Adversarially Regularized Autoencoders

Jake Zhao* 1,3  Yoon Kim* 2  Kelly Zhang 1  Alexander Rush 2
Yann LeCun 1,3

1NYU CILVR Lab
2Harvard NLP
3Facebook AI Research
Training Deep Latent Variable Models

Two dominant approaches

- **Variational inference**: bound $\log p_{\theta}(x)$ with the evidence lower bound (ELBO) and find a variational distribution that approximates the posterior $\Rightarrow$ Variational Autoencoders (VAE)

- **Implicit density methods**: Avoid dealing with the likelihood directly and learn a discriminator that distinguishes between real/fake samples $\Rightarrow$ Generative Adversarial Networks (GAN)
Training Deep Latent Variable Models

Two dominant approaches

- **Variational inference**: bound $\log p_\theta(x)$ with the evidence lower bound (ELBO) and find a variational distribution that approximates the posterior $\Rightarrow$ Variational Autoencoders (VAE)

- **Implicit density methods**: Avoid dealing with the likelihood directly and learn a discriminator that distinguishes between real/fake samples $\Rightarrow$ Generative Adversarial Networks (GAN)
Training Deep Latent Variable Models

Two dominant approaches

- **Variational inference**: bound $\log p_\theta(x)$ with the evidence lower bound (ELBO) and find a variational distribution that approximates the posterior $\Rightarrow$ **Variational Autoencoders (VAE)**

- **Implicit density methods**: Avoid dealing with the likelihood directly and learn a discriminator that distinguishes between real/fake samples $\Rightarrow$ **Generative Adversarial Networks (GAN)**
Training GANs for natural language is hard because the loss is not differentiable with respect to the generator
GAN: Problem

Possible solutions

- Use policy gradient techniques from reinforcement learning (Yu et al. 2017, Lin et al. 2017)
  - unbiased but high variance gradients
  - need to pre-train with MLE
- Consider a “soft” approximation to the discrete space (Rajeswar et al. 2017, Shen et al. 2017):
  - e.g. with the Gumbel-Softmax distribution (Maddison et al. 2017, Jang et al. 2017)
  - hard to scale to longer sentences/larger vocabulary sizes
GAN: Problem

Possible solutions

- Use policy gradient techniques from reinforcement learning (Yu et al. 2017, Lin et al. 2017)
  - unbiased but high variance gradients
  - need to pre-train with MLE
- Consider a “soft” approximation to the discrete space (Rajeswar et al. 2017, Shen et al. 2017):
  - e.g. with the Gumbel-Softmax distribution (Maddison et al. 2017, Jang et al. 2017)
  - hard to scale to longer sentences/larger vocabulary sizes
Our Work

Adversarially Regularized Autoencoders (ARAE)

- Learns an autoencoder that encodes discrete input into a continuous space and decode from it.
- Adversarial training in the continuous space at the same time
Our Work

Adversarially Regularized Autoencoders (ARAE)

- Learns an autoencoder that encodes discrete input into a continuous space and decode from it.
- Adversarial training in the continuous space at the same time
Adversarially Regularized Autoencoders

\[
\begin{align*}
\text{discrete} & \quad \text{encoder} & \quad \text{real} \ (\mathbb{P}_Q) & \quad \text{decoder} & \quad \text{reconstruction} \\
\mathbf{x} \sim \mathbb{P}_x & \quad \xrightarrow{\text{enc}_\phi} & \quad \mathbf{z} & \quad \xrightarrow{p_\psi} & \quad \mathbf{\hat{x}} & \quad \mathcal{L}_{\text{rec}} + \\
\mathbf{s} \sim \mathcal{N} & \quad \xrightarrow{g_\theta} & \quad \tilde{\mathbf{z}} & \quad \xrightarrow{f_w} & \quad \mathbf{W} & \quad W(\mathbb{P}_Q, \mathbb{P}_z) \\
\text{sample} & \quad \text{generator} & \quad \text{prior} \ (\mathbb{P}_z) & \quad \text{critic} & \quad \text{regularization}
\end{align*}
\]
In Corollary 1, we proved the equivalency of training ARAE and a latent variable model using the prior distribution, in the discrete case.

Text generation

Latent space manipulation: interpolation / vector arithmetic
Adversarially Regularized Autoencoders

\[ x \sim P_* \xrightarrow{\text{enc}_\phi} z \xrightarrow{p_\psi} \hat{x} \]

\[ s \sim \mathcal{N} \xrightarrow{g_\theta} \tilde{z} \xrightarrow{f_w} W \]

\[ L_{\text{rec}} + W(P_Q, P_z) \]

- Semi-supervised learning
- Unaligned style transfer
New metric: **Reverse perplexity**, w/ normally used **Forward perplexity**

- Generate **synthetic** training data from generative model
- Train a RNN language model on generated data
- Evaluate perplexity, $PPL = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log p(x^{(i)})\right)$ on real data
- Captures mode-collapse (vs regular PPL)
- Baselines
  - Autoregressive model: RNN language model
  - Autoencoder without adversarial regularization
  - Adversarial Autoencoders with no standalone generator (mode-collapse, Reverse PPL 980)
  - Unable to train VAEs on this dataset
Adversarially Regularized Autoencoders: experiments

New metric: **Reverse perplexity**, w/ normally used **Forward perplexity**

- Generate **synthetic** training data from generative model
- Train a RNN language model on generated data
- Evaluate perplexity, \( PPL = \exp\left( -\frac{1}{N} \sum_{i=1}^{N} \log p(x^{(i)}) \right) \) on real data
- Captures mode-collapse (vs regular PPL)

**Baselines**

- Autoregressive model: RNN language model
- Autoencoder without adversarial regularization
- Aversarial Autoencoders with no standalone generator (mode-collapse, Reverse PPL 980)
- Unable to train VAEs on this dataset
Adversarially Regularized Autoencoders: experiments

New metric: **Reverse perplexity**, w/ normally used **Forward perplexity**

- Generate **synthetic** training data from generative model
- Train a RNN language model on generated data
- Evaluate perplexity, \( PPL = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log p(x^{(i)})\right) \) on **real** data
- Captures mode-collapse (vs regular PPL)

**Baselines**

- Autoregressive model: RNN language model
- Autoencoder without adversarial regularization
- Adversarial Autoencoders with no standalone generator (mode-collapse, Reverse PPL 980)
- Unable to train VAEs on this dataset
Adversarially Regularized Autoencoders: experiments

New metric: **Reverse perplexity**, w/ normally used **Forward perplexity**

- Generate **synthetic** training data from generative model
- Train a RNN language model on generated data
- Evaluate perplexity, \( PPL = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log p(x^{(i)})\right) \) on **real** data
- Captures mode-collapse (vs regular PPL)

**Baselines**
- Autoregressive model: RNN language model
- Autoencoder without adversarial regularization
- Aversarial Autoencoders with no standalone generator (mode-collapse, Reverse PPL 980)
- Unable to train VAEs on this dataset
New metric: **Reverse perplexity**, w/ normally used **Forward perplexity**

- Generate **synthetic** training data from generative model
- Train a RNN language model on generated data
- Evaluate perplexity, \( PPL = \exp \left( -\frac{1}{N} \sum_{i=1}^{N} \log p(x^{(i)}) \right) \) on **real** data
- Captures mode-collapse (vs regular PPL)
- Baselines
  - Autoregressive model: RNN language model
  - Autoencoder without adversarial regularization
  - Aversarial Autoencoders with no standalone generator (mode-collapse, Reverse PPL 980)
  - Unable to train VAEs on this dataset
### Adversarially Regularized Autoencoders

<table>
<thead>
<tr>
<th>Data for Training LM</th>
<th>Reverse PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real data</td>
<td>27.4</td>
</tr>
<tr>
<td>Language Model samples</td>
<td>90.6</td>
</tr>
<tr>
<td>Autoencoder samples</td>
<td>97.3</td>
</tr>
<tr>
<td>ARAE samples</td>
<td>82.2</td>
</tr>
</tbody>
</table>

(Lower perplexity means higher likelihood)
**ARAE: Unaligned Style Transfer**

**Transfer Sentiment**

- Train a classifier on top of the code space:
  
  $\text{classifier}(c) = \text{probability} \ c \ \text{is a positive sentiment sentence}$

- The encoder is trained to **fool** the classifier

- To transfer sentiment:
  - Encode sentence to get code $c$
  - Switch the sentiment label, concatenate with $c$
  - Generate using the concatenated vector
Transfer Sentiment

- Train a classifier on top of the code space:
  \[ \text{classifier}(c) = \text{probability } c \text{ is a positive sentiment sentence} \]
- The encoder is trained to **fool** the classifier
- To transfer sentiment:
  - Encode sentence to get code \( c \)
  - Switch the sentiment label, concatenate with \( c \)
  - Generate using the concatenated vector
ARAE: Unaligned Style Transfer

Transfer Sentiment

- Train a classifier on top of the code space:
  \[ \text{classifier}(c) = \text{probability } c \text{ is a positive sentiment sentence} \]
- The encoder is trained to **fool** the classifier
- To transfer sentiment:
  - Encode sentence to get code \( c \)
  - Switch the sentiment label, concatenate with \( c \)
  - Generate using the concatenated vector
### ARAE: Unaligned Style Transfer

### Cross-AE: State-of-the-art model from Shen et al. 2017

<table>
<thead>
<tr>
<th>Positive ⇒ Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
</tr>
<tr>
<td>ARAE</td>
</tr>
<tr>
<td>Cross-AE</td>
</tr>
<tr>
<td>Original</td>
</tr>
<tr>
<td>ARAE</td>
</tr>
<tr>
<td>Cross-AE</td>
</tr>
<tr>
<td>Original</td>
</tr>
<tr>
<td>ARAE</td>
</tr>
<tr>
<td>Cross-AE</td>
</tr>
</tbody>
</table>
### ARAE: Unaligned Style Transfer

Cross-AE: State-of-the-art model from Shen et al. 2017

<table>
<thead>
<tr>
<th></th>
<th>Negative ⇒ Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>hell no !</td>
</tr>
<tr>
<td>ARAE</td>
<td>hell great !</td>
</tr>
<tr>
<td>Cross-AE</td>
<td>incredible pork !</td>
</tr>
<tr>
<td>Original</td>
<td>small , smokey , dark and rude management .</td>
</tr>
<tr>
<td>ARAE</td>
<td>small , intimate , and cozy friendly staff .</td>
</tr>
<tr>
<td>Cross-AE</td>
<td>great , , , chips and wine .</td>
</tr>
<tr>
<td>Original</td>
<td>the people who ordered off the menu did n’t seem to do much better .</td>
</tr>
<tr>
<td>ARAE</td>
<td>the people who work there are super friendly and the menu is good .</td>
</tr>
<tr>
<td>Cross-AE</td>
<td>the place , one of the office is always worth you do a business .</td>
</tr>
</tbody>
</table>
# ARAE: Unaligned Style Transfer

## Automatic Evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>Transfer</th>
<th>BLEU</th>
<th>PPL</th>
<th>Reverse PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Aligned AE</td>
<td>77.1%</td>
<td>17.75</td>
<td>65.9</td>
<td>124.2</td>
</tr>
<tr>
<td>ARAE</td>
<td>81.8%</td>
<td>20.18</td>
<td>27.7</td>
<td>77.0</td>
</tr>
</tbody>
</table>

## Human Evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>Transfer</th>
<th>Similarity</th>
<th>Naturalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Aligned AE</td>
<td>57%</td>
<td>3.8</td>
<td>2.7</td>
</tr>
<tr>
<td>ARAE</td>
<td>74%</td>
<td>3.7</td>
<td>3.8</td>
</tr>
</tbody>
</table>

(Similarity/Naturalness scores are between [1,5], 5 being best)
### ARAE: Unaligned Style Transfer

**Topic Transfer from Yahoo! Answers Dataset**

<table>
<thead>
<tr>
<th>Science</th>
<th>what is an event horizon with regards to black holes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music</td>
<td>what is your favorite sitcom with adam sandler?</td>
</tr>
<tr>
<td>Politics</td>
<td>what is an event with black people?</td>
</tr>
<tr>
<td>Music</td>
<td>do you know a website that you can find people who want to join bands?</td>
</tr>
<tr>
<td>Science</td>
<td>do you know a website that can help me with science?</td>
</tr>
<tr>
<td>Politics</td>
<td>do you think that you can find a person who is in prison?</td>
</tr>
<tr>
<td>Politics</td>
<td>republicans: would you vote for a cheney / satan ticket in 2008?</td>
</tr>
<tr>
<td>Science</td>
<td>guys: how would you solve this question?</td>
</tr>
<tr>
<td>Music</td>
<td>guys: would you rather be a good movie?</td>
</tr>
</tbody>
</table>
ARAE: Conclusion

- Introduced a simple method for training a GAN for text by performing generation/discrimination in a continuous code space
- A (somewhat) successful text-GAN instantiation
- Can do unaligned style transfer through training an additional classifier (much exciting work in this area: Shen et al. 2017, Prabhumoye et al. 2018)
ARAE: Open source

All our code is available at: https://github.com/jakezhaojb/ARAE.

Poster: #58