Similarity-Induced Embeddings for Classification

Di Qi

Advisor: Professor Yoram Singer

A thesis presented for the degree of

Bachelor of Science in Engineering

Department of Computer Science, Princeton University

May 15, 2018

Contents

Abstract

We apply similarity learning, a technique commonly used in the ranking setting, to obtain a task-independent embedding. Such an embedding space is more expressive than the embedding space induced by standard classifiers, in that it can express more detailed relationships between and within classes. In doing so, we can significantly reduce the dimensionality of the representation of the data in a way that does not depend directly on the number of classes. We apply this embedding to the task of image classification. This approach also has the advantage that similarity learning is a weaker form of supervision than classification. Using category-level similarity data, we obtain comparable performance to classifiers trained specifically for this task.

Acknowledgements

My advisor, Yoram, without whom this thesis would not exist. Thank you for guiding me in exploring a new topic, and being so patient and understanding with me. Ive learned so much over this past year with you.

Prof. Schapire, Karan, and Riley, for helping me think through roadblocks and giving me new ideas to look into.

Sergiy, Danny, and Xiaoyan for helpful discussions as I wrote this thesis.

Jeung for the coffee, food, and company while I tried to distract myself from this thesis.

Yannis, because our COS511 final project was really great.

The Official Princeton Class of 2018+ Page, without which I would not have made my first Princeton friend, Fida. Thanks for being there for four years – through break ups, classes, and theses.

My lovely roommates over the past four years Marissa, Carly, Zoe, Anna, and Casey. Thank you for all of the love and affection and friendship.

Brown Co-op, for a sweet home my last two years. For the cooking, laughs, quote walls, alignment charts, late nights, busy nights, and more. I'm happy for the friends that I made here.

Brittany, Heeral, and Khelan, for always, always being there for me, even from a distance.

My mom and dad. For loving me always, supporting me always, and wanting the best for me always. I love you.

1 Introduction

1.1 Motivation

Good representations are useful. A good representation of data reduces the dimensionality of data and encodes useful information, enabling machine learning methods to perform better. The Tiny images dataset contains images on the order of 3072 pixels [52]. In this work, we consider embedding spaces of dimensionality 10, 20, and 40, a significant reduction.

An embedding space also does not need to depend directly on the number of classes, which is useful when the number of classes is large. Language embeddings have been very successful, and this success has transferred to other fields. For example, some work in computer vision utilizes language embeddings to get useful image embeddings that improve performance in computer vision tasks.

However, language-based embeddings do not necessarily capture the distribution of images. Even within images under the same classification label, there is considerable visual variance [10]. Thus, we find that it would be fruitful to consider representations specifically for the visual space. We specifically consider similarity learning for this task because it allows us to build more meaningful representations that extend beyond the task of classification. This is because it can encode beyond just category-level relationships, but also within-category relationships.

Similarity learning is a weaker form of supervision. Large, labeled datasets combined with advances in deep learning have enabled state-of-the-art results in image classification. ImageNet [45] is one example of a large, labeled dataset often used for classification. It contains over 14 million images. The Tiny Images dataset is a larger dataset, containing 80 million images, but is not as often used for classification. This is because it contains

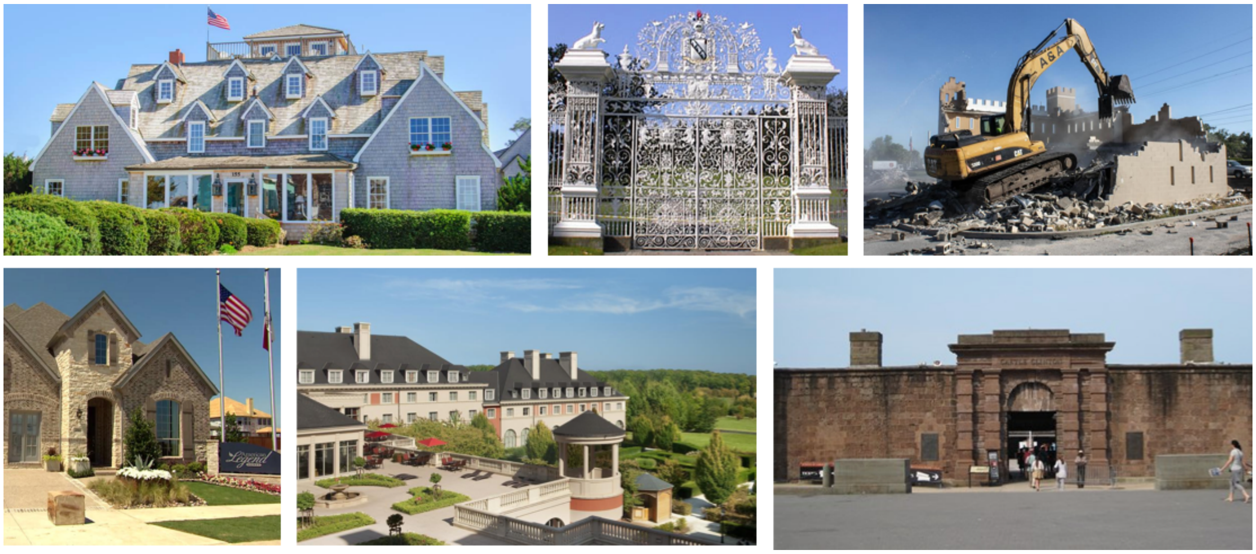


Figure 1: Images of castles found with Google

noisy class labels. It is constructed from Google Image Search queries. Figure 1 contains an examples of images for the query “castle, one of the labels in the dataset. Some images were irrelevant to castles. Others (depicted) were not castles, but were related to castles.

Similarity learning is a weaker form of supervision that can be better suited to noise in datasets. Noisy datapoints do not necessarily distort the similarity-induced embedding, although they might distort the classification-induced embedding. In addition, we can make use of linguistic information that already exists to improve results from similarity learning. For example, in CIFAR-100, a subset of the Tiny Images dataset, castle, house, road, skyscraper, and bridge all fall under the provided coarse label “large man-made outdoor things”.

1.2 Contributions

Similarity learning is often applied to ranking problems. In this thesis, we consider its ability to generate meaningful and informative representations for data. We focus our attention on image data, but we believe that it should be clear that the utility of this framework extends to other media. We introduce an architecture to generate a similarity-induced embedding space, test the usefulness of the embedding space on the task of classification, and evaluate its performance on academic baselines. Lastly, we visualize the embeddings. Along the way, we provide an overview of fields including representation learning, similarity learning, and zero-shot learning.

2 Related Work

2.1 Semantic Embedding Spaces

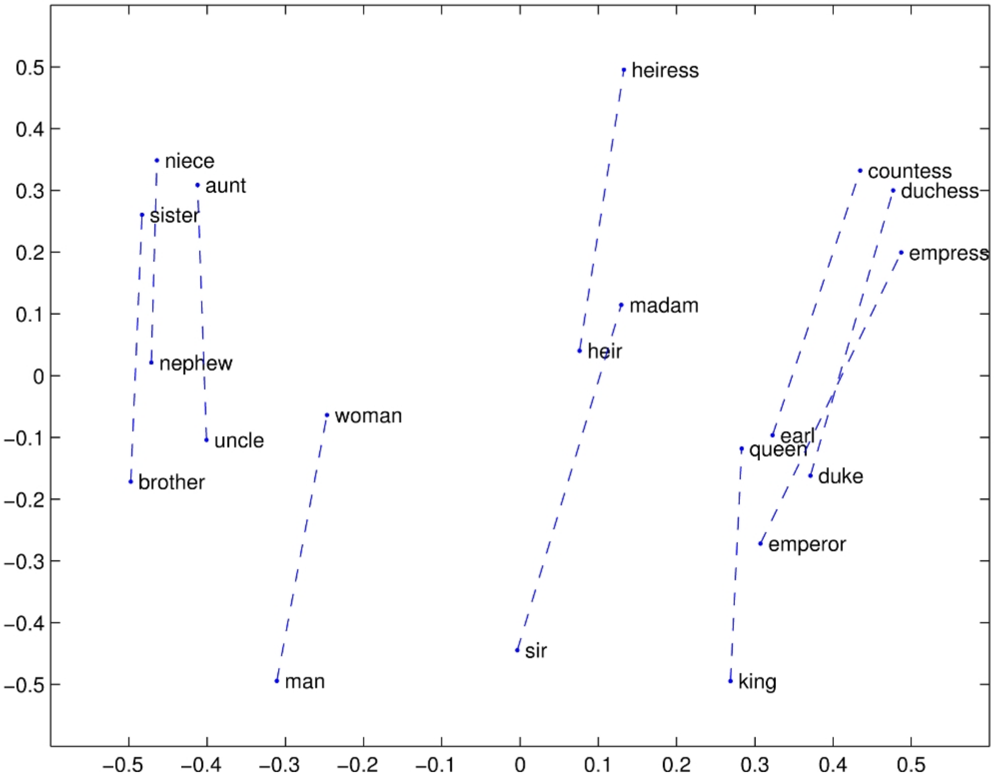


Figure 2: Visualization of a subset of GloVe word embeddings. Demonstrates an underlying concept that distinguishes man from woman. [43]

High-dimensional data is common in many areas. Consider text. The English language is estimated to contain roughly one million words [1]. We represent these words in letters. With this representation, humans can perform many complex analytical tasks, such as sentiment analysis, question answering, and translation. But it is much more challenging for computers to perform these tasks with the same representations. Representation learning attempts to automatically learn good representations of data to facilitate automation of useful tasks.

Considering words as discrete symbols that can be encoded in a one-hot manner reflects how we use language. But this approach has several downsides. It is a sparse representation, which usually means that we need more data to successfully train statistical models [2]. It also doesnt reflect our understanding of language. There is no natural notion of similarity or relationships between words. A resource containing lists of synonym sets and hypernyms like WordNet [14] could potentially address the latter issue, but it lacks nuance and does not adapt well to changes over time [47].

A more natural approach that mimics aspects of human understanding of language is to encode words into a vector space–an embedding. Here there are natural notions of similarity (distances between words in the embedding) and relationships (differences between words). An example of such embeddings can be seen in Figure 2. They can be trained on a large corpus of text available on the Internet, such as Wikipedia articles. These work because words that appear in the same contexts share semantic meaning [2].

The two main approaches for generating semantic embeddings are are count-based methods and predictive methods. Count-based methods compute word co-occurrence statistics and aim to generate contexts for words. Predictive methods aim to predict a word from its context. Two of the most well known of these are word2vec [39] and GloVe [43] respectively. These embeddings have facilitated much of the progress in natural language processing [47, 8, 54]. Distributed language representations have also been extended [24] to the level of sentences, phrases, and paragraphs [53, 51, 40, 20, 32] entities and relationships [3, 48], and semantic categories [9, 15]. More interestingly, semantic embeddings have also contributed to progress in other fields, such as computer vision, which we discuss in the next subsection.

2.2 Images and Semantic Embedding Spaces

Mapping images to semantic embedding spaces. Semantic embeddings have been extremely successful. They are very expressive despite not depending on a large labeled dataset and have generated significant results in a multitude of fields. In contrast, for images, the standard machine learning approach to classification presents more challenges, since it depends on training data that is expensive and challenging to obtain. The task demands high-quality, fine-grained labels, but there are many object categories, and new object categories are always appearing that demand revisions to previous classification models.

These challenges have motivated several recent papers to approach classification by mapping images into semantic embedding spaces using class labels [41, 15, 49, 13, 29]. Most of these take a regression-based approach, learning a direct mapping of images into semantic embedding spaces [15, 49, 13, 29]. [41] simply takes a weighted combination of the word embeddings associated with the image labels and gets comparable results. A general observation that underlies these works is that nearest neighbor search within the embedding space can address zero-shot learning.

Joint image-word embedding spaces and multi-modal embedding spaces. There have also been attempts to build off of the success of word representations by creating joint word-image embedding spaces [59, 60, 57], where mapping functions for words and images are different, but are trained together for the task. Since the purpose of embeddings is also for scalable learning, we note the scale of [59, 60]. Performed by researchers at Google in 2010 and 2011 respectively, both used 10 million training examples and 100 thousand annotations, the largest scale for image annotation work at the time. Other works have developed multi-modal embedding spaces, where examples from the two modalities are mapped to the same space, and thus can be compared directly [22, 50, 21]. These embedding spaces have been applied to fields such as visual question answering and image caption generation [55, 23, 21, 36, 11, 5, 12, 63, 33]. As an example, Kiros. et al, inspired by work in machine translation, combine image-text embedding models with multi-modal neural language models into an encoder-decoder pipeline in [23]. The encoder enables ranking of images and sentences, and the decoder can generate novel descriptions for images.

Discussion. In general, while there is a relationship between visual similarity and semantic similarity, images cannot be described by their labels alone, and thus image embeddings based solely on labels are limited in value. Work that combines image representations and semantic embedding spaces utilize information from both sources. This paper considers how we can improve the image representations, in particular with an image-specific embedding space, analogous to word embedding spaces. There is not much existing work on image-embedding spaces. However, work in similarity learning for images is relevant, and we discuss this subsequently. First we give an overview of similarity learning.

2.3 Similarity Learning: Overview

Similarity learning is the task of learning a similarity function over objects. The similarity function is defined between for pairs of instances. In determining similarity between two instances, the learned representation should assign a larger similarity score for more similar instances. Similarity learning is closely related to metric learning, which seeks to learn a distance function over objects.

For similarity learning, training data usually takes the form of pairs or triplets. In the pairs framework, there is a training set with pairs of instances with similarity feedback $S = \{(x_i^1, x_i^2, y_i)\}_{i=1}^m$. The feedback may be binary, with $y_i \in \{-1, +1\}$ indicating whether the two instances are similar or dissimilar; [46] or real-valued, with $y_i \in \mathbb{R}$ representing the degree of similarity. The first type of feedback often appears in classification similarity learning, and the second in regression similarity learning.

In the triplets framework, there is a training set with triplets of instances $\{(x_i, x_i^+, x_i^-)\}_{i=1}^m$, where x_i is more similar to x_i^+ and x_i^-. We note that this latter approach is more flexible than the former. Where the former approach resembles classification, in that objects are either similar or not similar, the triplet-based approach allows distinction of objects within the same category, and distinction of objects across categories. Because of the additional level of expressiveness of this training data, it is often used in the ranking similarity learning setup.

Similarity learning is often applied to ranking problems [6, 37, 56, 4]. It can be used to improve search results or to tailor music recommendations to someone's music interests. Similarity learning can also be formulated as a classification [4, 6] or regression [18] problem. For example, given two images of people, it can be used to determine if the same person in each photo.

2.4 Similarity Learning Applications

We take a standard approach to learning embeddings where the task is to get similar objects closer to each other than dissimilar objects [38, 62]. Alternative approaches for learning embeddings utilize the k-nearest neighbor criterion [58] or a clustering perspective [28, 30]. Other work has formulated the problem as multiple instance learning, where instances are labeled with "bags" of labels [31] Koch et. al apply pairwise similarity to learn an embedding space for one-shot image classification to get state-of-the-art results [25].

Similarity learning can also be applied to a broad array of other tasks. Examples include similarity search [27], ranking [4, 56], generating images [44], and classification [16, 17, 25].

The primary application of similarity learning is to solving a nearest neighbors-type problem such as ranking or search. In this area, [27] Introduced a method that used hash functions to perform nearest neighbor searches on linearly learned metrics. OASIS [4] was one of the first large-scale applications of similarity learning to ranking. It addressed the problem of scalability with respect to CPU and storage requirements by using introducing a linear algorithm that works over sparse representations. Deep learning approaches have also popped up in recent years [56]. Surprisingly OASIS performs comparably to these. A 2014 application of deep learning to the problem increased the precision from about 79.2% for OASIS to only about 85.7% [56].

There is limited work on applications of similarity learning to classification, although it is a natural extension of the ranking application. In [16, 17], Frome et. al learn a local distance metric for each example, and apply a nearest neighbor classifier to label test images. They also apply this framework to image retrieval. Koch. et al [25] remark that the embedding generated in training could be applied to the task of classification.

3 Approach

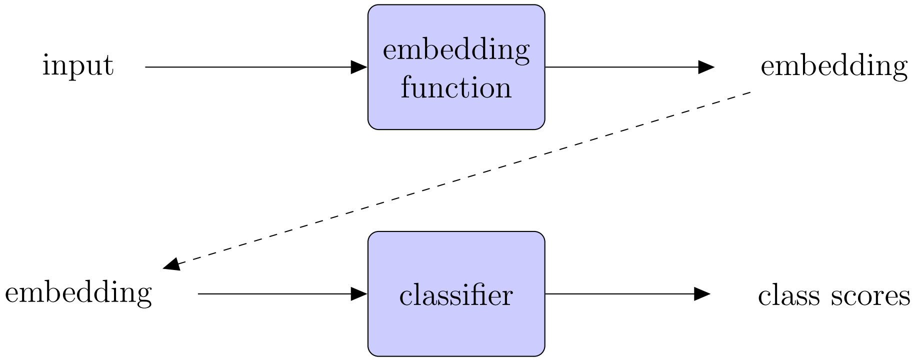


Figure 3: Structure of experiment

Our goal is an image embedding space that is useful for other tasks. We train a deep neural network on category-level similarity data to obtain the embedding. To evaluate the embedding, we use a linear classifier on the embeddings. The structure of our experiment can seen in Figure 3. We found results comparable to those obtained by a classifier of the same architecture trained specifically for the task.

Traditional deep neural network-based classifiers implicitly learn representations of data, but these representations are linked closely to the task of classification, and the existing classes. When new data appears, the classifiers must be retrained at significant time and computational cost. Similarity learning is a weaker form of supervision. The benefit of this similarity-induced embedding approach is that we build a complex, expressive representation that does not depend directly on the number of classes. A simple classifier can quickly be built on top of the existing representation when new data appears.

In designing an embedding function f, we want images from like classes to be similar and images from unlike classes to be dissimilar. We use the inner product $\langle \cdot \rangle$ and Euclidean distance $\|\cdot\|_2^2$ as measures of similarity. This work focuses on the inner product-based definition of similarity for reasons explained in the following subsection. For the inner product definition of similarity, if a pair of instances x, y are from the same class, then we want

$$\langle f(x), f(y) \rangle \gg 0 \tag{1}$$

and otherwise, we want

$$\langle f(x), f(y) \rangle \ll 0 \tag{2}$$

For triples x, y, z where x, y are from the same class and z is from a different class, we want

$$\langle f(x), f(y) \rangle \gg \langle f(x), f(z) \rangle \tag{3}$$

For the distance-based definition of similarity, we aim to minimize distance between like classes, and hence consider the negative of Euclidean distance as a measure of similarity. Subsequent definitions are also analogous.

Let $c(x, y) = 1$ if x, y are from the same class and -1 otherwise. We define the error for similarity learning for pairs as

$$e(x,y) = \begin{cases} 0 & \operatorname{sign}\left(\langle f(x), f(y)\rangle\right) = c(x,y) \\ 1 & o.w. \end{cases} \tag{4}$$

and the error for triplets as

$$e(x,y,z) = \begin{cases} 0 & \langle f(x), f(y)\rangle > \langle f(x), f(z)\rangle \\ 1 & o.w. \end{cases} \tag{5}$$

Then we can define the logistic and hinge surrogate loss for pairs as

$$l_L(x,y) = \max(0, \gamma - c(x,y)\langle f(x), f(y)\rangle) \tag{6}$$

$$l_H(x,y) = \log\left(1 + \exp(-c(x,y)\langle f(x), f(y)\rangle)\right) \tag{7}$$

and for triplets as

$$l_L(x,y,z) = \max(0, \gamma - \langle f(x), f(y) - f(z)\rangle) \tag{8}$$

$$l_H(x,y,z) = \log\left(1 + \exp(-\langle f(x), f(y) - f(z)\rangle)\right) \tag{9}$$

3.1 Theoretical Discussion

We consider the setting in which data is independently and identically distributed. Let $x \sim c$ if the instance x belongs to the class c. Let S_c be the space of instances from class c. Imagine we have a function f that satisfies the following property: For all $x, y \in S_c$, and $z \in S_{c'}$ such that $c \neq c'$, $f(x) \cdot f(y) > f(x) \cdot f(z)$. Then given some point $x \sim c$, for any $x_c \in S_c, x_{c'} \in S_{c'}$, we have that

$$f(x) \cdot f(x_c) > f(x) \cdot f(x_{c'}) \tag{10}$$

This would imply that the classes are linearly separable.

Consider a more realistic scenario. Suppose that with probability ϵ, where ϵ is relatively small, for $x, y \in S_c$ and $z \in S_{c'}$ where $c \neq c'$, $f(x) \cdot f(y) \leq f(x) \cdot f(z)$. Suppose $x \sim c$ and there are K classes. Select some instance $x_{c'}$ at random from each of the other $K-1$ classes. By the union bound, for all $x_{c'}$, $f(x) \cdot f(x_c) > f(x) \cdot f(x_{c'})$ with probability $1-(K-1)\epsilon$.

4 Experiments

Our experiments were done in TensorFlow, and can be found at the following Github repository:

`https://github.com/dqii/thesis`

We evaluate our method on four academic baselines: MNIST [34], Fashion MNIST [61], CIFAR-10, and CIFAR-100 [26]. These datasets are split into 54,000 training instances, 6,000 validation instances, and 10,000 test instances. We performed data augmentation, doubling the size of the training set with images that were randomly rotated, shifted, and/or flipped. The datasets for similarity learning were derived from uniformly sampling pairs and triplets from the datasets. For triplets, this was done by randomly selecting query images, then randomly selecting an image in the same class, and an image in a different class.

CIFAR-100 contains more classes than the other three baselines, and is more representative of the desired scenario in that the classes can be grouped into related categories. Specifically, CIFAR-100 contains 100 classes, and these can be grouped into 20 superclasses containing 5 classes each. One example is the superclass “flowers”, which contains orchids, poppies, roses, sunflowers, and tulips. Thus, we also considered a separate experiment that considered these superclass-level similarities in addition to the class-level similarities. For triplets based on superclass-level similarities, we chose the second image to be an image in the same superclass, and the third image to be from a different superclass.

Because the datasets vary in difficulty, we used different architectures for the embedding function. These architectures are described in Table 1. For the more difficult datasets, we used group equivariant convolutional neural networks (G-CNNs) [7], a generalization of convolutional neural networks that is equivariant to rotations. Spatial transformer networks [19] are a more well-known architecture for learning more general types of spatial invariance, but we were not able to achieve good performance with them. For the more difficult embeddings, we also had more iterations over the dataset.

Due to the differences in difficulty, we also train for different numbers of iterations. At each iteration, we sample 300 triplets. For MNIST, we run 20,000 iterations to train the embedding. For Fashion MNIST, we run 40,000 iterations. For CIFAR-10 and CIFAR-100, we run 60,000 iterations. To train the classifier, we run 10,000 training iterations. When running the baselines, we combine the number of iterations to train the embedding and the classifier.

We compared the performance of linear classifiers and two-layer neural nets that used the embedding against the performance of a baseline classifier with the same architecture that used the raw data. We trained our classifiers using softmax cross-entropy.

Table 1: Model Architectures. G refers to a G-CNN, batch normalization, ReLU, and dropout layer and accepts as arguments the kernel size, number of output channels, the stride, and the dropout keep rate. C is defined analogously for convolutional layers. FC refers to a fully-connected and ReLU layer. Out refers to a fully-connected layer, and is used for the final output layer of the model. For all models, we use L2 regularization over all parameters with a lambda value of 0.001.

Dataset	**Model Architecture**

MNIST	C(kernel=3x3, channels=16, stride=1, keeprate=0.8)
	C(kernel=3x3, filters=16, stride=2, keeprate=0.8) × 3
	F(in=4 * 4 * 16, out=64)
	Out(in=64, out=#features)

Fashion MNIST	G(kernel=3x3, channels=16, stride=2, keeprate=0.8) × 2
	G(kernel=3x3, filters=32, stride=2, keeprate=0.8) × 2
	F(in=4096, out=1024)
	F(in=1024, out=256)
	Out(in=256, out=#features)

CIFAR-10
CIFAR-100
	G(kernel=3x3, channels=16, stride=1, keeprate=0.6) × 2
	G(kernel=3x3, filters=32, stride=2, keeprate=0.6) × 3
	F(in=4096, out=512)
	F(in=512, out=256)
	Out(in=128, out=#features)

4.1 Results

Our main results can be seen in Figure 4.

We verified that similarity learning is an easier task than classification. The accuracy we achieve for this task is much higher than we achieve for classification. We confirmed previous work that triplet-based similarity learning got better results than pair-based similarity learning, and we also found that the resulting embeddings performed better for the task of classification. Experiments for pairs on CIFAR-10 and CIFAR-100, since pairs had significantly worse results for triplets on MNIST and Fashion MNIST, were not run.

We found that embedding spaces generated by similarity learning on category-level data got comparable accuracy. This implies that the embedding space does encode information relevant to solving other tasks, such as classification.

We also found that doubling the dimensions of the embedding space was not associated with an increase in accuracy, but rather often a slight decrease. Since we want the embeddings to be class-independent and thus scalable, this is a desirable feature. The slight decrease in accuracy is likely due to overfitting.

We found that using a two layer neural network rather than a linear classifier only resulted in minor gains in accuracy. These metrics can be seen in Figure 5. This is desirable because we want subsequent tasks that use these embeddings to be as simple as possible. One of our goals for the embedding was that distinct classes would be linearly separable.

We found disappointing preliminary results for the CIFAR-100 embedding when trained on both the coarse and fine labels. We speculate that this is because the network did not yet converge. In general, our results for CIFAR-100 were very poor. We believe that these needed to be trained for more iterations, which we did not have the computing power to do.

4.2 Visualizations

To give a better understanding of the embeddings generated by the baseline classifier and feature extractors, we give visualizations the embeddings using t-SNE [35, 42], a well known technique for visualizing data. These can be seen in Figures 6, 7, and 8.

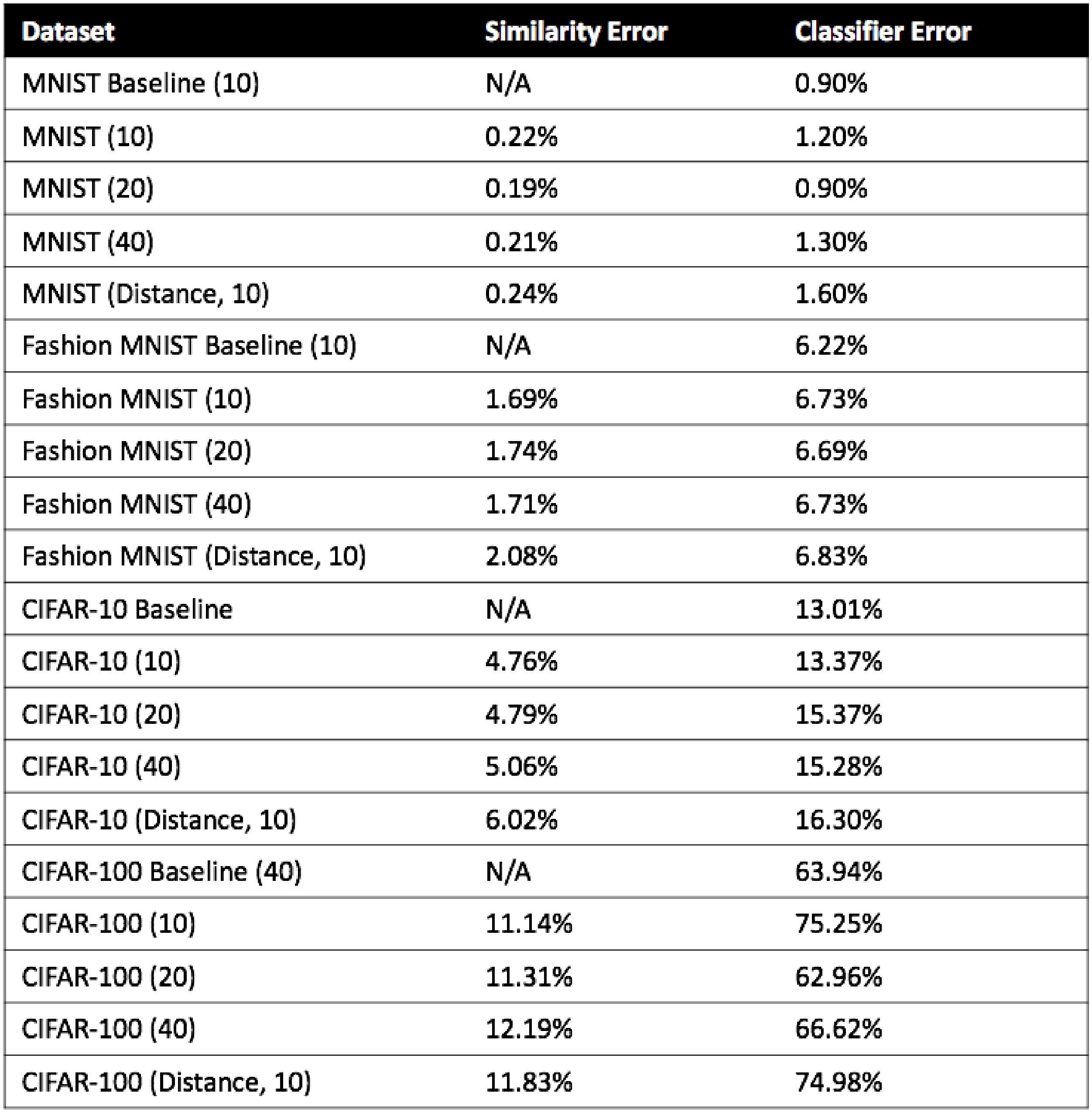
5 Conclusion

Future work should further explore the benefit of coarse labels for embeddings on CIFAR-100. It should also extend the experiment to ImageNet, for which we have more complex between-class information in the form of hyponym-hypernym relationships.

It would also be worth considering image-specific similarity data, rather than similarity data obtained from classification data, to obtain a more informative embedding space. One could compare embeddings generated by image-specific similarity data to embeddings generated by mapping images to semantic embeddings. Image-specific similarity data could be obtained manually, but could also utilize available data. For example, the Tiny Images dataset, obtained from image sources, could be a good source of similarity data. Similarity data could also be obtained by marking images that are located on the same page as similar, as with word embeddings.

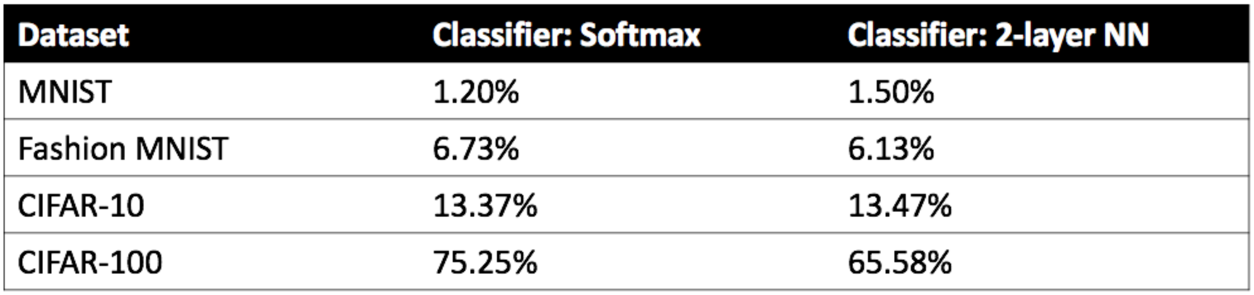
Lastly, it is also worthwhile to evaluate the embedding space on additional tasks. We only evaluated the embedding on classification.

In this thesis, we defined the task of generating a similarity-induced embedding and argued why it would be useful. We demonstrated the usefulness of this embedding task. To do so, we generated an embedding using academic baselines and evaluated the embedding on the task of classification. We achieved results comparable to classifiers with the same architecture trained directly for the task.



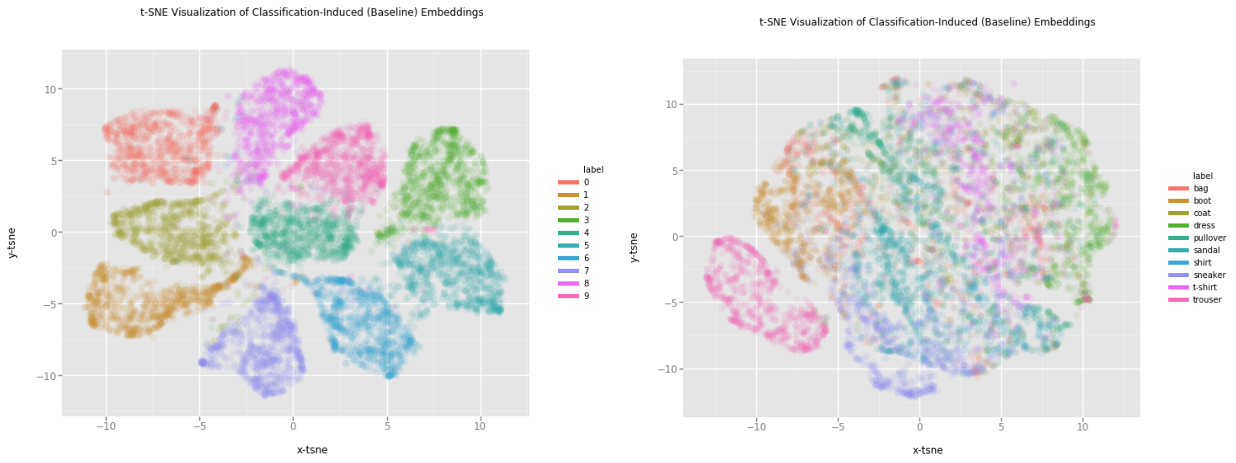
Dataset	Similarity Error	Classifier Error
MNIST Baseline (10)	N/A	0.90%
MNIST (10)	0.22%	1.20%
MNIST (20)	0.19%	0.90%
MNIST (40)	0.21%	1.30%
MNIST (Distance, 10)	0.24%	1.60%
Fashion MNIST Baseline (10)	N/A	6.22%
Fashion MNIST (10)	1.69%	6.73%
Fashion MNIST (20)	1.74%	6.69%
Fashion MNIST (40)	1.71%	6.73%
Fashion MNIST (Distance, 10)	2.08%	6.83%
CIFAR-10 Baseline	N/A	13.01%
CIFAR-10 (10)	4.76%	13.37%
CIFAR-10 (20)	4.79%	15.37%
CIFAR-10 (40)	5.06%	15.28%
CIFAR-10 (Distance, 10)	6.02%	16.30%
CIFAR-100 Baseline (40)	N/A	63.94%
CIFAR-100 (10)	11.14%	75.25%
CIFAR-100 (20)	11.31%	62.96%
CIFAR-100 (40)	12.19%	66.62%
CIFAR-100 (Distance, 10)	11.83%	74.98%

Figure 4: Results of experiments: **similarity and classification error**. In parentheses are the dimensions of features used, and whether it used Euclidean distance as a metric to learn features.



Dataset	Classifier: Softmax	Classifier: 2-layer NN
MNIST	1.20%	1.50%
Fashion MNIST	6.73%	6.13%
CIFAR-10	13.37%	13.47%
CIFAR-100	75.25%	65.58%

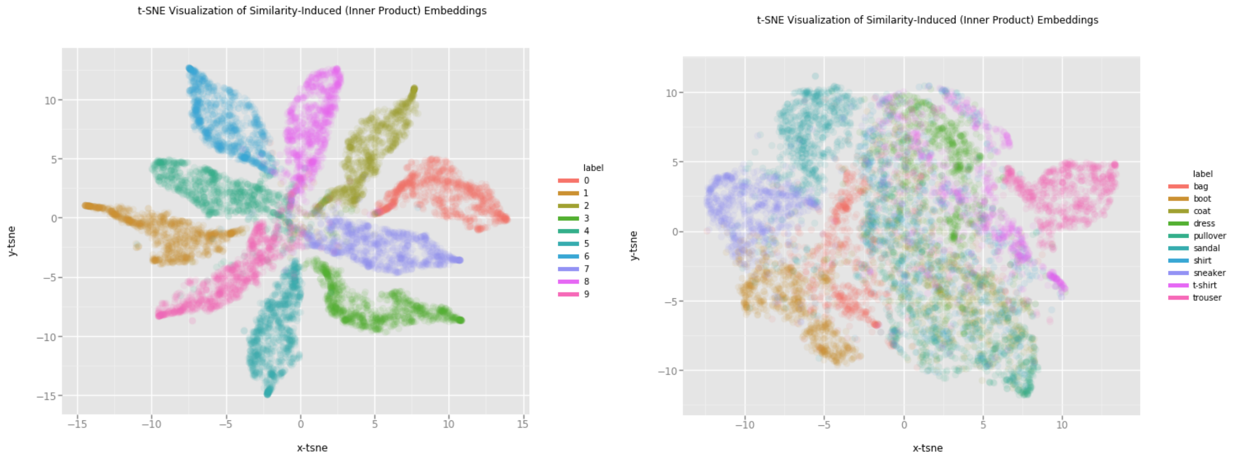
Figure 5: Comparing **classification error** when using a linear classifier and a two-layer neural net on the embeddings when the dimension of the embedding space is 10.



(a) MNIST

(b) Fashion MNIST

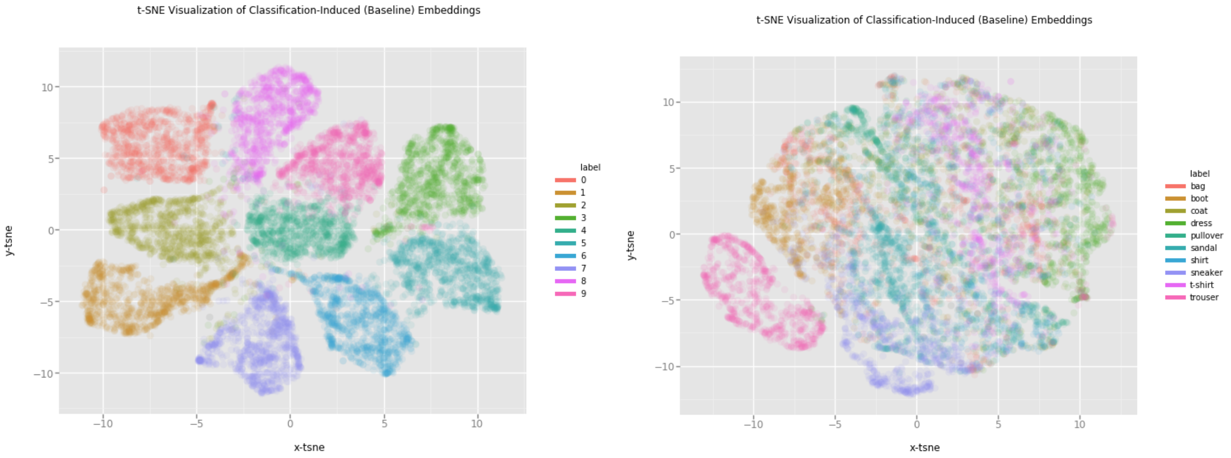
Figure 6: t-SNE Visualizations of Classification-Induced (Baseline) Embeddings



(a) MNIST

(b) Fashion MNIST

Figure 7: t-SNE Visualizations of Similarity-Induced (Inner Product) Embeddings



(a) MNIST

(b) Fashion MNIST

Figure 8: t-SNE Visualizations of Similarity-Induced (Euclidean Distance) Embeddings

References

[1] How many words are there in english? [Online; retrieved 12-May-2018].

[2] Vector representations of words, April 2018. [Online].

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

[5] Xinlei Chen and C Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431, 2015.

[6] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10(Mar):747–776, 2009.

[7] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

[8] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[9] Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*, 2013.

[10] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1777–1784. IEEE, 2011.

[11] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[12] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, et al. From captions to visual concepts and back. 2015.

[13] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[14] Christiane Fellbaum. *WordNet: An Electronic Lexical Database.* Bradford Books, 1998.

[15] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[16] Andrea Frome, Yoram Singer, and Jitendra Malik. Image retrieval and classification using local distance functions. In *Advances in neural information processing systems*, pages 417–424, 2007.

[17] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[18] Noa Garcia and George Vogiatzis. Learning non-metric visual similarity for image retrieval. *CoRR*, abs/1709.01353, 2017.

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[20] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[21] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

[22] Douwe Kiela and Léon Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45, 2014.

[23] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[24] Ryan Kiros, Richard Zemel, and Ruslan R Salakhutdinov. A multiplicative model for learning distributed text-based attribute representations. In *Advances in neural information processing systems*, pages 2348–2356, 2014.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[26] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10.

[27] Brian Kulis, Prateek Jain, and Kristen Grauman. Fast similarity search for learned metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2143–2157, 2009.

[28] Rémi Lajugie, Francis Bach, and Sylvain Arlot. Large-margin metric learning for constrained partitioning problems. In *International Conference on Machine Learning*, pages 297–305, 2014.

[29] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[30] Marc T Law, Yaoliang Yu, Matthieu Cord, and Eric P Xing. Closed-form training of mahalanobis distance for supervised clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3909–3917, 2016.

[31] Marc T Law, Yaoliang Yu, Raquel Urtasun, Richard S Zemel, and Eric P Xing. Efficient multiple instance metric learning using weakly supervised data. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[33] Rémi Lebret, Pedro O Pinheiro, and Ronan Collobert. Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*, 2015.

[34] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

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

[36] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[37] Brian McFee and Gert R Lanckriet. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 775–782, 2010.

[38] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012.

[39] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[40] 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.

[41] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[44] Karl Ridgeway, Jake Snell, Brett Roads, Richard S Zemel, and Michael C Mozer. Learning to generate images with perceptual similarity metrics. *arXiv preprint arXiv:1511.06409*, 2015.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[46] Yoram Singer. Lecture notes in cos324 on similarity learning, November 2017.

[47] Richard Socher. Cs224n: Natural language processing with deep learning: Lectures 1-3, January 2018. [Online].

[48] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[49] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[50] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association of Computational Linguistics*, 2(1):207–218, 2014.

[51] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[52] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

[53] Eleni Triantafillou, Jamie Ryan Kiros, Raquel Urtasun, and Richard Zemel. Towards generalizable sentence embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 239–248, 2016.

[54] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[55] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.

[56] Jiang Wang, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, Ying Wu, et al. Learning fine-grained image similarity with deep ranking. *arXiv preprint arXiv:1404.4661*, 2014.

[57] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[58] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[59] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

[60] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, volume 11, pages 2764–2770, 2011.

[61] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[62] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

[63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.