Analyzing the Dynamics of Gated Auto-encoders

Daniel Jiwoong Im

Outline

- Preliminaries
- Methodology
- Experiments
- Conclusion

Preliminary : Auto-encoders

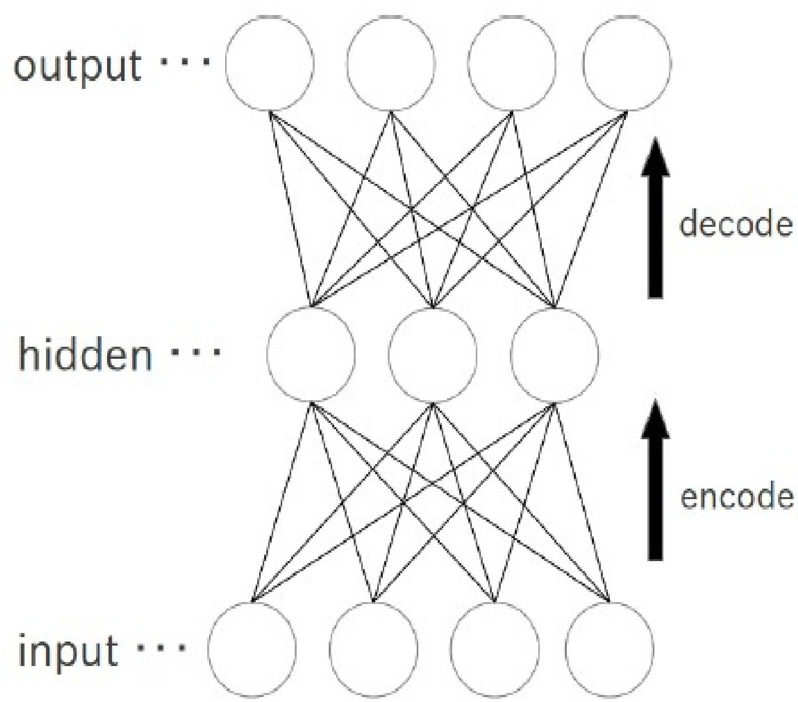
- Unsupervised Learning
- Given Input X -> Reproduce X

Preliminary : Auto-encoders

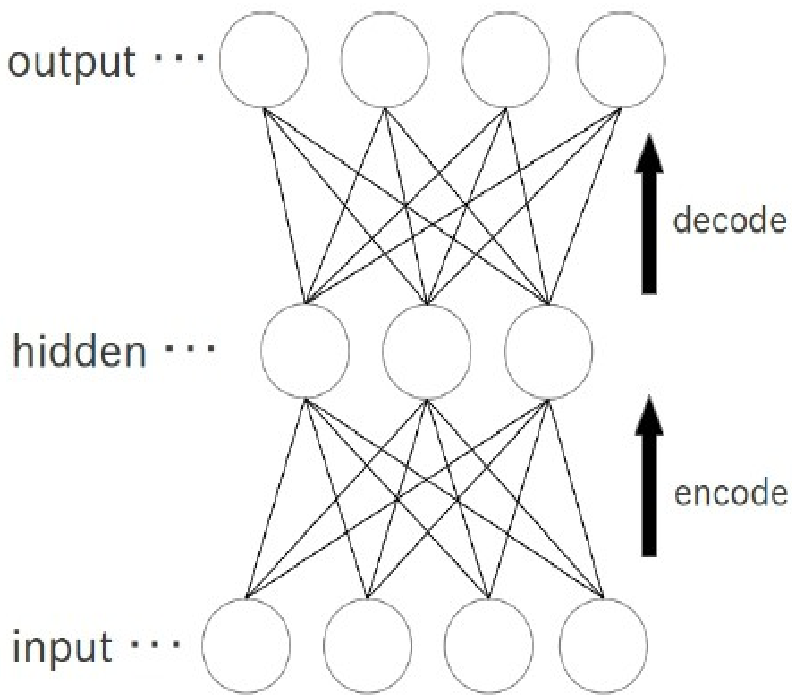
- Unsupervised Learning
- Input X -> Encode -> Decode -> reproduced X

Preliminary : Auto-encoders

- Unsupervised Learning
- Input X -> Encode -> Decode -> reproduced X

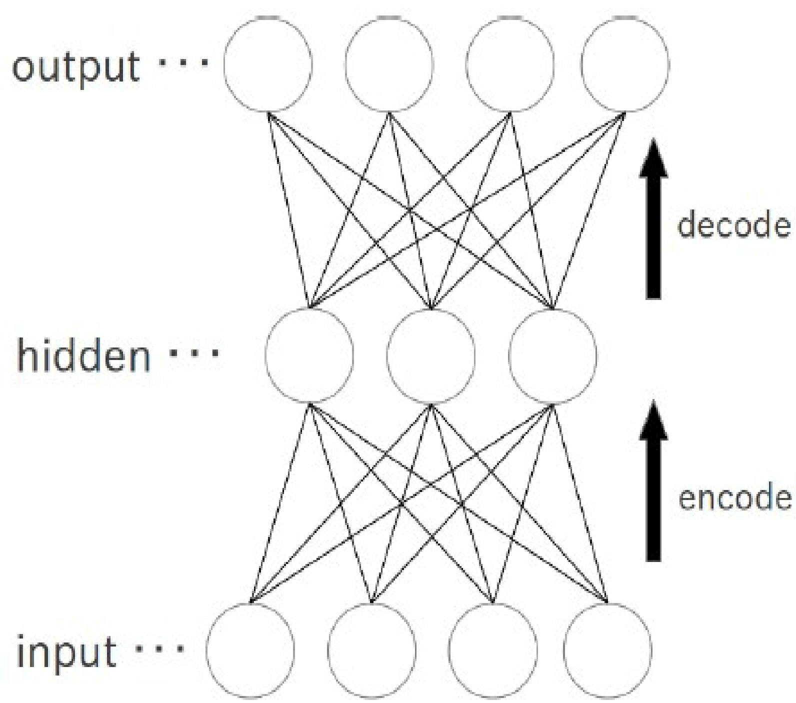


Preliminary : Auto-encoders



$$\mathbf{h} = f(\mathbf{x})$$

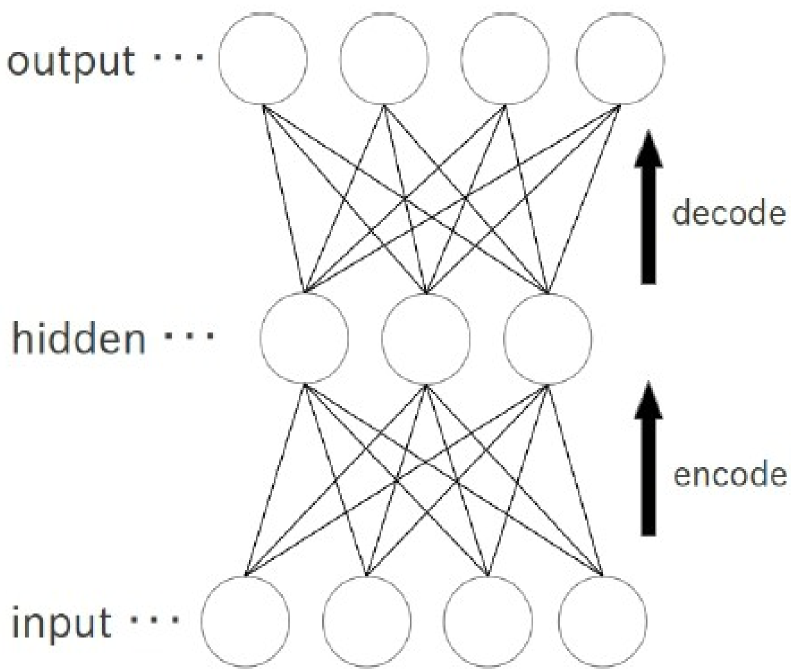
Preliminary : Auto-encoders



$$\mathbf{x} = r(\mathbf{h})$$

$$\mathbf{h} = f(\mathbf{x})$$

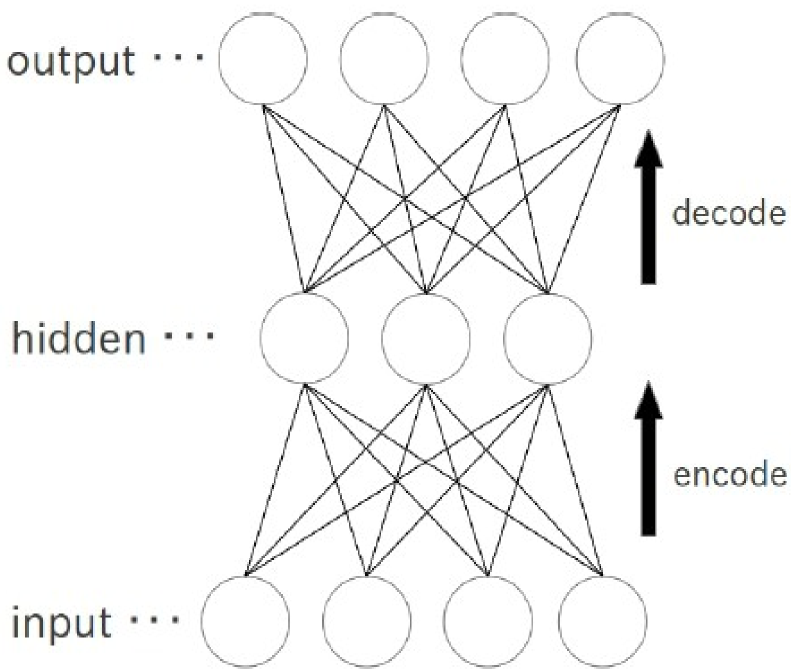
Preliminary : Auto-encoders



$$\mathbf{x} = r(\mathbf{h}) = \sum_i W_{ji}\mathbf{h}_i + \mathbf{c}_i$$

$$\mathbf{h} = f(\mathbf{x}) = \sigma(\sum_j W_{ij}\mathbf{x}_j + \mathbf{b}_j)$$

Preliminary : Auto-encoders



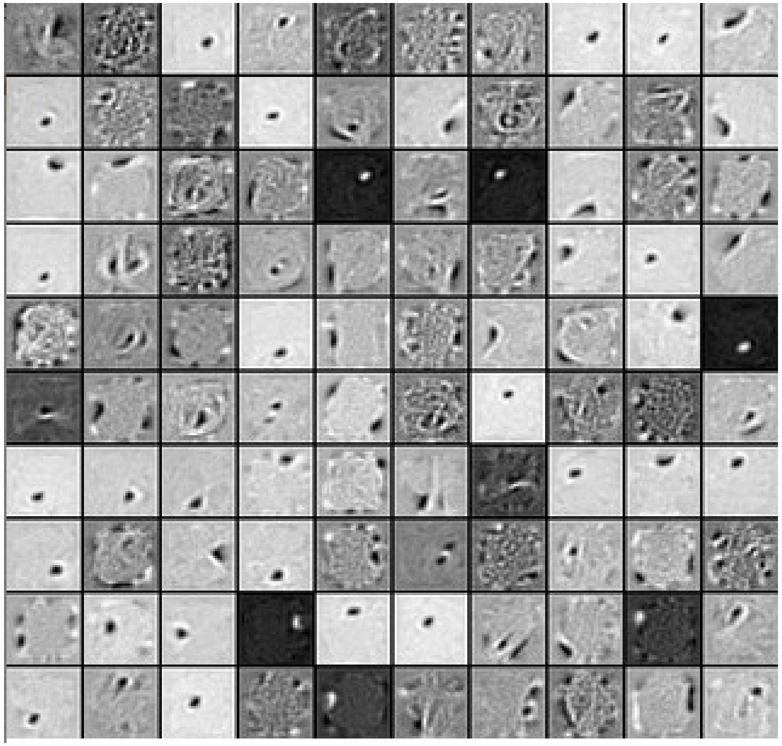
$$\mathbf{x} = r(\mathbf{h}) = \sum_i W_{ji}\mathbf{h}_i + \mathbf{c}_i$$

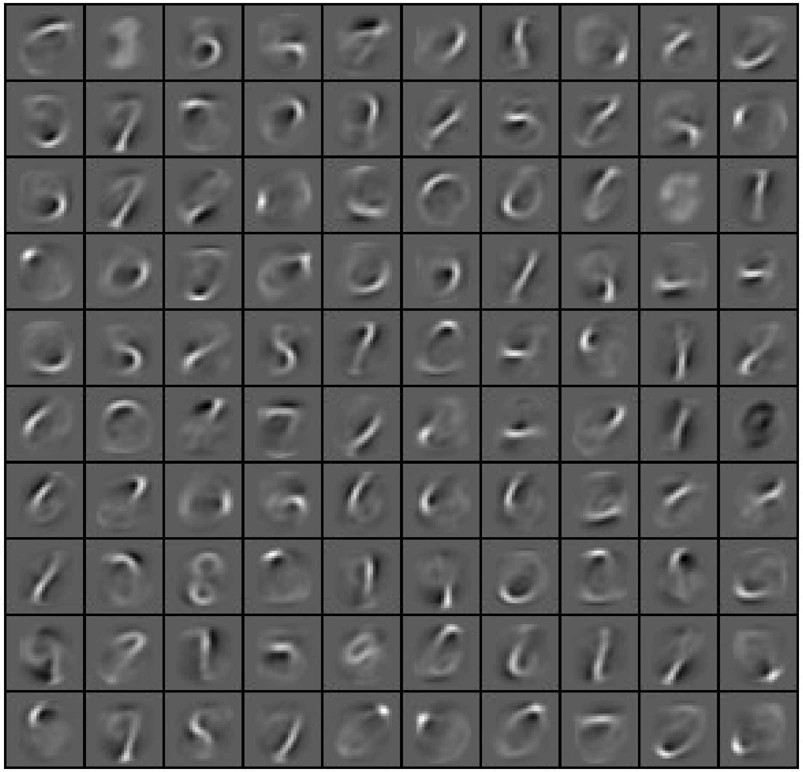
$$\mathbf{h} = f(\mathbf{x}) = \sigma(\sum_j W_{ij}\mathbf{x}_j + \mathbf{b}_j)$$

$$\min_\theta J(\mathbf{x}) = \min_\theta \|\mathbf{x} - r(f(\mathbf{x}))\|^2$$

Preliminary : Auto-encoders

Preliminary : Auto-encoders





Preliminary : Gated Auto-encoders

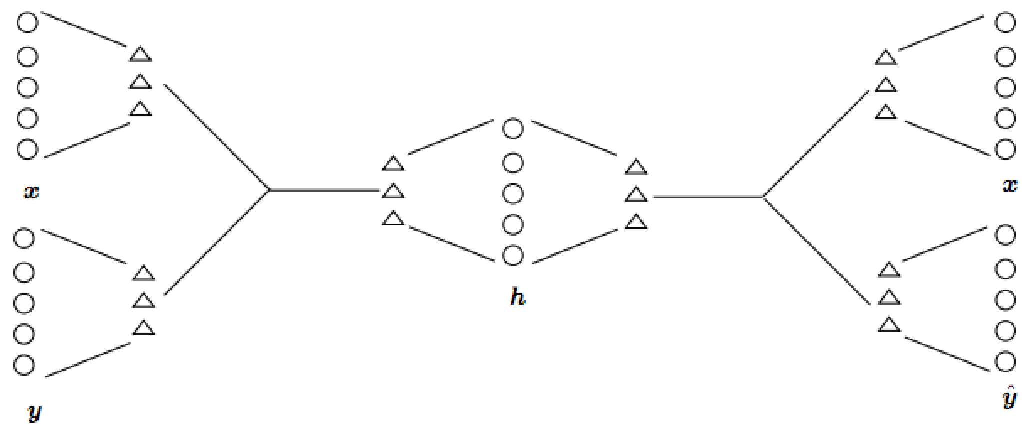
- Unsupervised Learning
- Given Input X,Y -> reproduced X|Y

Preliminary : Gated Auto-encoders

- Unsupervised Learning
- Input X,Y -> Encode X|Y -> Decode -> reproduced X|Y

Preliminary : Gated Auto-encoders

- Unsupervised Learning
- Input X,Y -> Encode X|Y -> Decode -> reproduced X|Y



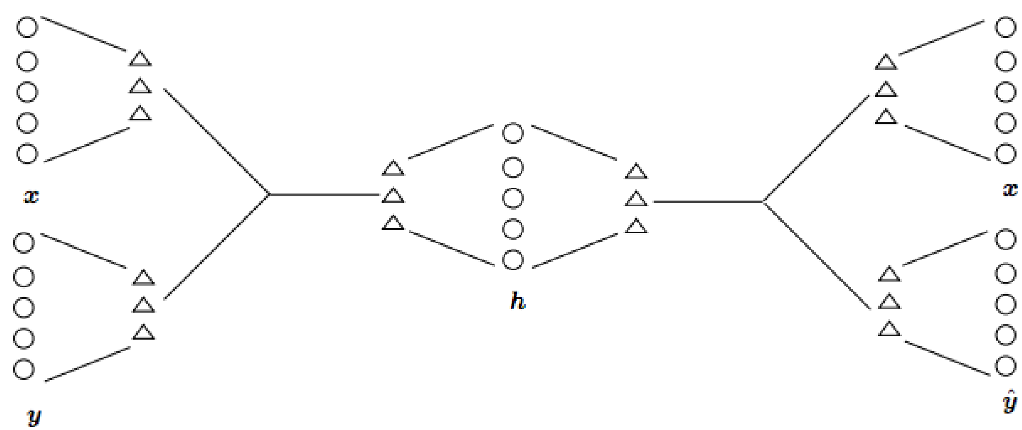
$$\mathbf{h} = f(\mathbf{x}|\mathbf{y}) = \sigma(W^M(W^F\mathbf{x} \odot W^F\mathbf{y}))$$

$$\mathbf{x} = r(\mathbf{x},\mathbf{y}) = \sigma(W^{F^T}(W^{M^T}\mathbf{h} \odot W^F\mathbf{y}))$$

(Memisevic 2011)

Preliminary : Gated Auto-encoders

- Unsupervised Learning
- Input X,Y -> Encode X|Y -> Decode -> reproduced X|Y

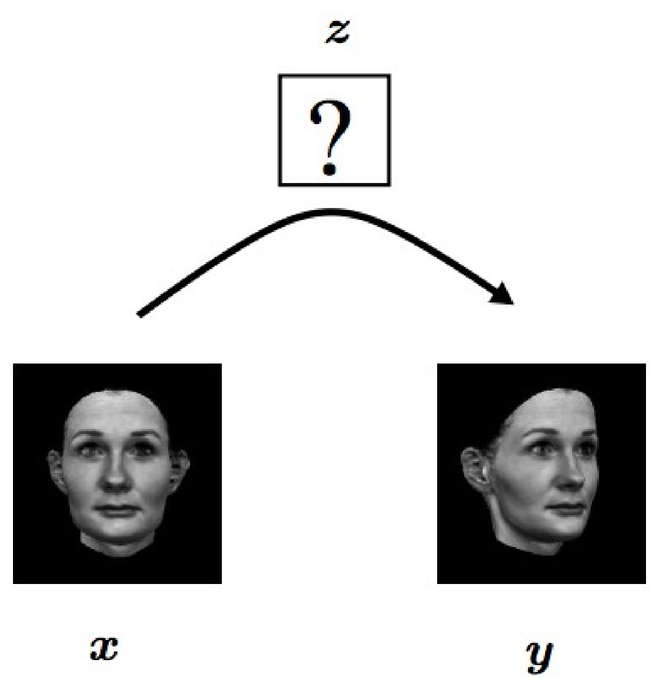


$$\min_{\theta} J(\mathbf{x}|\mathbf{y}) = \min_{\theta} \|\mathbf{x} - r(f(\mathbf{x}|\mathbf{y})|\mathbf{y})\|^2$$

(Memisevic 2011)

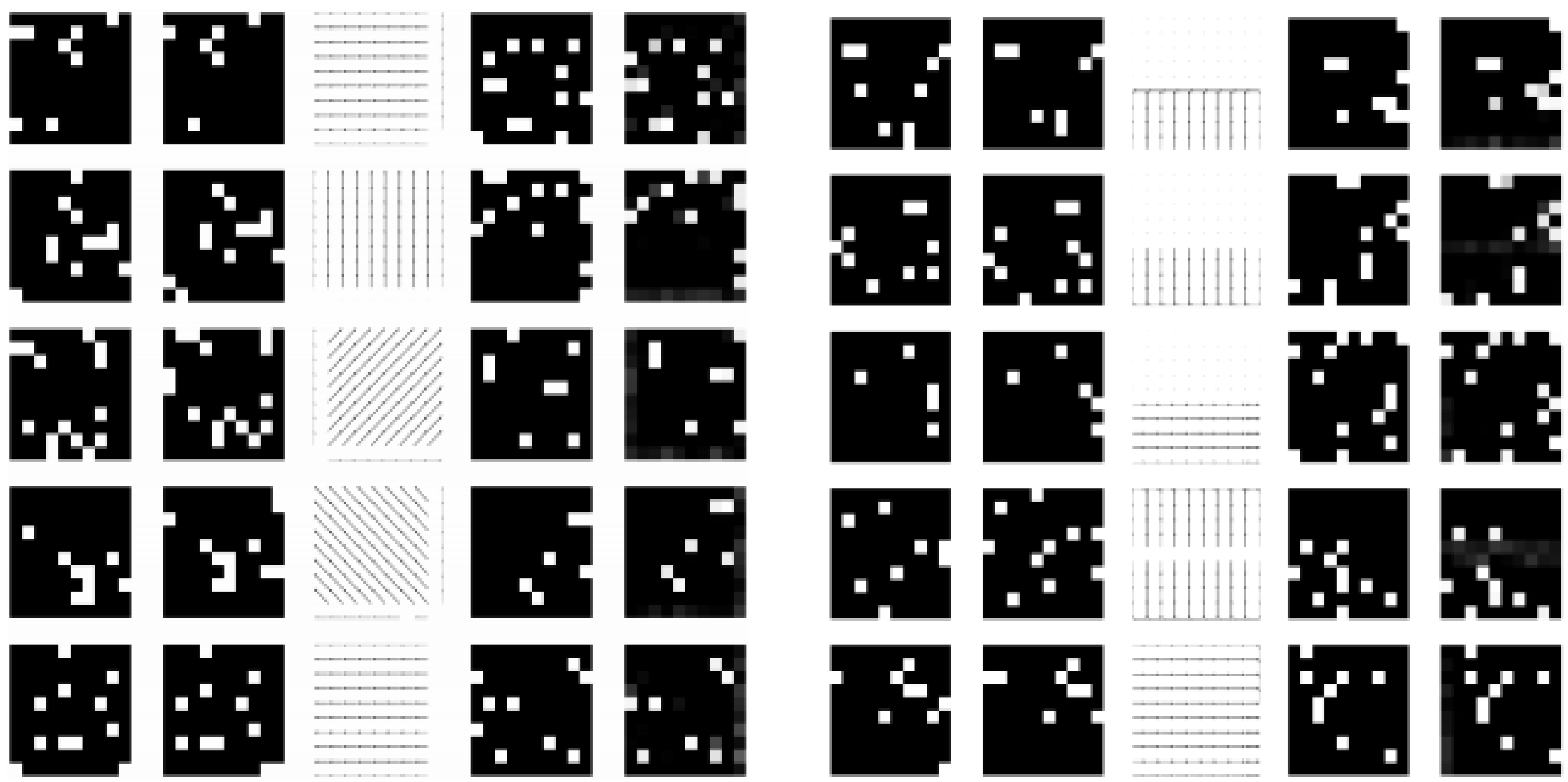
Preliminary : Gated Auto-encoders

- Unsupervised Learning
- Input X,Y -> Encode X|Y -> Decode -> reproduced X|Y
- Learns to relate X and Y.



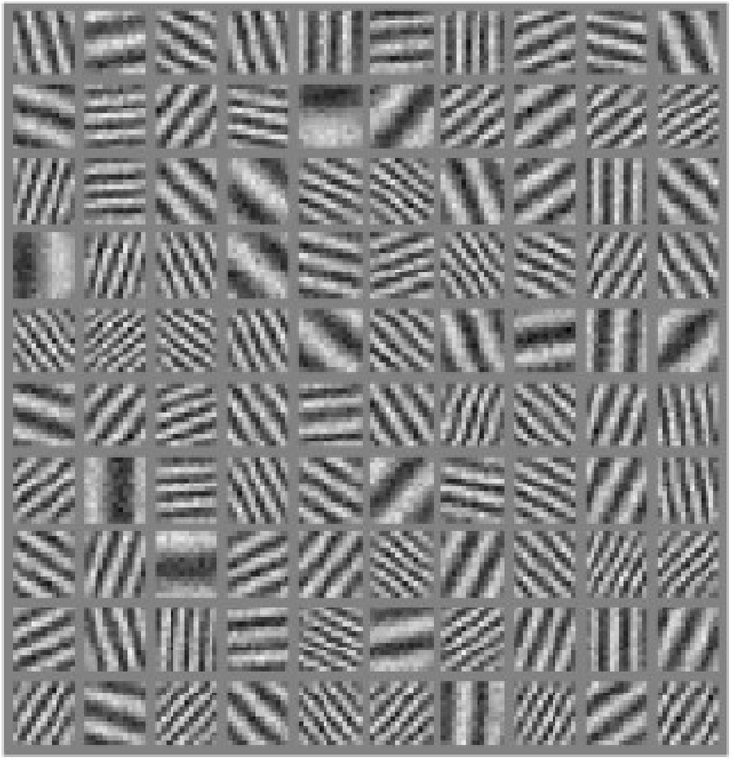
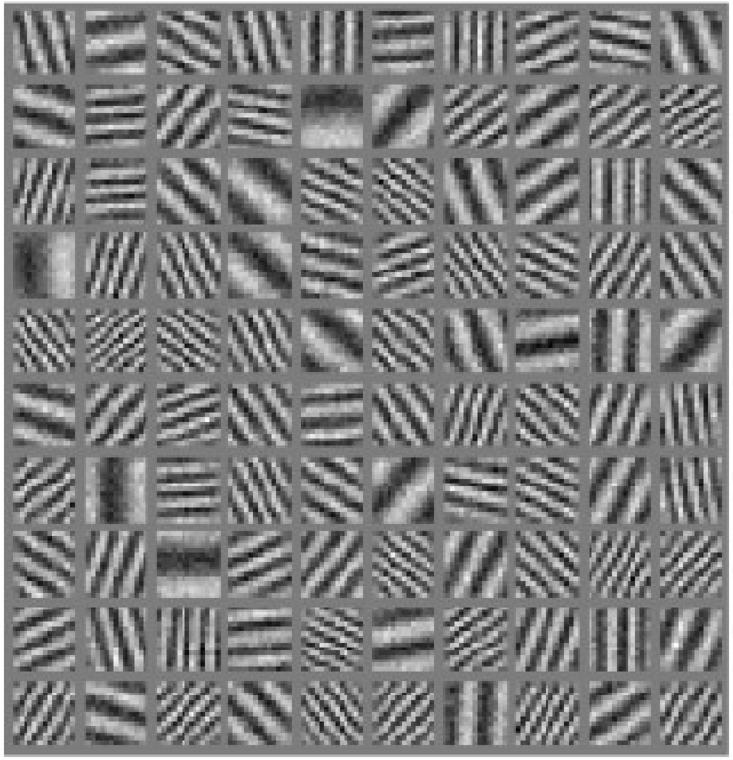
(Memisevic 2013)

Preliminary : Gated Auto-encoders

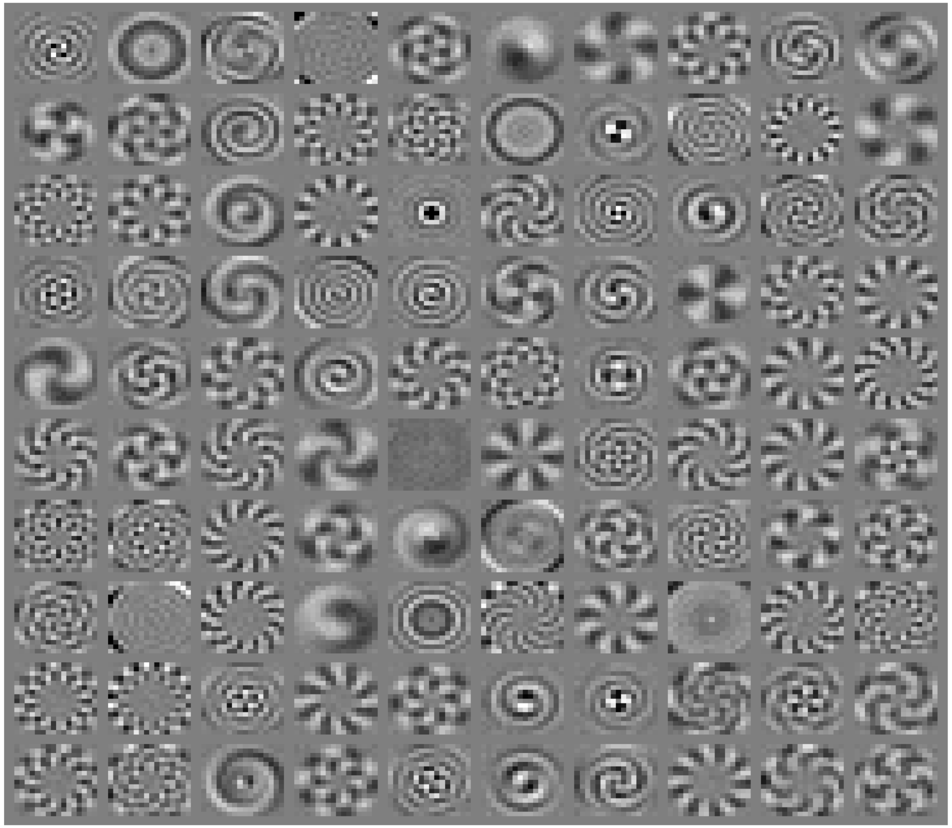


(Memisevic 2013)

Preliminary : Gated Auto-encoders

Preliminary : Gated Auto-encoders



(Memisevic 2013)

What do you have to remember from Preliminaries?

- Not much!

What do you have to remember from Preliminaries?

- Not much!
- Just ...

What do you have to remember from Preliminaries?

- Not much!
- Just ...

$$\hat{\mathbf{y}} = r(\mathbf{y}|\mathbf{x})$$

Thinking in terms of Dynamic System...

Vector Field of : $\mathbf{y} - r(\mathbf{y}|\mathbf{x})$

- Let Gated Auto-encoder to run under this dynamics

Thinking in terms of Dynamic System...

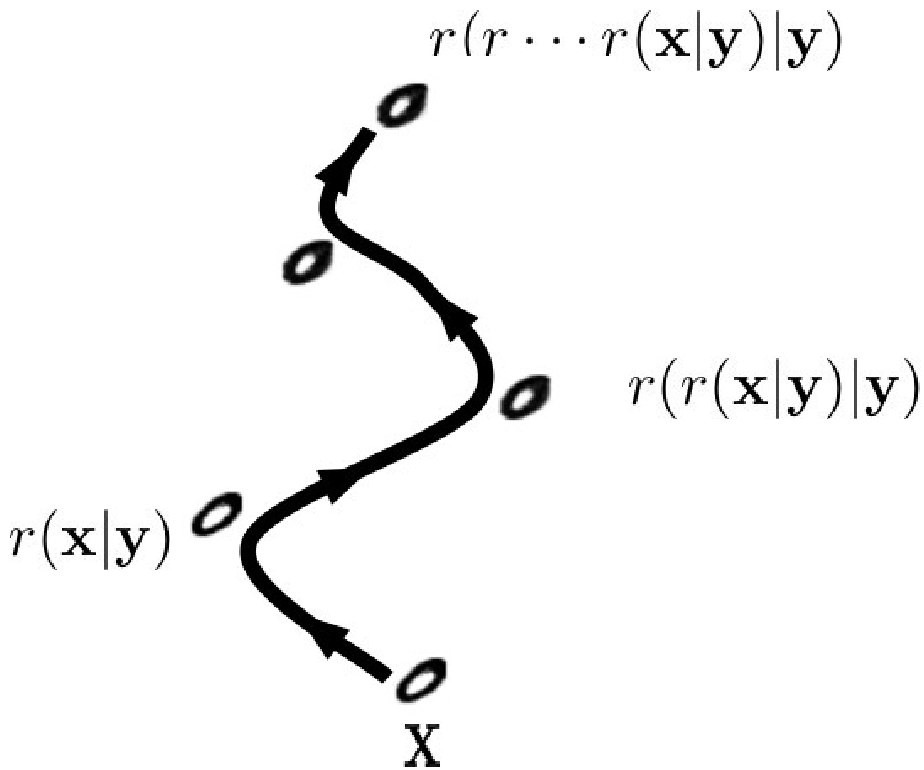
Vector Field of : $\mathbf{y} - r(\mathbf{y}|\mathbf{x})$

- Let Gated Auto-encoder to run under this dynamics
- i.e. $\mathbf{x} \rightarrow r(\mathbf{x}|\mathbf{y}) \rightarrow r(r(\mathbf{x}|\mathbf{y})|\mathbf{y}) \cdots r(\cdots r(\mathbf{x}|\mathbf{y})|\mathbf{y})$

Thinking in terms of Dynamic System...

Vector Field of : $\mathbf{y} - r(\mathbf{y}|\mathbf{x})$

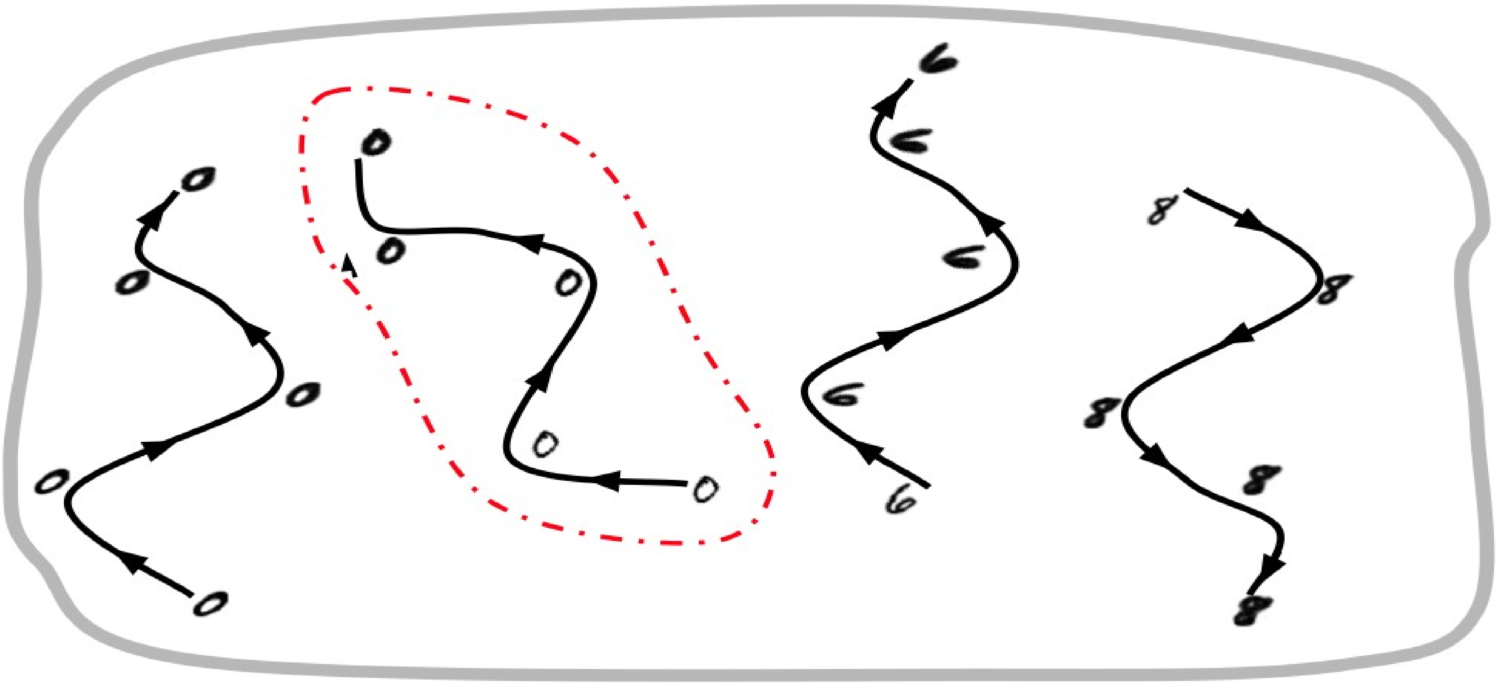
- Let Gated Auto-encoder to run under this dynamics
- i.e. $\mathbf{x} \rightarrow r(\mathbf{x}|\mathbf{y}) \rightarrow r(r(\mathbf{x}|\mathbf{y})|\mathbf{y}) \cdots r(\cdots r(\mathbf{x}|\mathbf{y})|\mathbf{y})$



Thinking in terms of Dynamic System...

Vector Field of : $\mathbf{y} - r(\mathbf{y}|\mathbf{x})$

- Let Gated Auto-encoder to run under this dynamics
- i.e. $\mathbf{x} \rightarrow r(\mathbf{x}|\mathbf{y}) \rightarrow r(r(\mathbf{x}|\mathbf{y})|\mathbf{y}) \cdots r(\cdots r(\mathbf{x}|\mathbf{y})|\mathbf{y})$



Poincare Integrability Criterion

- For some open set $\mathcal{U}$, a continuously differentiable function $F : \mathcal{U} \rightarrow \mathbb{R}^m$ defines a gradient field if and only if

$$\frac{\partial F_i(\mathbf{y})}{\partial y_j} = \frac{\partial F_j(\mathbf{y})}{\partial y_i} \qquad \forall i, j = 1 \cdots n$$

Poincare Integrability Criterion

- For some open set $\mathcal{U}$, a continuously differentiable function $F : \mathcal{U} \to \mathbb{R}^m$ defines a gradient field if and only if

$$\frac{\partial F_i(\mathbf{y})}{\partial y_j} = \frac{\partial F_j(\mathbf{y})}{\partial y_i} \qquad \forall i, j = 1 \cdots n$$

- Implication?

Poincare Integrability Criterion

- For some open set $\mathcal{U}$, a continuously differentiable function $F : \mathcal{U} \to \mathbb{R}^m$ defines a gradient field if and only if

$$\frac{\partial F_i(\mathbf{y})}{\partial y_j} = \frac{\partial F_j(\mathbf{y})}{\partial y_i} \qquad \forall i, j = 1 \cdots n$$

- Implication?
  - The vector field is conservative field

Poincare Integrability Criterion

- For some open set $\mathcal{U}$, a continuously differentiable function $F : \mathcal{U} \to \mathbb{R}^m$ defines a gradient field if and only if

$$\frac{\partial F_i(\mathbf{y})}{\partial y_j} = \frac{\partial F_j(\mathbf{y})}{\partial y_i} \qquad \forall i, j = 1 \cdots n$$

- Implication?
  - The vector field is conservative field
  - The vector field is the gradient of scaler field

$$r(\mathbf{y}|\mathbf{x}) - \mathbf{y} = \nabla E$$

Gated Auto-encoder Scoring

- How much energy did GAE took to run??

Gated Auto-encoder Scoring

- How much energy did GAE took to run??

$$E = \int \nabla E dt = \int F dt$$

Gated Auto-encoder Scoring

- How much energy did GAE took to run??

$$E = \int \mathbf{y} - r(\mathbf{y}|\mathbf{x}) d\mathbf{y}$$

Gated Auto-encoder Scoring

- How much energy did GAE took to run??

$$E = \int \mathbf{y} - r(\mathbf{y}|\mathbf{x}) d\mathbf{y}$$

$$E(\mathbf{y}|\mathbf{x}) = \int h(\mathbf{u}) d\mathbf{u} - \frac{1}{2}\mathbf{y}^2 + const$$

Relationship to Restricted Boltzmann Machines

- Consider our non-linear function h to be a sigmoid function $\sigma(\cdot) = 1/(1+\exp(-\cdot))$

$$E(\mathbf{y}|\mathbf{x}) = \int h(\mathbf{u})d\mathbf{u} - \frac{1}{2}\mathbf{y}^2 + const$$

Relationship to Restricted Boltzmann Machines

- Consider our non-linear function h to be a sigmoid function $\sigma(\cdot) = 1/(1 + \exp(-\cdot))$

$$E(\mathbf{y}|\mathbf{x}) = \int h(\mathbf{u}) d\mathbf{u} - \frac{1}{2}\mathbf{y}^2 + const$$

$$E_\sigma(\mathbf{y}|\mathbf{x}) = \sum_k \log\left(1 + \exp\left(W^M (W^F_{k\cdot}\mathbf{y} \odot W^F_{k\cdot}\mathbf{x})\right)\right) - \frac{\mathbf{y}^2}{2} + const$$

Relationship to Restricted Boltzmann Machines

- Consider our non-linear function h to be a sigmoid function $\sigma(\cdot) = 1/(1+\exp(-\cdot))$

$$E_\sigma(\mathbf{y}|\mathbf{x}) = \sum_k \log\left(1+\exp\left(W^M(W^F_{k\cdot}\mathbf{y} \odot W^F_{k\cdot}\mathbf{x})\right)\right) - \frac{\mathbf{y}^2}{2} + const$$

- This equation is same as Free Energy of Factored Gated Restricted Boltzmann Machine with ignoring bias for simplicity

Relationship to Restricted Boltzmann Machines

Theorem 1. *Consider a CAE with encoder and decoder:*

$$h(\mathbf{x}) = h(W^M((W^F\mathbf{x})^2) + \mathbf{b})$$
$$r(\mathbf{x}|h) = (W^F)^T(W^F\mathbf{x} \odot (W^M)^T h(\mathbf{x})) + \mathbf{a},$$

where $\theta = \{W^F, W^M, \mathbf{a}, \mathbf{b}\}$ are the parameters of the model, and $h(\cdot) = \frac{1}{1+\exp(-\cdot)}$ is a sigmoid function. Moreover, consider a Covariance Restricted Boltzmann Machine [12] with Gaussian-distributed visibles and Bernoulli-distributed hiddens, such that its energy function is defined by

$$E^c(\mathbf{x}, \mathbf{h}) = \frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} - \sum_f P\mathbf{h}(C\mathbf{x})^2 - \mathbf{b}\mathbf{h}.$$

Then the energy function of the CAE with dynamics $r(\mathbf{x}|\mathbf{y}) - \mathbf{x}$ is equivalent to the free energy of Covariance RBM up to a constant:

$$E(\mathbf{x}, \mathbf{x}) = \sum_k \log\left(1 + \exp\left(W^M(W^F\mathbf{x})^2 + \mathbf{b}\right)\right) - \frac{\mathbf{x}^2}{2} + \text{const} \tag{11}$$

Relationship to Restricted Boltzmann Machines

- Kamyshanska et al 2013 showed that free energy of Auto-encoder is same as Gaussian Bernoulli RBM

Relationship to Restricted Boltzmann Machines

- Kamyshanska et al 2013 showed that free energy of Auto-encoder is same as Gaussian Bernoulli RBM

Corollary 1.1. *The energy function of a Mean-covariance auto-encoder and the free energy of a Mean-covariance RBM (mcRBM) with Gaussian-distributed visibles and Bernoulli-distributed hiddens are equivalent up to a constant. The energy of the mcAE is:*

$$E = \sum_k \log\left(1 + \exp\left(-W^M (W^F \mathbf{x})^2 - \mathbf{b}\right)\right) + \sum_k \log\left(1 + \exp\left(W\mathbf{x} + \mathbf{c}\right)\right) - \mathbf{x}^2 + \text{const} \quad (12)$$

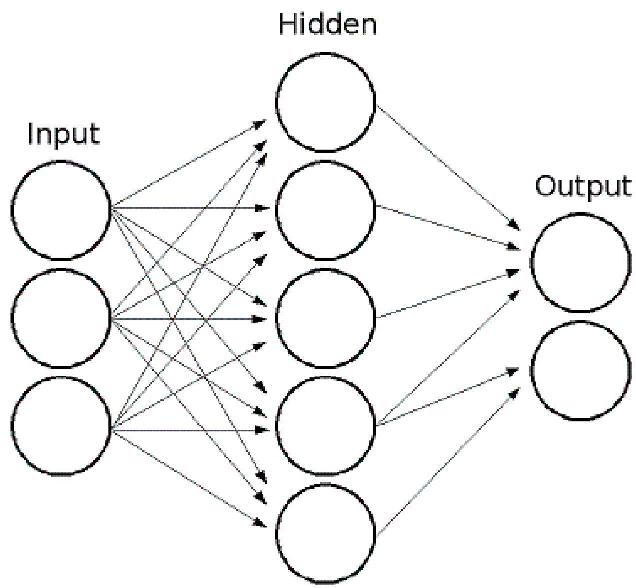
where $\theta = \{W, \mathbf{c}\}$ *parameterize the mean mapping units and* $\theta = \{W^F, W^M, \mathbf{a}, \mathbf{b}\}$ *parameterize the covariance mapping units.*

Application: Structured Prediction

- Input X & Structured Output Y
- Predict Y with Neural Network, $\tilde{\mathbf{y}} = f_{NN}(\mathbf{x})$

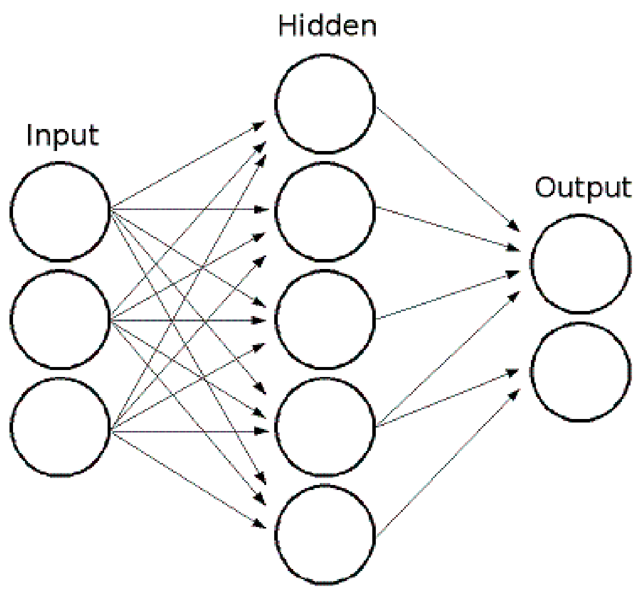
Application: Structured Prediction

- Back to structured Prediction!!
- Input X & Structured Output Y
- Predict Y with Neural Network, $\tilde{\mathbf{y}} = f_{NN}(\mathbf{x})$



Application: Structured Prediction

- Input X & Structured Output Y
- Predict Y with Neural Network, $\tilde{\mathbf{y}} = f_{NN}(\mathbf{x})$



- Problem : each component of $\tilde{\mathbf{y}}$ is independently predicted by Neural Net!!

Application: Structured Prediction

- Remember our score function?

$$E_\sigma(\mathbf{y}|\mathbf{x}) = \sum_k \log\left(1 + \exp\left(W^M(W^F_{k\cdot}\mathbf{y} \odot W^F_{k\cdot}\mathbf{x})\right)\right) - \frac{\mathbf{y}^2}{2} + const$$

- Let's minimize the energy with respect to $\tilde{\mathbf{y}}$

$$\nabla_{\tilde{\mathbf{y}}} E_\sigma(\mathbf{y}|\mathbf{x})$$

Application: Structured Prediction

- Neural Net gave good initialization of $\tilde{\mathbf{y}}$. We fine-tune the structured prediction using our score function

---

Algorithm 1 Structured Output Learning via GAE scoring

- Train a Neural Network to predict the structured output.

$$\underset{\theta}{\operatorname{argmin}}\, l(\mathbf{x}, \mathbf{y}; \theta) = \|\mathcal{NN}(\mathbf{x}_{train}; \theta) - \mathbf{y}_{train}\|^2 \tag{20}$$

- Train a Gated Auto-encoder with input $(\mathbf{x}, \mathbf{y})_{train}$.

$$\underset{\theta}{\operatorname{argmin}}\, l(\mathbf{x}, \mathbf{y}; \theta) = \|r(\mathbf{y}_{train}|\mathbf{x}_{train}, h, \theta) - \mathbf{y}_{train}\|^2 \tag{21}$$

  where $r(\mathbf{y}_{train}|\mathbf{x}_{train}, h, \theta)$ is the reconstruction function of $\mathbf{y}_{train}$
- **For** each test data point $\mathbf{x}_{\mathrm{i}} \in \mathcal{X}_{test}$ **do**
  1. Initialize the output using Neural Network.

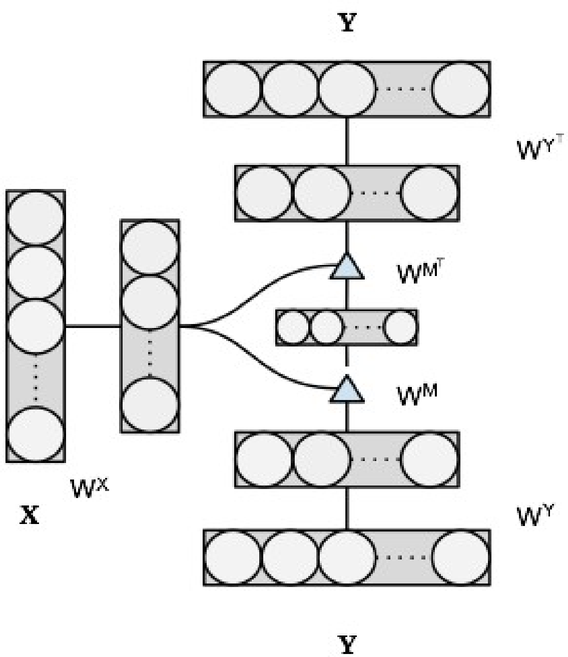
$$\hat{\mathbf{y}} = \mathcal{NN}(\mathbf{x}_{test}) \tag{22}$$

  While $\|E_{t+1}(\hat{\mathbf{y}}|\mathbf{x}) - E_t(\hat{\mathbf{y}}|\mathbf{x})\| > \epsilon$ converge **do**
  - Compute $\nabla_{\hat{\mathbf{y}}} E$
  - Update $\hat{\mathbf{y}} = \hat{\mathbf{y}} - \lambda \nabla_{\hat{\mathbf{y}}} E$

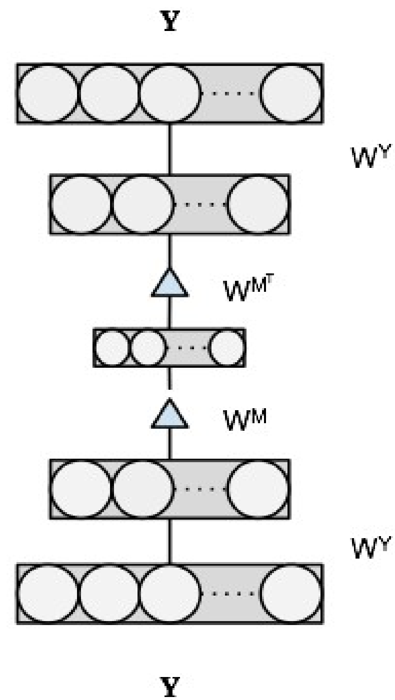
  where ϵ is the tolerance rate.

Application: Structured Prediction

- Tested on two Gated Auto-encoder architecture.



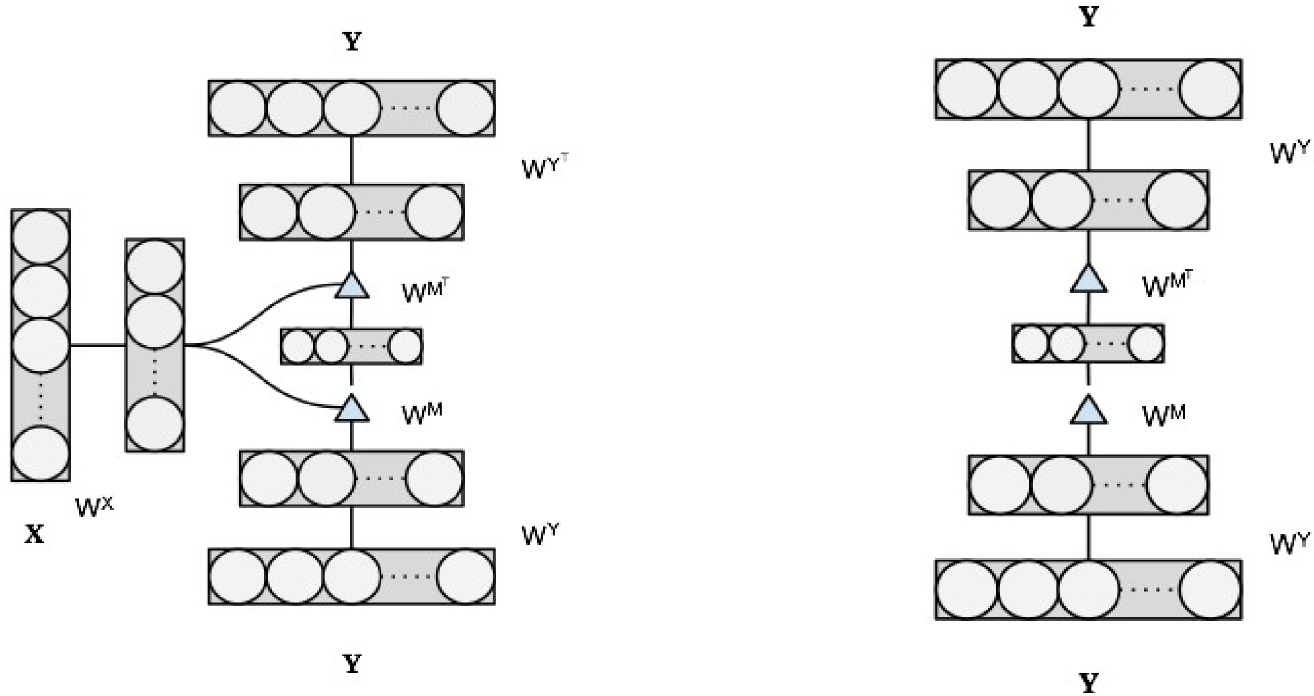
(a) Model 1



(b) Model 2

Application: Structured Prediction

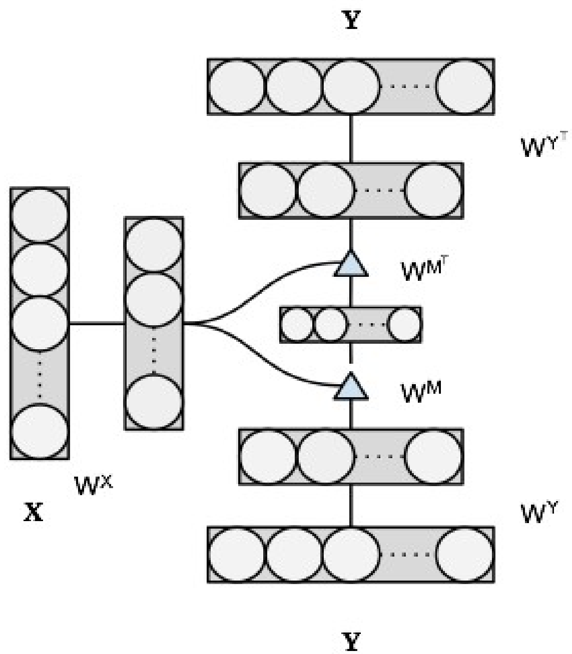
- Tested on two Gated Auto-encoder architecture.



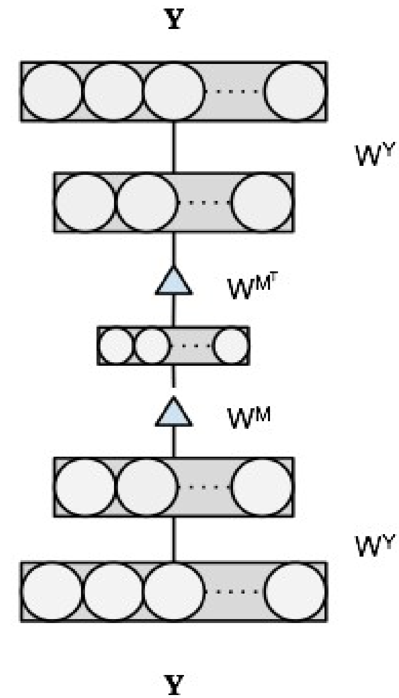
(a) Model 1 (b) Model 2

$$\tilde{\mathbf{y}} = r(\mathbf{y}|\mathbf{x}) \text{ VS. } \tilde{\mathbf{y}} = r(\mathbf{y}|\mathbf{y})$$

Application: Structured Prediction



(a) Model 1



(b) Model 2

$$\tilde{\mathbf{y}} = r(\mathbf{y}|\mathbf{x}) \text{ VS. } \tilde{\mathbf{y}} = r(\mathbf{y}|\mathbf{y})$$

- Model 1 – models relationship between **x** and **y**
- Model 2 - models correlation between **y** itself

Experiments : Classification

- Deep Learning Benchmark Dataset
  - Rectangle
  - Rectangle with background Image
  - Convex
  - MNIST
  - Rotated MNIST
  - MNIST with Random Noisy Background
  - MNIST with background Image
  - Rotated MNIST with background Image

Experiments : Classification

- Problem: Energy function is unnormalized function.

Experiments : Classification

- Problem: Energy function is unnormalized function.
- Solution: We adapt Gated softmax Classifier

(Memisevic 2011)

Experiments : Classification

- Solution: We adapt Gated softmax Classifier

(Memisevic 2011)

Algorithm:

1. Train an Auto-encoder and Gated Auto-encoder for each class

Experiments : Classification

- Solution: We adapt Gated softmax Classifier

(Memisevic 2011)

Algorithm:

1. Train an Auto-encoder and Gated Auto-encoder for each class

Experiments : Classification

- Solution: We adapt Gated softmax Classifier

(Memisevic 2011)

Algorithm:

1. Train an (denoised) mean covariance (gated) Auto-encoder for each class with tied weights and inputs.
   - We have score for each class

Experiments : Classification

- Solution: We adapt Gated softmax Classifier

(Memisevic 2011)

Algorithm:

1. Train an (denoised) mean covariance (gated) Auto-encoder for each class with tied weights and inputs.
   - We have score for each class
2. Train the mean-covariance Auto-encoder scoring coefficient

$$P_{GAE}(y_i|\mathbf{x}) = \frac{\exp(E_i^C(\mathbf{x}) + B_i)}{\sum_j \exp(E_j^C(\mathbf{x}) + B_j)}, P_{mcAE}(y_i|\mathbf{x}) = \frac{\exp(E_i^M(\mathbf{x}) + E_i^C(\mathbf{x}) + B_j)}{\sum_j \exp(E_j^M(\mathbf{x}) + E_j^C(\mathbf{x}) + B_j)} \quad (19)$$

Experiments : Classification

DATA	SVM RBF	RBM	DEEP SAA_3	GSM	AES	GAES	mcAES
RECT	2.15	4.71	2.14	0.56	0.84	0.61	**0.54**
RECTIMG	24.04	23.69	24.05	22.51	21.45	22.85	**21.41**
CONVEX	19.13	19.92	18.41	**17.08**	21.52	21.6	20.63
MNISTSMALL	3.03	3.94	3.46	3.70	**2.61**	3.65	3.65
MNISTROT	11.11	14.69	**10.30**	11.75	11.25	16.5	13.42
MNISTRAND	14.58	9.80	11.28	10.48	**9.70**	18.65	16.73
MNISTROTIM	55.18	52.21	51.93	55.16	47.14	39.98	**35.52**

Experiments: Structured Output

- Four dataset: Yeast, Scene, Mturk, Majmin
  - Yeast, Scene – image labelling task
  - Mturk, Majmin – music tagging task

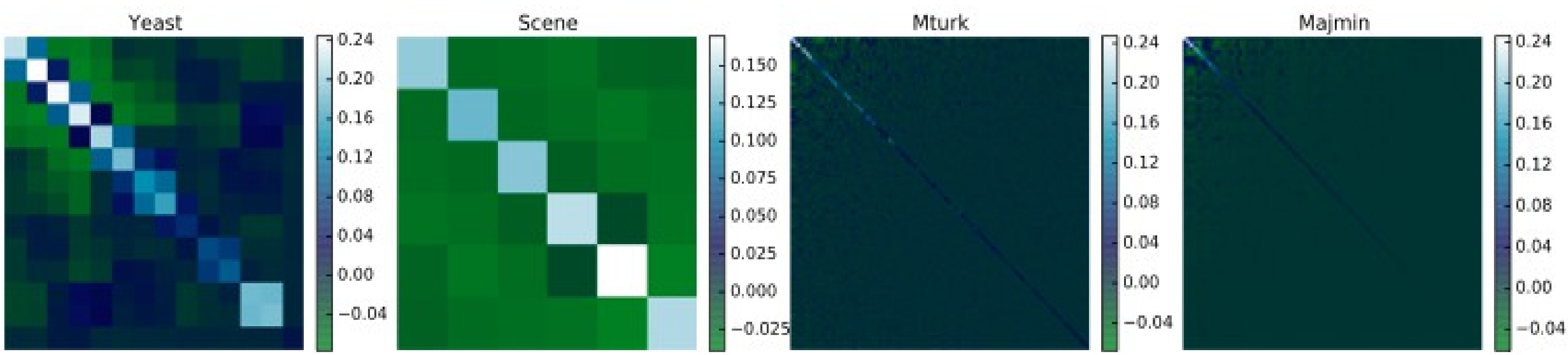
Experiments: Structured Output

- Four dataset: Yeast, Scene, Mturk, Majmin
  - Yeast, Scene – image labelling task
  - Mturk, Majmin – music tagging task

	# of dimension in **X**	# of dimension in **Y**
Yeast	103	14
Scene	294	6
Mturk	389	92
Majmin	389	96

Experiments: Structured Output

- Four dataset: Yeast, Scene, Mturk, Majmin
  - Yeast, Scene – image labelling task
  - Mturk, Majmin – music tagging task



Experiments: Structured Output

- 10 Folds
- 80 % training 10 % validation 10 % testing
- Model1 - $GAES_{XY}$
- Model2 - $GAES_{Y^2}$

Method	Yeast	Scene	MTurk	MajMin
LogReg	20.16	10.11	8.10	4.34
HashCRBM*	20.02	8.80	7.24	4.24
NeuralNet	19.79	8.99	7.13	4.23
GAES_{XY}	19.67	8.90	7.11	4.22
GAES_{Y^2}	19.76	8.90	7.13	4.22

Table 1: Error rate on multi-label datasets

Conclusion

- Showed that the GAE could be scored according to an energy function.
- Demonstrated the equivalency of the GAE energy to the free energy of RBM types of model.
- The main advantage of our suggested model is that optimization is fast by sorely taking gradient descent with respect to the Gated Auto-encoder scoring function. This leverage allows enormous eff ciency and the model’s ability.

Questions??