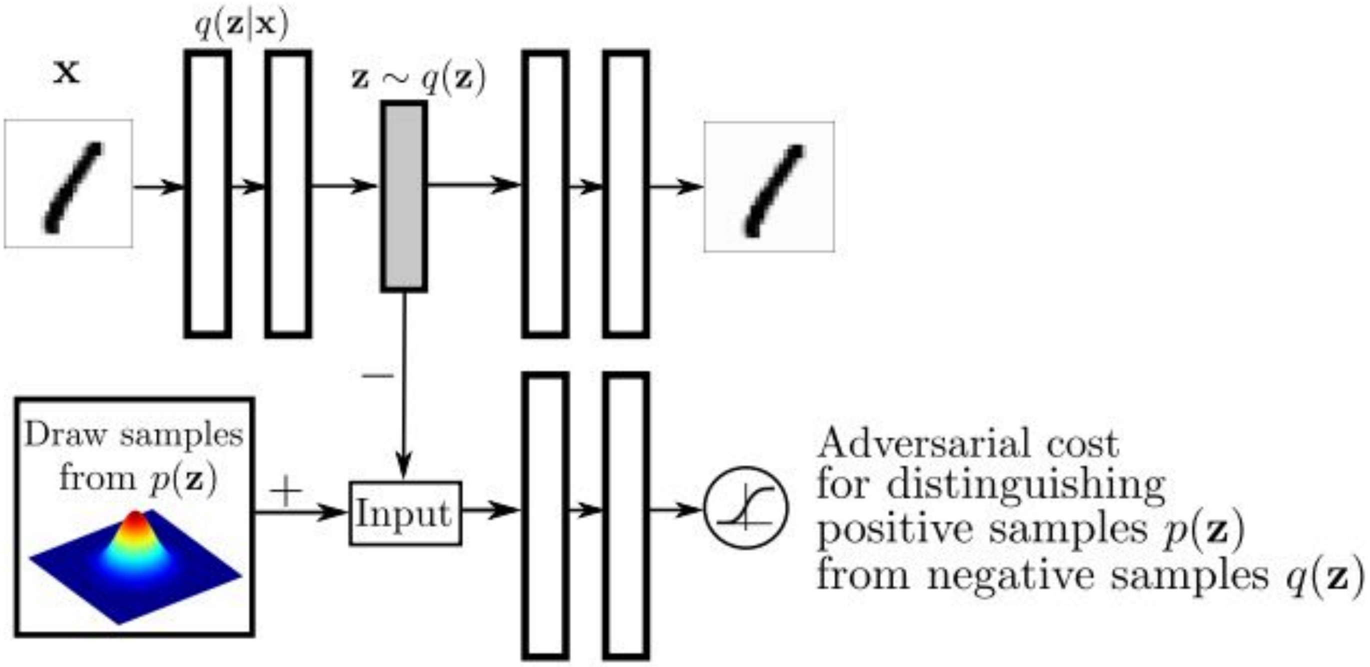
Adversarially Regularized Autoencoders

Junbo (Jake) Zhao, Yoon Kim, Kelly Zhang,
Alexander M. Rush, Yann LeCun

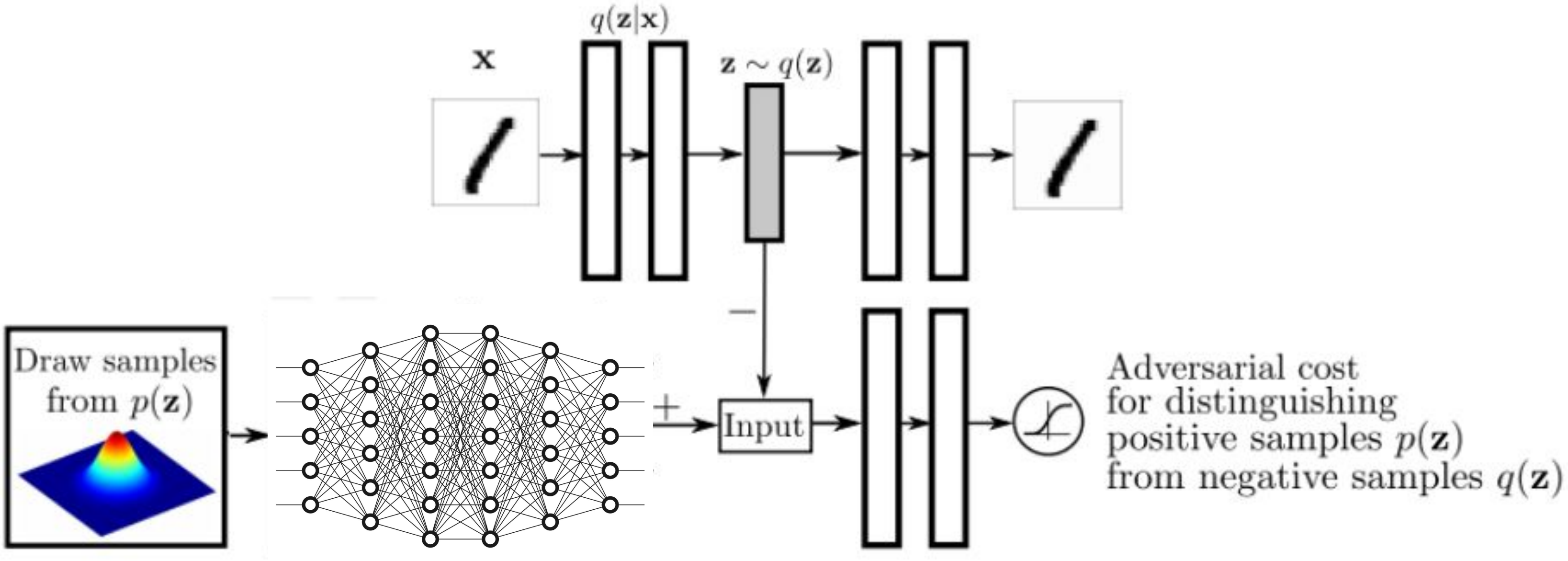
Wei Zhen Teoh and Mathieu Ravaut

Refresh: Adversarial Autoencoder



[From Adversarial Autoencoders by Makhzani et al 2015]

Some Changes - Learned Generator



Generator distribution is also learned

Some Changes - Wasserstein GAN

- The distance measure between two distributions is defined by the Earth-mover distance, or Wasserstein-1:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\|x - y\| \big] ,$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$.

[From Wasserstein GAN by Arjovsky et al 2017]

Some Changes - Wasserstein GAN

- This is equivalent to the following supremum over Lipschitz-1 functions:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

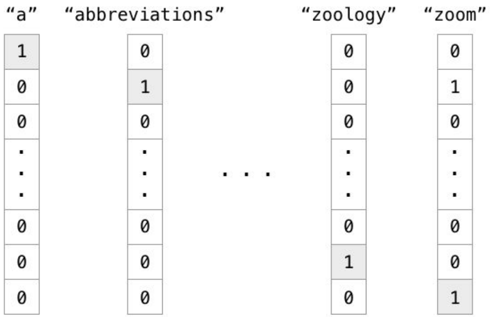
- In practice, f is approximated by a neural network f_w where all the weights are clipped to lie in a compact space such as a hypercube of size epsilon.

Some Changes - Discrete Data

Instead of a continuous vector, X is now discrete data:

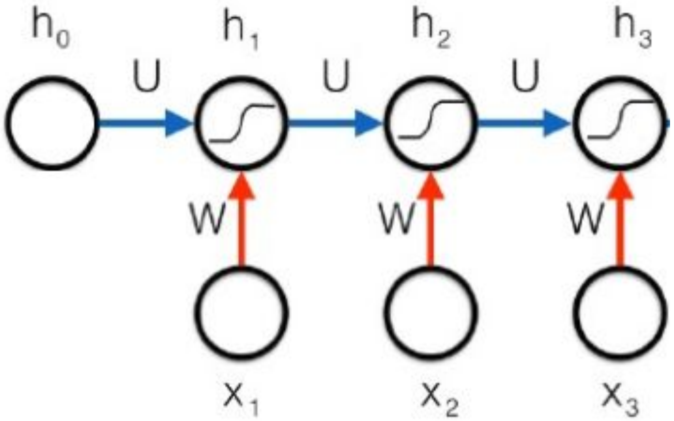
- Binarized MNIST

- Text (sequences of one-hot vocabulary vector)



[From https://ayearofai.com/lenny-2-autoencoders-and-word-embeddings-oh-my-576403b0113a]

Some Changes - Encoder (for sequential data)

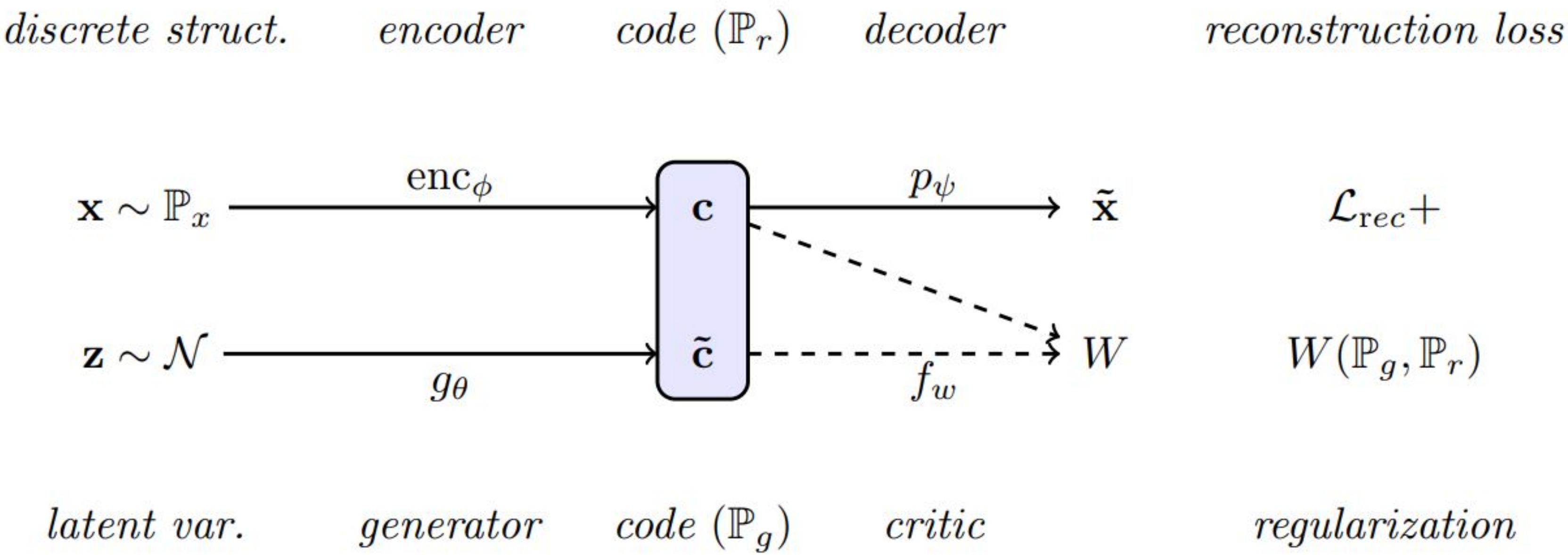


$\mathbf{c} \quad (\mathbb{P}_r)$

h_n becomes the latent code c

[From https://mlalgorithm.wordpress.com/2016/08/04/deep-learning-part-2-recurrent-neural-networks-rnn/]

Model



discrete struct.
encoder
code ($\mathbb{P}_r$)
decoder
reconstruction loss
$\mathbf{x} \sim \mathbb{P}_x$
enc_ϕ
$\mathbf{c}$
p_ψ
$\tilde{\mathbf{x}}$
$\mathcal{L}_{\mathrm{rec}} +$
$\mathbf{z} \sim \mathcal{N}$
g_θ
$\tilde{\mathbf{c}}$
f_w
W
$W(\mathbb{P}_g, \mathbb{P}_r)$
latent var.
generator
code ($\mathbb{P}_g$)
critic
regularization

Training Objective

$$\min_{\phi,\psi,\theta} \quad \mathcal{L}_{\text{rec}}(\phi,\psi) + \lambda^{(1)} W(\mathbb{P}_r, \mathbb{P}_g)$$

Reconstruction loss

Wasserstein distance between two distributions

Training Objective Components

- Reconstruction from decoder:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p_\psi(\mathbf{x} \mid \mathrm{enc}_\phi(\mathbf{x}))$$

- Reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\phi, \psi) = -\log p_\psi(\mathbf{x} \mid \mathrm{enc}_\phi(\mathbf{x}))$$

Training Objective Components

Discriminator maximizing objective:

$$\mathcal{L}_{\text{cri}}(w) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_x}\left[f_w(\text{enc}_\phi(\mathbf{x}))\right] - \mathbb{E}_{\tilde{\mathbf{c}}\sim\mathbb{P}_g}\left[f_w(\tilde{\mathbf{c}})\right]$$

The max of this function approximates the Wasserstein distance

Generator minimizing objective:

$$\mathcal{L}_{\text{encs}}(\phi,\theta) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_x}\left[f_w(\text{enc}_\phi(\mathbf{x}))\right] - \mathbb{E}_{\tilde{\mathbf{c}}\sim\mathbb{P}_g}\left[f_w(\tilde{\mathbf{c}})\right]$$

Training

Algorithm 1 ARAE Training

for number of training iterations **do**

(1) Train the autoencoder for reconstruction $[\mathcal{L}_{\text{rec}}(\phi, \psi)]$.

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_x$ and compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$.

Backpropagate reconstruction loss, $\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{i=1}^{m} \log p_\psi(\mathbf{x}^{(i)} \,|\mathbf{c}^{(i)}, [\mathbf{y}^{(i)}])$, and update.

Training

Algorithm 1 ARAE Training

for number of training iterations **do**

(1) Train the autoencoder for reconstruction $[\mathcal{L}_{\text{rec}}(\phi, \psi)]$.

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_x$ and compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$.

Backpropagate reconstruction loss, $\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{i=1}^{m} \log p_\psi(\mathbf{x}^{(i)} \, | \mathbf{c}^{(i)}, [\mathbf{y}^{(i)}])$, and update.

(2) Train the critic $[\mathcal{L}_{\text{cri}}(w)]$ (Repeat k times)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_x$ and $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim \mathcal{N}(0, \mathbf{I})$.

Compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{c}}^{(i)} = g_\theta(\mathbf{z}^{(i)})$.

Backpropagate loss $-\frac{1}{m}\sum_{i=1}^{m} f_w(\mathbf{c}^{(i)}) + \frac{1}{m}\sum_{i=1}^{m} f_w(\tilde{\mathbf{c}}^{(i)})$, update, clip the critic w to $[-\epsilon, \epsilon]^d$.

Training

Algorithm 1 ARAE Training

for number of training iterations **do**

(1) Train the autoencoder for reconstruction $[\mathcal{L}_{\text{rec}}(\phi, \psi)]$.

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_x$ and compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$.

Backpropagate reconstruction loss, $\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{i=1}^m \log p_\psi(\mathbf{x}^{(i)} \,|\mathbf{c}^{(i)}, [\mathbf{y}^{(i)}])$, and update.

(2) Train the critic $[\mathcal{L}_{\text{cri}}(w)]$ (Repeat k times)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_x$ and $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$.

Compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{c}}^{(i)} = g_\theta(\mathbf{z}^{(i)})$.

Backpropagate loss $-\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{c}^{(i)}) + \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{c}}^{(i)})$, update, clip the critic w to $[-\epsilon, \epsilon]^d$.

(3) Train the generator and encoder adversarially to critic $[\mathcal{L}_{\text{encs}}(\phi, \theta)]$

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_x$ and $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$

Compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{c}}^{(i)} = g_\theta(\mathbf{z}^{(i)})$.

Backpropagate adversarial loss $\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{c}^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{c}}^{(i)})$ and update.

Extension: Code Space Transfer

Unaligned transfer for text:

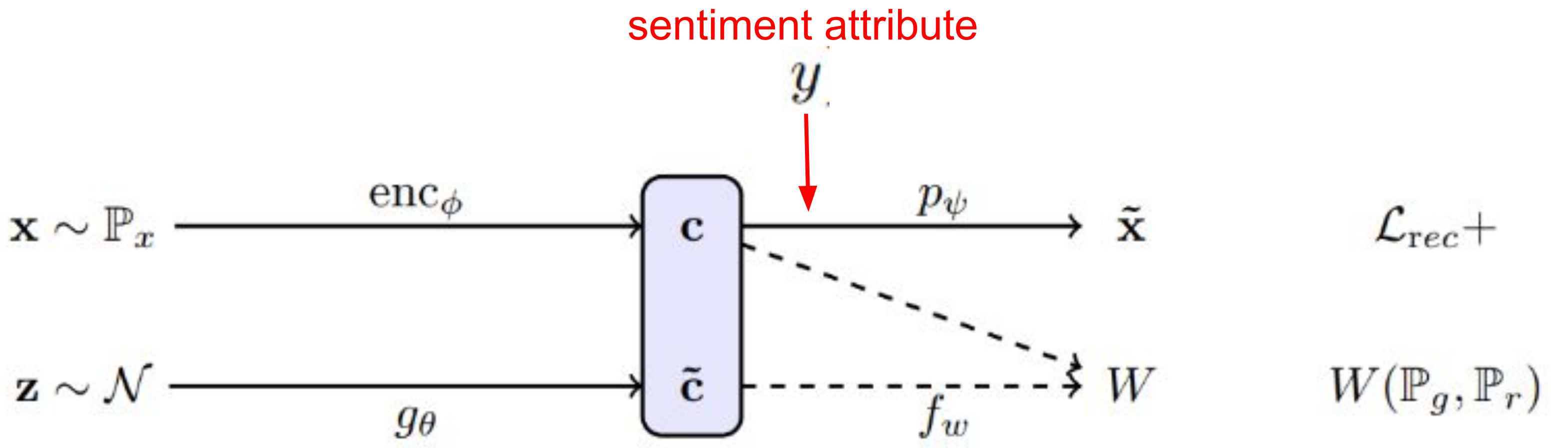
Can we change an attribute (e.g. sentiment) of the text without changing the content using this autoencoder?

Example:

Original	it has a great atmosphere , with wonderful service .
ARAE	it has no taste , with a complete jerk .

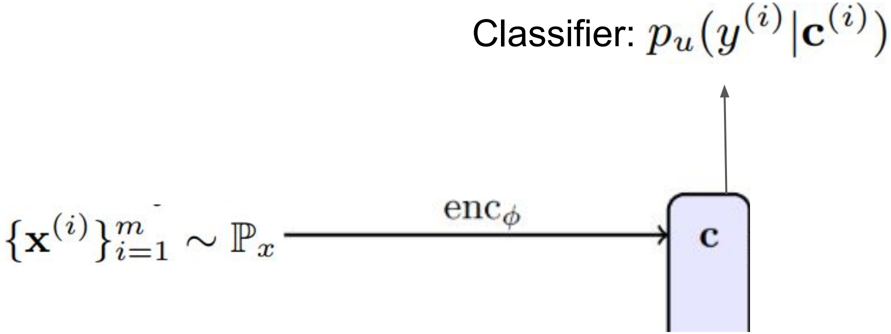
Extension: Code Space Transfer

- Extend decoder to condition on a transfer variable y to learn $p_\psi(\mathbf{x} \mid \mathbf{c}, y)$



Extension: Code Space Transfer

- Train the encoder adversarially against a classifier so that the code space is invariant to attribute y.



Additional Training

Algorithm 2 ARAE Transfer Extension

[Each loop additionally:]

(2b) Train the code classifier $[\min_u \mathcal{L}_{\text{class}}(\phi, u)]$

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_x$, lookup $y^{(i)}$, and compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$.

Backpropagate loss $-\frac{1}{m}\sum_{i=1}^m \log p_u(y^{(i)}|\mathbf{c}^{(i)})$, update.

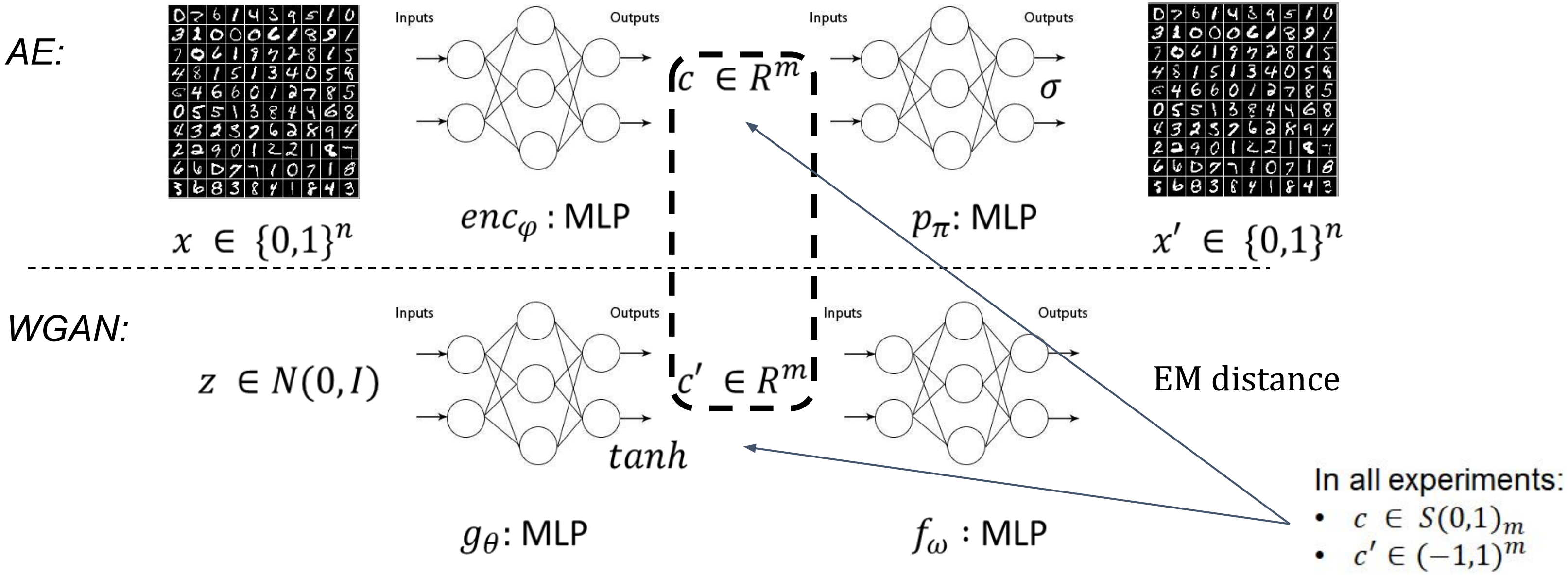
(3b) Train the encoder adversarially to code classifier $[\max_\phi \mathcal{L}_{\text{class}}(\phi, u)]$

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_x$, lookup $y^{(i)}$, and compute code-vectors $\mathbf{c}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$.

Backpropagate adversarial classifier loss $-\frac{1}{m}\sum_{i=1}^m \log p_u(1 - y^{(i)} \mid \mathbf{c}^{(i)})$, update.

Image model

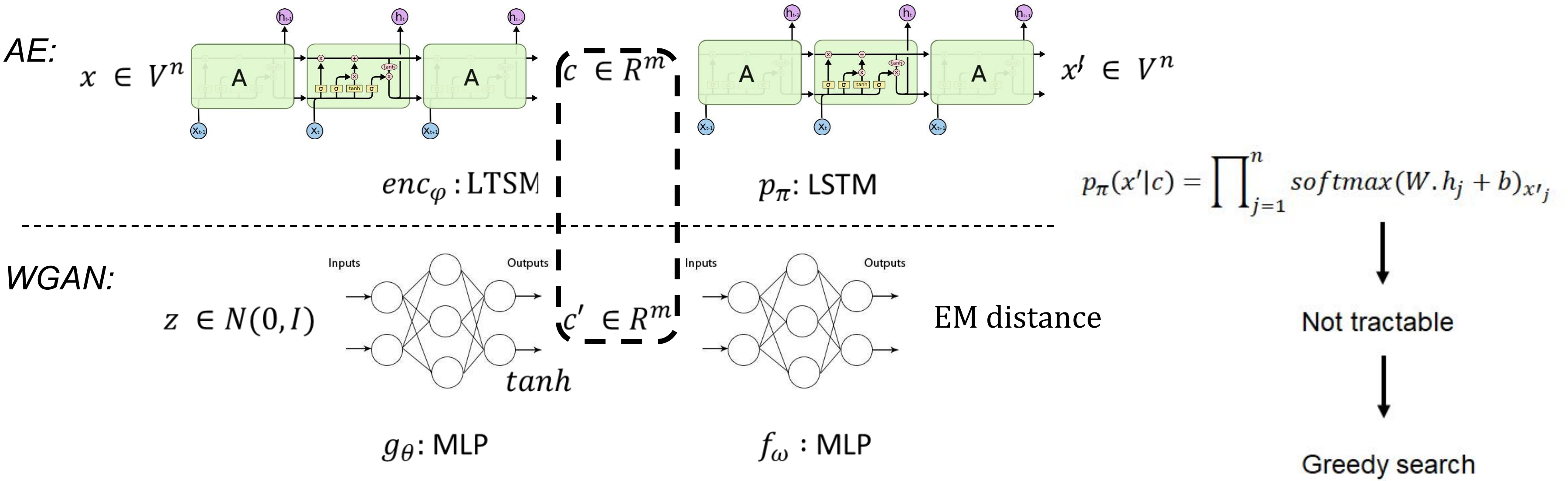
[From Adversarially Regularized Autoencoders by Zhao et al, 2017]



Input images are **binarized MNIST**, but normal MNIST would work as well.

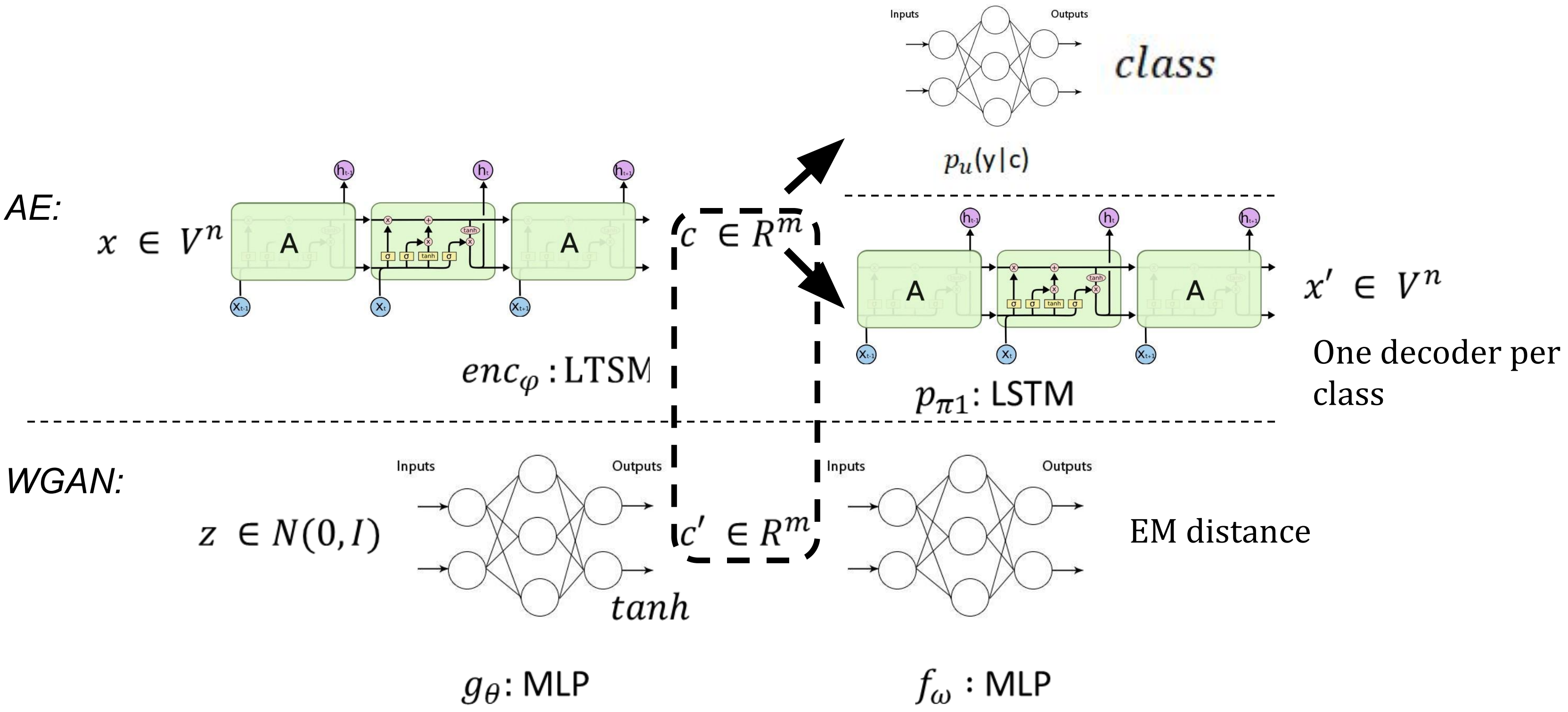
Text model

[Partly from https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f]



Same generator architecture

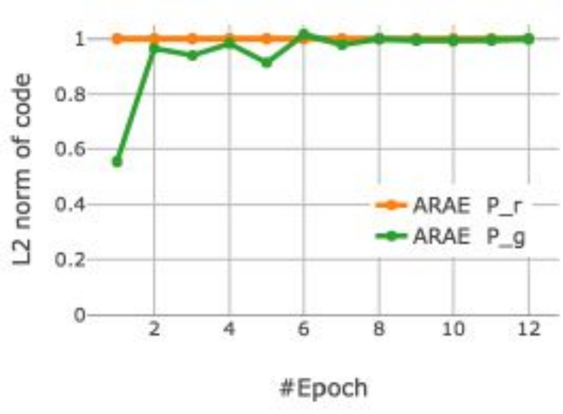
Text transfer model



Same generator architecture

Experiment #1: effects of regularizing with WGAN

Checkpoint 1:
How does the norm of c’ behave over training?

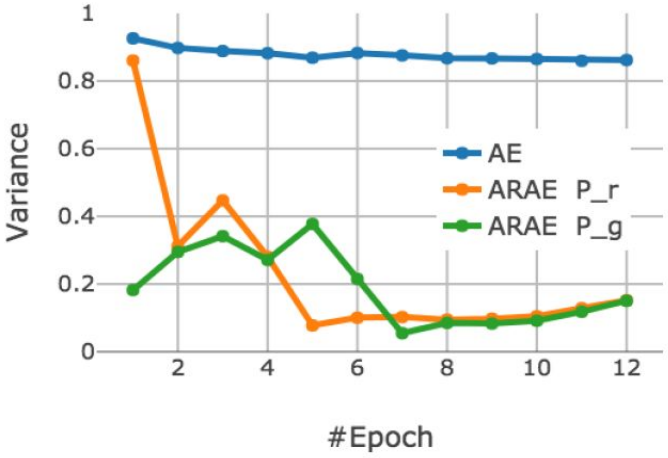


[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

c’ **L2 norm** matching c L2 norm

Experiment #1: effects of regularizing with WGAN

Checkpoint 2:
How does the encoding space behave? Is it noisy?

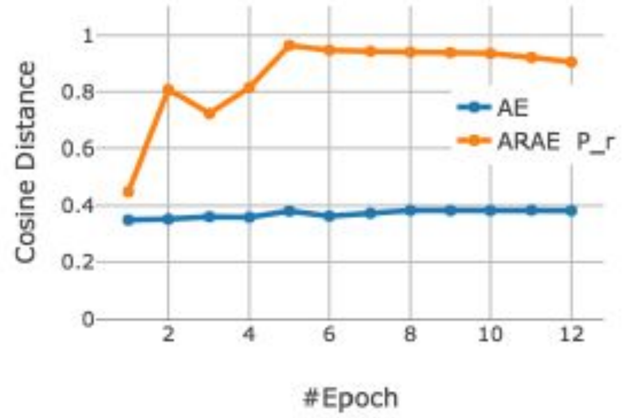


[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

c' and c sum of **dimension-wise variance** matching over time

Experiment #1: effects of regularizing with WGAN

Checkpoint 3:
Choose one sentence, then 100 other sentences within an edit-distance inferior to 5



[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

Average **cosine similarity** in latent space.
Maps similar input to nearby code.

Experiment #1: effects of regularizing with WGAN

Checkpoint 4:
Swap k words from an original sentence.

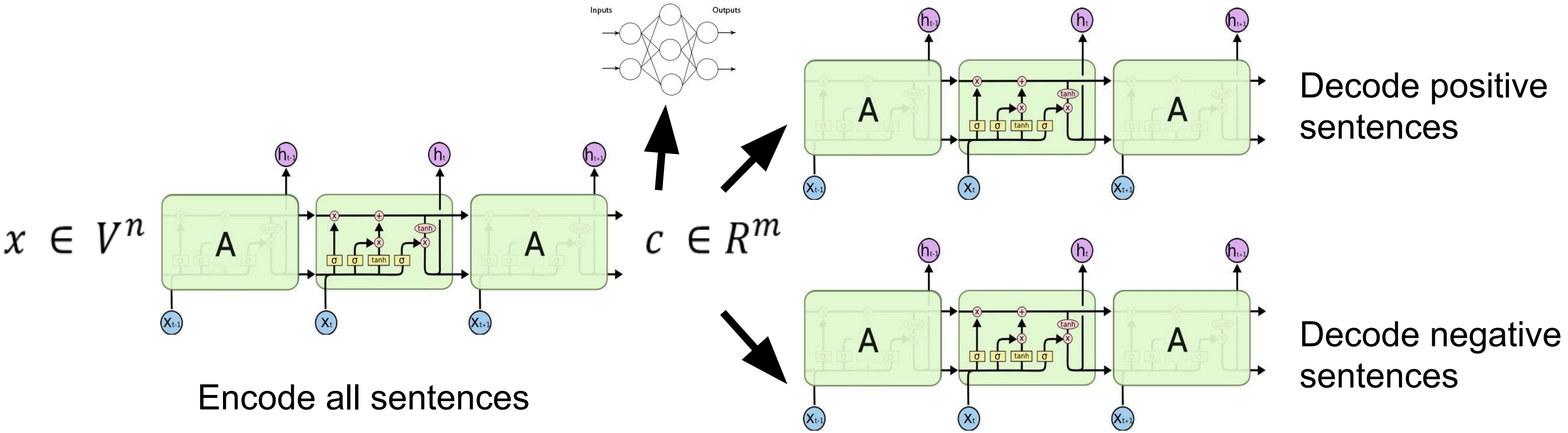
k	AE	ARAE
0	1.06	2.19
1	4.51	4.07
2	6.61	5.39
3	9.14	6.86
4	9.97	7.47

Original	A woman wearing sunglasses .	Original	They have been swimming .
Noised	A woman sunglasses wearing .	Noised	been have They swimming .
AE	A woman sunglasses wearing sunglasses .	AE	been have been swimming .
ARAE	A woman wearing sunglasses .	ARAE	Children have been swimming .
Original	Pets galloping down the street .	Original	The child is sleeping .
Noised	Pets down the galloping street .	Noised	child The is sleeping .
AE	Pets riding the down galloping .	AE	The child is sleeping is .
ARAE	Pets congregate down the street near a ravine .	ARAE	The child is sleeping .

[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

Left: reconstruction error (NLL). *Right*: reconstruction examples.

Experiment #2: unaligned text transfer



[Partly from https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f]

Remove sentiment information from the latent space:

- At training time: adversarial training.
- At test time: pass sentences of one class, decode with the decoder from the other class

Experiment #2: unaligned text transfer

Results:

Model	Automatic Evaluation Transfer	BLEU	PPL	Reverse PPL	Human Evaluation Transfer	Similarity	Naturalness
Cross-Aligned AE	77.1%	17.75	65.9	124.2	57%	3.8	2.7
AE	59.3%	37.28	31.9	68.9	-	-	-
ARAE, $\lambda_a^{(1)}$	73.4%	31.15	29.7	70.1	-	-	-
ARAE, $\lambda_b^{(1)}$	81.8%	20.18	27.7	77.0	74%	3.7	3.8

	Positive ⇒ Negative		Negative ⇒ Positive
	great indoor mall .		hell no !
ARAE	no smoking mall .	ARAE	hell great !
Cross-AE	terrible outdoor urine .	Cross-AE	incredible pork !

- Better transfer
- Better perplexity
- Transferred text less similar to original text

[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

Experiment #3: semi-supervised classification

SNLI dataset:

- 570k human-written English sentence pairs
- 3 classes: entailment, contradiction, neutral

Model	Medium	Small	Tiny
Supervised Encoder	65.9%	62.5%	57.9%
Semi-Supervised AE	68.5%	64.6%	59.9%
Semi-Supervised ARAE	70.9%	66.8%	62.5%

Medium: 22.% of labels
Small: 10.8% of labels
Tiny: 5.25% of labels

[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

Playground: latent space interpolation

Idea:

GAN input: $z \in N(0, I)$ → **Sample 2 elements:** $z_1, z_2 \sim p(z)$ → **For $\alpha \in [0,1]$:** $z_\alpha = \alpha . z_1 + (1 - \alpha) z_2$ → **Feed to the generator:** $z_\alpha \rightarrow$ G $\rightarrow$ D $\rightarrow$?

Results:

A man is on the corner in a sport area .
A man is on corner in a road all .
A lady is on outside a racetrack .
A lady is outside on a racetrack .
A lot of people is outdoors in an urban setting .
A lot of people is outdoors in an urban setting .
A lot of people is outdoors in an urban setting .

A man is on a ship path with the woman .
A man is on a ship path with the woman .
A man is passing on a bridge with the girl .
A man is passing on a bridge with the girl .
A man is passing on a bridge with the girl .
A man is passing on a bridge with the dogs .
A man is passing on a bridge with the dogs .

A man in a cave is used an escalator .
A man in a cave is used an escalator
A man in a cave is used chairs .
A man in a number is used many equipment
A man in a number is posing so on a big rock .
People are posing in a rural area .
People are posing in a rural area.

z_1

z_α

z_2

[From Adversarially Regularized Autoencoders by Zhao et al, 2017]

Conclusion about Adversarially Regularized AEs

Pros:

- ✓ Better discrete autoencoder
  - Semi-supervision
  - Text transfer
- ✓ Different approach to text generation
- ✓ Robust latent space

Cons:

- ❖ Sensitive to hyperparameters (GANs…)
- ❖ Unclear why **W**GAN
- ❖ Not so much novelty compared to Adversarial Auto Encoders (AAE)
- ❖ Discrete data but no discrete latent structure :/