Active Learning through Adversarial Exploration in Contrastive Objectives

Avishek (Joey) Bose
joey.bose@mail.utoronto.ca

Abstract

Learning by contrasting positive and negative samples is a general strategy adopted by many methods in cases where the original formulation has intractable computational requirements or is difficult to model. Noise Contrastive Estimation (NCE) and negative sampling in word embeddings are examples that make use of this contrastive learning strategy. In this paper, we take an active learning perspective on negative mining whereby negative examples are chosen via a mixture distribution containing an adversarially learned sampler. The resulting adaptive sampler finds harder negative examples, which forces the main model to learn better representations of the data. We empirically show on a toy task that negative examples mined through our strategy explores regions near the decision boundaries in the embedded space a behavior that is different from other active learning strategies. We give a characterization between the connection of selected active learning strategies and recent work in intrinsic motivation in Deep Reinforcement Learning and visual semantic embeddings. We evaluate our proposal on learning word embeddings and observe both faster convergence and improved results on multiple metrics.

1 Introduction

Many models learn by contrasting losses on observed positive examples with those on some fictitious negative examples, trying to decrease some score on positive ones while increasing it on negative ones. There are multiple reasons why such contrastive learning approach is needed. Computational tractability is one. For instance, instead of using softmax to predict a word for learning word embeddings, noise contrastive estimation (NCE) can be used in skip-gram or CBOW word embedding models [18]. Another reason is modeling need, as certain assumptions are best expressed as some score or energy in margin based or un-normalized probability models such as in triplet loss for face recognition and verification [24].

Given a scoring function, the gradient of model parameters on observed positive examples can be readily computed, but the negative phase requires a design decision about how to sample data. In noise contrastive estimation for word embeddings, a negative example is formed by replacing a component of a positive pair by randomly selecting a sampled word from the vocabulary, resulting in a fictitious word-context pair which would be unlikely to actually exist in the dataset.

Typically the corruption distribution is the same for all inputs like in skip-gram or CBOW NCE [5], rather than being a conditional distribution that takes into account information about the input sample under consideration. Furthermore, the corruption process usually only encodes human prior about what constitutes a hard negative sample rather than being learned from data. For these two reasons, the fixed simple corruption process often yield only easy negative examples. Easy negatives are sub-optimal for learning discriminative representation as they do not force the model to find critical characteristics of observed positive data, which has been observed in applications outside NLP previously [26].

CSC2547 Winter 2018

We hypothesize that if the negative sampling strategy is influenced by the model's current state learning converges faster in the number of samples observed by the model. Consequently, the decision to learn about which negative sample to pick next can be interpreted as a stochastic policy, continuously adapted to optimize some notion of learning progress. The concept of learning progress is prevalent in intrinsically motivated reinforcement learning where learning progress has been used to drive exploration [2][12]. In this work, we propose to augment the simple corruption noise process in embedding models with an adversarially learned conditional distribution, forming a mixture negative sampler that adapts to the underlying data and the embedding model training progress. The resulting method is referred to as adversarial contrastive estimation (ACE). The adaptive conditional model engages in a minimax game with the primary embedding model, much like in Generative Adversarial Networks (GANs)[8], where a discriminator net (D), tries to distinguish samples produced by a generator (G) from real data [9]. In ACE, the main model learns to distinguish between a real positive example and a fake negative sample selected by the mixture of a weak fixed NCE sampler and an adversarial generator. The main model and the generator take alternating turns to update their parameters. In fact, our method can be viewed as a conditional GAN [19] on discrete inputs, with a mixture generator consisting of a learned and a fixed distribution.

In our proposed ACE approach, the conditional sampler finds harder negatives than NCE, while being able to gracefully fall back to NCE whenever the generator cannot find hard negatives. The rest of the paper is organized as follows: we first survey a few related works that have negative sampling approaches for learning embeddings models. We then provid a formal description of the ACE framework and the learning procedure for the generator. We then give an outline of various active learning strategies and provide an explicit connection between one such strategy and the intrinsic reward in [14]. Empirically, we observe that ACE samples on a toy task are usually near the discriminator decision boundary. Guided by the intuition on the toy task we evaluate ACE on learning word embeddings on a medium size dataset.

2 Related Work

Concurrent to this work, there has been many interests in applying the GAN approach to NLP problems [7, 28, 3]. Knowledge graph models naturally lend to a GAN setup, and has been the subject of study in [28] and [3]. Instead we focus on negative mining and specifically hard negatives for learning embedding models as we wish compare and contrast the negative sampling strategies employed by ACE and the following relevant work.

2.1 Noise contrastive estimation

The typical NCE [5] approach in tasks such as word embeddings[18], order embeddings[27], and knowledge graph embeddings can be viewed as a special case of Expression. 4 by taking $p(y^-|x^+)$ to be some unconditional $p_{nce}(y)$.

This leads to efficient computation during training, however, $p_{nce}(y)$ sacrifices the sample efficiency of learning as the negatives produced by a fixed distribution, which does not tailor toward x^+, are not necessarily hard negative examples, and would not force the model to discover discriminative representation of observed positive data. As training progresses, more and more negative examples are correctly learned, the probability of drawing a hard negative examples diminishes further, causing slow convergence.

2.2 Hard Negative Mining in Visual Semantic Embeddings

Visual Semantic embeddings for cross modal retrieval refers to retrieving an image i and its corresponding caption c taken from a set of image caption pairs $S = \{i_n, c_n\}_{n=1}^N$ and vice versa. To do this the authors use a contrastive objective that assigns higher scores to positive examples while assigning lower scores to negative ones by a margin α. The objective function is equivalent to triplet loss [24].

$$L = \sum_{k}^{K} (\alpha - s(i, c) + s(i, c')) + (\alpha - s(i, c) + s(i', c)) \tag{1}$$

where K is number of data points and $c' \neq c$ represents a caption that does not correspond to image i, and conversely image $i' \neq i$ does not correspond to caption c. Minimizing the triplet loss is equivalent to maximizing the margin between positive pairs and negative pairs. Positive examples are readily available but constructing negative examples are a design choice. In [6], authors introduce hard negative for visual semantic embedding task, which are negatives closest to each training query in embedding space, which we later show is equivalent to uncertainty sampling in an active learning setting. Then the triplet loss is rewritten as:

$$L = \sum_{k}^{K} (\max_{(i,c')\in N1} (\alpha - s(i,c) + s(i,c')) + \max_{(i',c)\in N2} (\alpha - s(i,c) + s(i',c))) \tag{2}$$

3 Method

3.1 Contrastive learning

In the most general form, our method applies to supervised learning problems with a contrastive objective of the following form:

$$L(\omega) = \mathbb{E}_{p(x^+,y^+,y^-)}\, l_\omega(x^+, y^+, y^-) \tag{3}$$

where $l_\omega(x^+, y^+, y^-)$ captures both the model with parameters ω and the loss that scores a positive tuple (x^+, y^+) against a negative one (x^+, y^-). $\mathbb{E}_{p(x^+,y^+,y^-)}(.)$ denotes expectation with respect to some joint distribution over positive and negative samples. Furthermore, by the law of total expectation, and the fact that given x^+, the negative sampling is not dependent on the positive label, i.e. $p(y^+, y^-|x^+) = p(y^+|x^+)p(y^-|x^+)$, Eq. 3 can be re-written as

$$\mathbb{E}_{p(x^+)} \left(\mathbb{E}_{p(y^+|x^+)p(y^-|x^+)}\, l_\omega(x^+, y^+, y^-) \right) \tag{4}$$

Separable loss

In the case where the loss decomposes into a sum of two terms as $l_\omega(x^+, y^+, y^-) = s_\omega\,(x^+, y^+) - \tilde{s}_\omega\,(x^+, y^-)$, then Expression. 4 becomes

$$\mathbb{E}_{p^+(x)}(\mathbb{E}_{p^+(y|x)}\, s_\omega\,(x,y) - \mathbb{E}_{p^-(y|x)}\, \tilde{s}_\omega\,(x,y)) \tag{5}$$

where we moved the $+$ and $-$ to p for notational brevity. Learning by stochastic gradient descent aims to adjust ω to pushing down $s_\omega\,(x,y)$ on samples from p^+ while pushing up $\tilde{s}_\omega\,(x,y)$ on samples from p^-. Note that for generality, the scoring function for negative samples, denoted by $\tilde{s}_\omega$, could be slightly different from s_ω.

Non separable loss

Eq. 3 is the general form that we would like to consider because for certain problems, the loss function cannot be separated into sums of terms containing only positive (x^+, y^+) and terms with negatives (x^+, y^-). An example of such non-separable loss is the triplet ranking loss [24]: $l_\omega = \max(0, \eta + s_\omega\,(x^+, y^+) - s_\omega\,(x^+, y^-))$, which does not decompose due to the rectification.

3.2 Adversarial mixture noise

To remedy the above mentioned problem of a fixed unconditional negative sampler, we propose to augment it into a mixture one $\lambda p_{nce}(y) + (1-\lambda) g_\theta(y|x)$, where g_θ is a conditional distribution with a learnable parameter θ and λ is a hyperparameter. The objective Expression. 4 can then be written as (conditioned on x for notational brevity):

$$\begin{aligned} L(\omega, \theta; x) &= \lambda\, \mathbb{E}_{p(y^+|x)p_{nce}(y^-)}\, l_\omega(x, y^+, y^-) \\ &+ (1-\lambda)\, \mathbb{E}_{p(y^+|x)g_\theta(y^-|x)}\, l_\omega(x, y^+, y^-) \end{aligned} \tag{6}$$

We learn (ω, θ) in a GAN-style minimax game:

$$\min_\omega \max_\theta V(\omega, \theta) = \min_\omega \max_\theta \mathbb{E}_{p^+(x)}\, L(\omega, \theta; x) \tag{7}$$

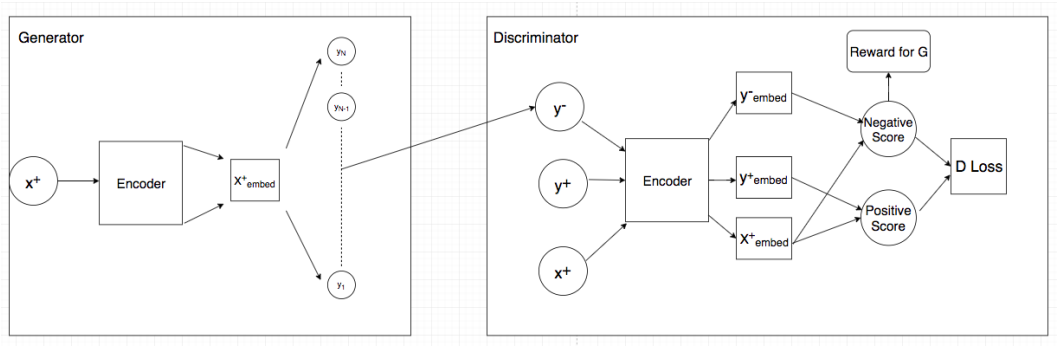


Figure 1: ACE architecture. The generator computes a categorical distribution over all possible actions before sampling a y^-. The discriminator computes a positive score on real data (x^+, y^+) and a negative score on (x^+, y^-). The generator learns to fool the discriminator using the negative score as its reward.

The embedding model behind $l_\omega(x, y^+, y^-)$ is similar to the discriminator in (conditional) GAN (or critic in Wasserstein[1] or Energy-based GAN[30], while $g_\theta(y|x)$ acts as the generator. Henceforth, we will use the term discriminator (D) and embedding model interchangeably, and refer to g_θ as the generator.

3.3 Learning the generator

There is one important distinction to typical GAN: $g_\theta(y|x)$ defines a categorical distribution over possible y values, and samples are drawn accordingly; in contrast to typical GAN over continuous data space such as images, where samples are generated by an implicit generative model that warps noise vectors into data points. Due to the discrete sampling step, g_θ cannot learn by receiving gradient through the discriminator. One possible solution is to use the Gumbel-softmax reparametrization trick [15], which gives a differentiable approximation. However, this differentiability comes at the cost of drawing N Gumbel samples per each categorical sample, where N is the number of categories. For word embeddings, N is the vocabulary size, and for knowledge graph embeddings, N is the number of entities, both leading to infeasible computational requirements.

Instead, we use the REINFORCE ([29]) gradient estimator for $\nabla_\theta L(\theta, x)$:

$$(1-\lambda)\,\mathbb{E}\left[-l_\omega(x, y^+, y^-)\nabla_\theta \log(g_\theta(y^-|x))\right] \tag{8}$$

where the expectation $\mathbb{E}$ is with respect to $p(y^+, y^-|x) = p(y^+|x)g_\theta(y^-|x)$, and the discriminator loss $l_\omega(x, y^+, y^-)$ acts as the reward.

With a separable loss, the (conditional) value function of the minimax game is:

$$\begin{aligned} L(\omega, \theta; x) &= \mathbb{E}_{p^+(y|x)}\, s_\omega\,(x, y) \\ &- \mathbb{E}_{p_{nce}(y)}\, \tilde{s}_\omega\,(x, y) - \mathbb{E}_{g_\theta(y|x)}\, \tilde{s}_\omega\,(x, y) \end{aligned} \tag{9}$$

and only the last term depends on the generator parameter ω. Hence, with a separable loss, the reward is $-\tilde{s}(x^+, y^-)$. This reduction does not happen with a non-separable loss, and we have to use $l_\omega(x, y^+, y^-)$. The entire ACE architecture is illustrated in Figure 1.

3.4 Variance Reduction

The basic REINFORCE gradient estimator is poised with high variance, so in practice one often needs to apply variance reduction technique. The most basic form of variance reduction is to subtract a baseline from the reward. As long as the baseline is not a function of actions (i.e. samples y^- being drawn), the REINFORCE gradient estimator remains unbiased. We propose to use the self-critical baseline method [21], where the baseline is $b(x) = l_\omega(y^+, y^\star, x)$, or $b(x) = -\tilde{s}_\omega(y^\star, x)$ in the separable loss case, and $y^\star = \text{argmax}_i g_\theta(y_i|x)$. In other words, the baseline is the reward of the most likely sample according to the generator.

4 Background on Active Learning

Active Learning refers to instances where there exists a small labeled set, L, which can readily be used by the model to learn from and a large unlabeled set, U, from which a sample(s) must be chosen to be labeled [25]. In active learning a model or learner may initially train for a few iterations on L, request labels for one or more selected instances chosen based on some metric such as the models uncertainty, learn from the query results, and then use its updated knowledge for a future query. Thus the goal in active learning is to minimize the number of queries and the associated labeling cost while maximizing accuracy. While a complete survey of query strategies is beyond the scope of this paper and interested readers should refer to [25], we present a few popular querying strategies that is most applicable to ACE.

4.1 Uncertainty Sampling

In uncertainty sampling the model chooses queries that it is least certain about. In binary classification this is simply picking the sample whose posterior probability of being positive is closest to 0.5 or in equations the optimal sample is $x^* = \arg\max(1 - P_\theta(\hat{y}|x))$. Hard negative mining in VSE++ and equation 2 is in fact a slightly modified multi-class setting of uncertainty sampling. The only difference being the objective in VSE++ is combined over images and captions as the goal is to learn joint embeddings for cross-modal retrieval. Uncertainty sampling in VSE++ has shown to be effective in learning embeddings models for image-caption retrieval [6] and achieves state of the art performance on various public benchmarks with a simple change in querying strategy that focuses on hard negative mining.

4.2 Expected Error Reduction

Expected Error Reduction (EER) presents a decision theoretic approach that picks a sample that minimizes the generalization error of the model. Specifically, samples are chosen to minimize the expected future error of the model. In binary classification, the optimal sample is $x^* = \arg\min \sum_i P_\theta(y_i|x)(\sum_{u=1}^{U}[1 - P_{\theta+(x_i,y_i)}(\hat{y}|x^u)]$ where the inner expectation comes from the fact that the true label is not known at query time and is approximated using an expectation over all possible labels under the current model. This definition serves to reduce the number of incorrect predictions. If instead the inner sum is replaced by log-loss then this is equivalent to minimizing the expected entropy over U [22].

$$x^* = \arg\min \sum_i P_\theta(y_i|x)(\sum_{u=1}^{U}\sum_j [P_{\theta+(x_i,y_i)}(y_j|x^u) \log P_{\theta+(x_i,y_i)}(y_j|x^u)]$$

In other words x^* is the sample that maximizes the information gain. A drawback of this approach is that it computationally expensive as it requires computing the expected future error for every query, before incrementally retraining the model.

4.3 Relationship to VIME

Variational Information Maximizing Exploration[14], uses an exploration strategy based on maximization of information gain about the agent's belief of environment dynamics. To do this VIME introduces an intrinsic motivation term that that encourages the agent to take actions that maximize the reduction in uncertainty about the environment dynamics. Specifically, the term calculates a difference in entropy between the current state and expected future state. Thus VIME measures the KL divergence between the posterior before and after a step in parameter space which is then added to the reward function. The intrinsic motivation or curiosity term in VIME is equivalent using an EER query strategy where we minimize expected entropy.

5 Applications of ACE

5.1 ACE on a Toy Task

We first apply ace to a contrived toy task of classifying MNIST digits by first learning an embedding for each training data point. To train the model we use triplet loss which is equivalent to using a non-separable loss in our formulation. Intuitively, as the model learns we expect distinct clusters to

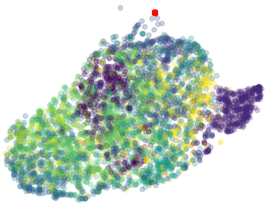


Figure 2: Epoch 1

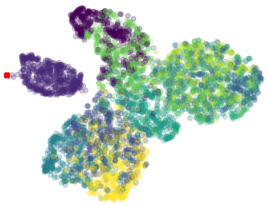


Figure 3: Epoch 3

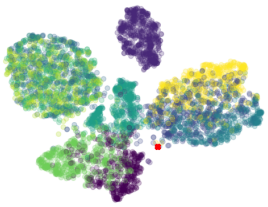


Figure 4: Epoch 5

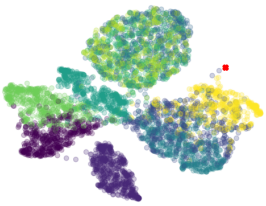


Figure 5: Epoch 7

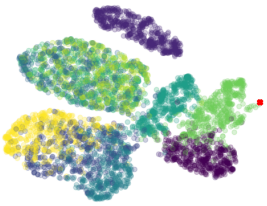


Figure 6: Epoch 9

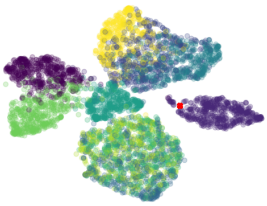


Figure 7: Epoch 10

be formed representing each class. The goal of learning an intermediate embedding representation first is not to achieve near state of the art classification accuracy, the model achieves 97.6%, but instead to explicitly visualize and understand the generator sampling process in two dimensions using TSNE [17]. We plot the embeddings by sampling 5000 random data points for alternating epochs up to a total of 10. In each figure a color represents a different digit and the red dot represents the training sample picked by the generator. If hard negatives are chosen based on an uncertainty sampling strategy then for a given data point the hardest negative is then another point that is closest in embedding space but is a different class [6]. Contrastingly, the samples chosen by ACE are observed to be near the edge of a decision boundary for a class whereas one might expect samples chosen using uncertainty to be taken from areas where classes are mixed or not well separated. Thus in ACE a hard negative example is not necessarily close to any data point in embedding space but instead is a point that the main model has trouble distinguishing from real data. These two definitions of what constitutes a hard negative leads to qualitatively different sampling behaviors and thus quantitative results.

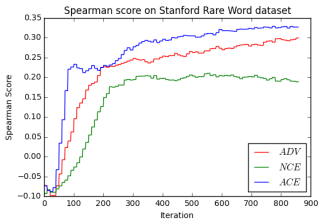
5.2 Application of ACE on Word Embeddings

Word embeddings learn vector representation of words from co-occurrences in a text corpus. NCE casts this learning problem as a binary classification where the model tries to distinguish positive word and context pairs, from negative noise samples composed of word and false context pairs. The NCE objective in Skip-gram ([18]) for word embeddings is a separable loss of the form:

$$\begin{aligned} L = -\sum_{w_t \in V} [&\log p(y=1|w_t, w_c^+) \\ &+ \sum_{c=1}^{K} \log p(y=0|w_t, w_c^-)] \end{aligned} \tag{10}$$

Here, w_c^+ is sampled from the true set of contexts and $w_c^- \sim Q$ is sampled k times from a fixed noise distribution. Mikolov *et al.* [18] introduced a further simplification of NCE, called "Negative Sampling". With respect to our ACE framework, the difference between NCE and Negative Sampling is inconsequential, so we continue the discussion using NCE. A drawback of this sampling scheme is that it favors more common word as context. Another issue is that the negative context words are sampled in the same way, rather than tailored toward the actual target word. To apply ACE to this problem we first define the value function for the minimax game, $V(D, G)$, as follows:

$$\begin{aligned} V(D,G) = &\mathbb{E}_{p^+(w_c)}[\log D(w_c, w_t)] \\ &- \mathbb{E}_{p_{nce}(w_c)}[-\log(1 - D(w_c, w_t))] \\ &- \mathbb{E}_{g_\theta(w_c|w_t)}[-\log(1 - D(w_c, w_t))] \end{aligned} \tag{11}$$



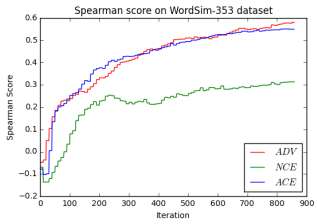


Figure 8: **Left**: Rare Word, **Right**: WS353 similarity scores during the first epoch of training.

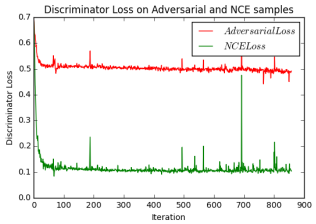


Figure 9: Training from scratch losses on the Discriminator

with $D = p(y = 1|w_t, w_c)$ and $G = g_\theta(w_c|w_t)$.

6 Results, Discussion, and Limitations

6.1 Training Word Embeddings from scratch

In this experiment we empirically observe that training word embeddings using ACE converges significantly faster than NCE after one epoch. As shown in Fig.3 both ACE (mixture of p_{nce} and g_θ) and just g_θ (denoted by ADV) significantly outperforms the NCE baseline, with an absolute of 73.1% and 58.5% respectively on RW score. We note similar results on WordSim-353 dataset where ACE and ADV outperforms NCE by 40.4% and 45.7%. We also evaluate our model qualitatively by inspecting the nearest neighbors of selected words in the appendix. We first present the five nearest neighbors to each word to show that both NCE and ACE models learn sensible embeddings. We then show that ACE embeddings have much better semantic relevance in larger neighborhood (nearest neighbor 45-50).

6.2 Finetuning Word Embeddings

We take off-the-shelf pre-trained Glove embeddings which were trained using 6 billion tokens [20] and fine-tune them using our algorithm. It is interesting to note that the original Glove objective does not fit into the contrastive learning framework, but nonetheless we find that they benefit from ACE. In fact, we observe that training such that 75% of the words appear as positive contexts is sufficient to beat the largest dimensionality pre-trained Glove model on word similarity tasks. We evaluate our performance on the Rare Word and WordSim353 data. As can be seen from our results in Table. 1, ACE on RW is not always better and for the 100d and 300d Glove embeddings is marginally worse. However, on WordSim353 ACE does considerably better across the board to the point where 50d Glove embeddings outperform the 300d baseline Glove model.

Table 1: Spearman score ($\rho * 100$) on RW and WS353 Datasets. We trained a skipgram model from scratch under various settings for only 1 epoch on wikipedia. For finetuned models we recomputed the scores based on the publicly available 6B tokens Glove models and we finetuned until roughly 75% of the vocabulary was seen.

	RW	WS353
Skipgram Only NCE baseline	18.90	31.35
Skipgram + Only ADV	29.96	58.05
Skipgram + ACE	32.71	55.00
Glove-50 (Recomputed based on[20])	34.02	49.51
Glove-100 (Recomputed based on[20])	36.64	52.76
Glove-300 (Recomputed based on[20])	**41.18**	60.12
Glove-50 + ACE	35.60	60.46
Glove-100 + ACE	36.51	63.29
Glove-300 + ACE	40.57	**66.50**

6.3 Hard Negative Analysis

To better understand the effect of the adversarial samples proposed by the generator we plot the discriminator loss on both p_{nce} and g_θ samples. In this context, a harder sample means a higher loss assigned by the discriminator. Fig. 2 shows that discriminator loss for the word embedding task on g_θ samples are always higher than on p_{nce} samples, confirming that the generator is indeed sampling harder negatives.

6.4 Limitations

The formulation of ACE as introduced in equation 6 has a few notable drawbacks. Firstly, generator defines a categorical distribution over a set of possible actions which is computed via softmax. In the case of word embeddings the softmax is computed over the vocabulary which can be very large and thus computationally expensive. Although ACE converges faster per iteration, it may converge more slowly on wall-clock time depending on the cost of the softmax. We believe that the computational cost could potentially be reduced via the "Augment and Reduce" variational inference trick of [23], or the application of adaptive softmax [11] but leave that as future work. Another limitation is that no theoretical guarantees can be provided as ACE is an application of GAN's. As noted in [10] GAN learning does not implement maximum likelihood estimation (MLE), while NCE has MLE as an asymptotic limit. To the best of our knowledge, more distant connections between GAN and MLE training are not known, and tools for analyzing MinMax game where player(s) are parametrized by neural nets are currently not available. Furthermore, GAN training can suffer from instability and degeneracy where the generator probability mass collapses to a few modes or points. Much work has been done to stabilize GAN training in the continuous case [1, 13, 4]. In ACE, if the generator g_θ probability mass collapses to a few candidates, then after the discriminator successfully learns about these negatives, g_θ cannot adapt to select new hard negatives, because the REINFORCE gradient estimator Eq. 8 relies on g_θ being able to explore other candidates during sampling. Therefore, if g_θ probability mass collapses, instead of leading to oscillation in typical GAN, the min-max game in ACE reaches an equilibrium where the discriminator wins and g_θ can no longer adapt. One potential fix that occasionally works is to add an entropy term to the generator loss that forces g_θ to have high entropy and thus not collapse but tuning this term is difficult and we leave it to future work. We have discussed the limitations of our work in detail throughout this section. Here, we briefly re-iterate the three major limitations:

- Computational overhead of the generator softmax is high when the set of actions is large.
- We provide no theoretical guarantees on ACE and note that it does not implement MLE as its asymptotic limit which NCE does.
- Instability in training can cause the generator to collapse to a few hard negatives preventing the generator to adapt further.

7 Conclusion

In this paper we propose Adversarial Contrastive Estimation as a general technique for improving supervised learning problems that learn by contrasting observed and fictitious samples. Specifically, we use a generator network in a conditional GAN like setting to propose hard negative examples for our discriminator model. We find that a mixture distribution of randomly sampling negative examples along with an adaptive negative sampler leads to improved performances on learning word embeddings. We validate our hypothesis that hard negative examples are critical to optimal learning and can be proposed via our ACE framework. We explore the negative sampling behavior of ACE on a toy task and provide commentary on the qualitative differences to other active learning strategies. Interestingly, we find that hard negative mining in VSE++ and intrinsic motivation in VIME have explicit connection to different active learning strategies while ACE to the best of our knowledge does not.

Using ACE leads to much faster convergence but there is computational overhead caused by the generator network learning. Techniques such as [23] and [11] are a promising future direction to scaling up ACE to even larger datasets.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Andrew G Barto, Satinder Singh, and Nuttapong Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, pages 112–19. Citeseer, 2004.

[3] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

[4] Yanshuai Cao, Gavin Weiguang Ding, Kry Yik-Chau Lui, and Ruitong Huang. Improving gan training via binarized representation entropy (bre) regularization. *International Conference on Learning Representations*, 2018. accepted as poster.

[5] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[6] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017.

[7] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*, 2018.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Ian J Goodfellow. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:1412.6515*, 2014.

[11] Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. Efficient softmax approximation for gpus. *arXiv preprint arXiv:1609.04309*, 2016.

[12] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[14] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016.

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *ArXiv e-prints*, November 2014.

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[21] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[22] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[23] Francisco JR Ruiz, Michalis K Titsias, and David M Blei. Scalable large-scale classification with latent variable augmentation.

[24] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[25] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[26] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[27] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *ICLR*, 2016.

[28] Peifeng Wang, Shuangyin Li, and Rong Pan. Incorporating gan for negative sampling in knowledge representation learning. 2018.

[29] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[30] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

A Implementation Details

For our experiments we train all our models on lowercased unigrams taken from one pass through the May 2017 dump of the English Wikipedia. The vocabulary size is restricted to the top $150k$ most frequent words when training from scratch while for finetuning we use the same vocabulary as [20]. We use 5 NCE samples for each positive sample and 1 adversarial sample in a window size of 10 and the same positive subsampling scheme proposed by [18]. Learning for both G and D uses Adam [16] optimizer with its default parameters. Our conditional discriminator is modeled using the Skip-Gram architecture, which is a two layer neural network with a linear mapping between the layers. The generator network consists of an embedding layer followed by two small hidden layers, followed by output softmax layer. The first layer of generator shares its weights with second embedding layer in the discriminator network, which we find really speeds up convergence as the generator doesn't have to relearn its own set of embeddings. The difference between the discriminator and generator is that a sigmoid nonlinearity is used after the second layer in the discriminator, while in the generator, a softmax layer is used to define categorical distribution over negative word candidates. We find that controlling the generator entropy is critical for finetuning experiments as otherwise the generator collapses to its favorite negative sample. The word embeddings are taken to be the first dense matrix in the conditional discriminator.

B Supplementary Material: Qualitative Analysis of Nearest Neighbors in Word Embedding Models

Table 2: Top 5 Nearest Neighbors of Words followed by Neighbors 45-50 for different Models.

	Queen	King	Computer	Man	Woman
Skip-Gram NCE Top 5	princess	prince	computers	woman	girl
	king	queen	computing	boy	man
	empress	kings	software	girl	prostitute
	pxqueen	emperor	microcomputer	stranger	person
	monarch	monarch	mainframe	person	divorcee
Skip-Gram NCE Top 45-50	sambiria	eraric	hypercard	angiomata	suitor
	phongsri	mumbere	neurotechnology	someone	nymphomaniac
	safrit	empress	lgp	bespectacled	barmaid
	mcelvoy	saxonvm	pcs	hero	redheaded
	tsarina	pretender	keystroke	clown	jew
Skip-Gram ACE Top 5	princess	prince	software	woman	girl
	prince	vi	computers	girl	herself
	elizabeth	kings	applications	tells	man
	duke	duke	computing	dead	lover
	consort	iii	hardware	boy	tells
Skip-Gram ACE Top 45-50	baron	earl	files	kid	aunt
	abbey	holy	information	told	maid
	throne	cardinal	device	revenge	wife
	marie	aragon	design	magic	lady
	victoria	princes	compatible	angry	bride