l)$$

- Layer $l = 0$ is the input, $\mathrm{h}^l(\mathrm{x}) = \mathrm{x}$
- Output layer, parameters (V, c)

$$\hat{\mathrm{x}} = \sigma((V \odot M^V)\mathrm{h}^L(\mathrm{x}) + \mathrm{c})$$

- Mini-batch (adaptive) first-order training, e.g., SGD, Adam, etc.

Deep MADE


$p(x_1|x_2,x_3)$
$p(x_2)$
$p(x_3|x_2)$
$\widehat{x}_1$
$\widehat{x}_2$
$\widehat{x}_3$
$\mathbf{V}$
$\mathbf{W}^2$
$\mathbf{W}^1$
$= \mathbf{M}^{\mathbf{V}}$
$= \mathbf{M}^{\mathbf{W}^2}$
$= \mathbf{M}^{\mathbf{W}^1}$
3 1 2
1 2 2 1
2 1 2 2
3 1 2
x_1 x_2 x_3
Autoencoder x Masks ⟶ MADE

Deep MADE Algorithm

Algorithm 1 Computation of $p(\mathbf{x})$ and learning gradients for MADE with order and connectivity sampling. D is the size of the input, L the number of hidden layers and K the number of hidden units.

Input: training observation vector $\mathbf{x}$
Output: $p(\mathbf{x})$ and gradients of $-\log p(\mathbf{x})$ on parameters

Sampling $\mathbf{m}^l$ vectors
$\mathbf{m}^0 \leftarrow \text{shuffle}([1, \dots, D])$
for l from 1 to L **do**
  for k from 1 to K^l **do**
    $m^l(k) \leftarrow \text{Uniform}([\min_{k'} m^{l-1}(k'), \dots, D-1])$
  end for
end for

# Constructing masks for each layer
for l from 1 to L **do**
  $\mathbf{M}^{\mathbf{W}^l} \leftarrow \mathbf{1}_{\mathbf{m}^l \geq \mathbf{m}^{l-1}}$
end for
$\mathbf{M}^{\mathbf{V}} \leftarrow \mathbf{1}_{\mathbf{m}^0 > \mathbf{m}^L}$

Computing $p(\mathbf{x})$
$\mathbf{h}^0(\mathbf{x}) \leftarrow \mathbf{x}$
for l from 1 to L **do**
  $\mathbf{h}^l(\mathbf{x}) \leftarrow \mathbf{g}(\mathbf{b}^l + (\mathbf{W}^l \odot \mathbf{M}^{\mathbf{W}^l})\mathbf{h}^{l-1}(\mathbf{x}))$
end for
$\widehat{\mathbf{x}} \leftarrow \text{sigm}(\mathbf{c} + (\mathbf{V} \odot \mathbf{M}^{\mathbf{V}})\mathbf{h}^L(\mathbf{x}))$
$p(\mathbf{x}) \leftarrow \exp\left(\sum_{d=1}^{D} x_d \log \widehat{x}_d + (1-x_d)\log(1-\widehat{x}_d)\right)$

Computing gradients of $-\log p(\mathbf{x})$
$\text{tmp} \leftarrow \widehat{\mathbf{x}} - \mathbf{x}$
$\delta\mathbf{c} \leftarrow \text{tmp}$
$\delta\mathbf{V} \leftarrow (\text{tmp}\ \mathbf{h}^L(\mathbf{x})^\top) \odot \mathbf{M}^{\mathbf{V}}$
$\text{tmp} \leftarrow (\text{tmp}^\top (\mathbf{V} \odot \mathbf{M}^{\mathbf{V}}))^\top$
for l from L to 1 **do**
  $\text{tmp} \leftarrow \text{tmp} \odot \mathbf{g}'(\mathbf{b}^l + (\mathbf{W}^l \odot \mathbf{M}^{\mathbf{W}^l})\mathbf{h}^{l-1}(\mathbf{x}))$
  $\delta\mathbf{b}^l \leftarrow \text{tmp}$
  $\delta\mathbf{W}^l \leftarrow (\text{tmp}\ \mathbf{h}^{l-1}(\mathbf{x})^\top) \odot \mathbf{M}^{\mathbf{W}^l}$
  $\text{tmp} \leftarrow (\text{tmp}^\top (\mathbf{W}^l \odot \mathbf{M}^{\mathbf{W}^l}))^\top$
end for
return $p(\mathbf{x}), \delta\mathbf{b}^1, \dots, \delta\mathbf{b}^L, \delta\mathbf{W}^1, \dots, \delta\mathbf{W}^L, \delta\mathbf{c}, \delta\mathbf{V}$

Order-agnostic Training

- Training assumes a fixed ordering of input

$$p(x_1, \ldots, x_D) = \prod_{d=1}^{D} p(x_d | x_{<d})$$

- Order agnostic: Training with different (all) orderings
  - Sampling an ordering before each mini-batch update
  - Set $\mathrm{m}^0 = [m^0(1), \ldots, m^0(D)]$ as permutation of $[1, \ldots, D]$
  - First hidden layer matrix can be created based on m^0
- (Potential) Advantages
  - Ensemble of AR models, over different orderings
  - Missing/occluded input can be ordered with missing at the end
  - Avoid specific order dependent artifacts

Connectivity-agnostic Training

- Masks in each layer choose specific connectivity
- Connectivity constraints $m^l(k)$ chosen at random
- Connectivity-agnostic: Resample $m^l(k)$ before each mini-batch
  - Creating masks is easy to parallelize
  - Sample m^l, construct $M^{W^l} = 1_{m^l \geq m^{l-1}}$
- Distinguish between no connection vs. connection with zero-valued unit

$$\mathrm{h}^l(\mathrm{x}) = \mathrm{g}((W^l \odot M^{W^l})\mathrm{h}^{l-1}(\mathrm{x}) + (U^l \odot M^{W^l})1 + \mathrm{b}^l)$$

  - Empirically not always helpful
- Sampling masks for every example: over-regularization
  - Choose a fixed set of masks, cycle through the list

Related Work

- Variant of single layer MADE [Bengio & Bengio, 2000]
- Neural Autoregressive Distribution Estimator (NADE) [Larochelle & Murray, 2011]
- Deep extension to NADE [Uria et al., 2014]
  - Still needs D steps to ensure AR property
- Deep Autoregressive Networks (DARN) [Gregor et al., 2014]
  - Training time is same as autoencoders
  - Representation: binary stochastic units
  - Evaluation of probability needs summing over configurations

Complexity of Different Models

Table 1. Complexity of the different models in Table 6, to compute an exact test negative log-likelihood. R is the number of orderings used, D is the input size, and K is the hidden layer size (assuming equally sized hidden layers).

Model	O_{NLL}
RBM 25 CD steps	$O(\min(2^D K,\ D2^K))$
DARN	$O(2^K D)$
NADE (fixed order)	$O(DK)$
EoNADE 1hl, R ord.	$O(RDK)$
EoNADE 2hl, R ord.	$O(RDK^2)$
MADE 1hl, 1 ord.	$O(DK+D^2)$
MADE 2hl, 1 ord.	$O(DK+K^2+D^2)$
MADE 1hl, R ord.	$O(R(DK+D^2))$
MADE 2hl, R ord.	$O(R(DK+K^2+D^2))$

Results: UCI Datasets

Table 4. Negative log-likelihood test results of different models on multiple datasets. The best result as well as any other result with an overlapping confidence interval is shown in bold. Note that since the variance of DARN was not available, we considered it to be zero.

Model	Adult	Connect4	DNA	Mushrooms	NIPS-0-12	OCR-letters	RCV1	Web
MoBernoullis	20.44	23.41	98.19	14.46	290.02	40.56	47.59	30.16
RBM	16.26	22.66	96.74	15.15	277.37	43.05	48.88	29.38
FVSBN	**13.17**	12.39	83.64	10.27	276.88	39.30	49.84	29.35
NADE (fixed order)	**13.19**	11.99	84.81	9.81	**273.08**	**27.22**	46.66	28.39
EoNADE 1hl (16 ord.)	**13.19**	12.58	82.31	9.69	**272.39**	**27.32**	**46.12**	**27.87**
DARN	13.19	11.91	81.04	**9.55**	274.68	≈28.17	≈**46.10**	≈28.83
MADE	**13.12**	**11.90**	83.63	9.68	280.25	28.34	47.10	28.53
MADE mask sampling	**13.13**	**11.90**	**79.66**	9.69	277.28	30.04	46.74	**28.25**

Results: Binarized MNIST

Table 6. Negative log-likelihood test results of different models on the binarized MNIST dataset.

Model	$-\log p$	
RBM (500 h, 25 CD steps)	≈ 86.34	Intractable
DBM 2hl	≈ 84.62	
DBN 2hl	≈ 84.55	
DARN n_h=500	≈ 84.71	
DARN n_h=500, adaNoise	≈ 84.13	
MoBernoullis K=10	168.95	Tractable
MoBernoullis K=500	137.64	
NADE 1hl (fixed order)	88.33	
EoNADE 1hl (128 orderings)	87.71	
EoNADE 2hl (128 orderings)	85.10	
MADE 1hl (1 mask)	88.40	
MADE 2hl (1 mask)	89.59	
MADE 1hl (32 masks)	88.04	
MADE 2hl (32 masks)	86.64	

MNIST Results: Samples

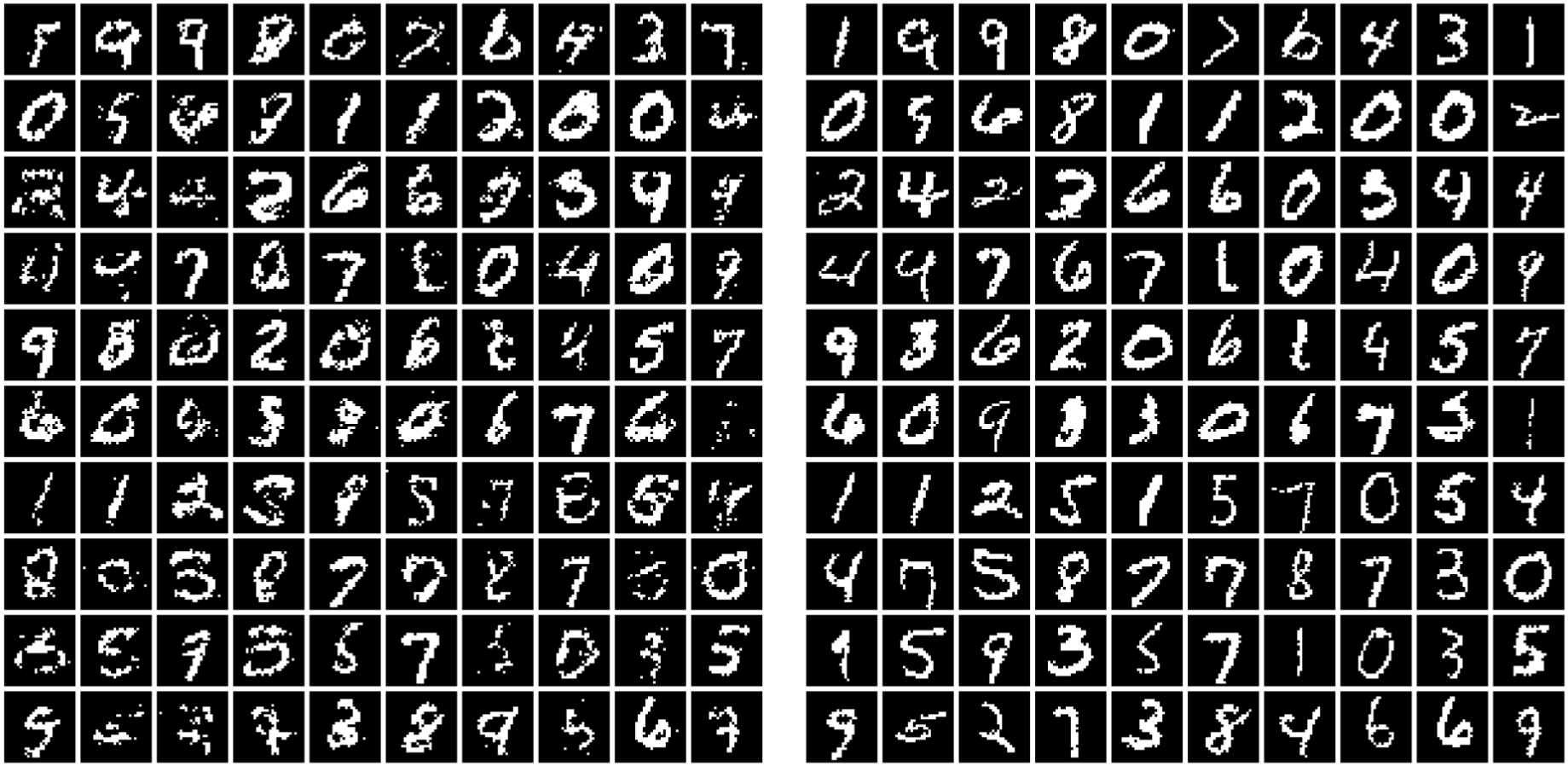


Figure 3. Left: Samples from a 2 hidden layer MADE. Right: Nearest neighbour in binarized MNIST.

MNIST Results: Impact of Number of Masks

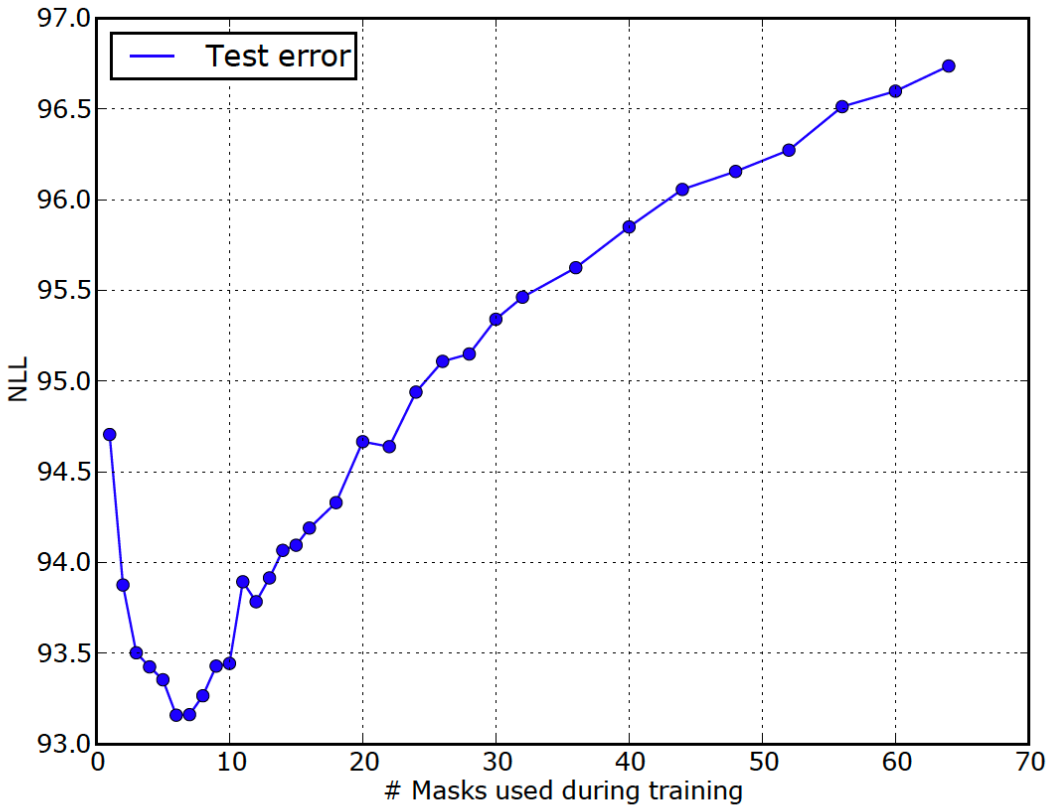


Figure 2. Impact of the number of masks used with a single hidden layer, 500 hidden units network, on binarized MNIST.

Conclusions

- Motivation: Avoid sequential nature of AR models
- Challenging for high-dimensional input
- MADE: models high-d probability
- Probability computation is one-pass through autoencoder
- Avoids sequential computation in AR, keeps AR property

References

- M. Germain, K. Gregor, I. Murray, H. Larochelle, MADE: Masked autoencoder for distribution estimation, ICML, 2015.