DEEP METRIC LEARNING USING TRIPLET NETWORK

Elad Hoffer
Department of Electrical Engineering
Technion Israel Institute of Technology
ehoffer@tx.technion.ac.il

Nir Ailon
Department of Computer Science
Technion Israel Institute of Technology
nailon@cs.technion.ac.il

ABSTRACT

Deep learning has proven itself as a successful set of models for learning useful semantic representations of data. These, however, are mostly implicitly learned as part of a classification task. In this paper we propose the *triplet network* model, which aims to learn useful representations by distance comparisons. A similar model was defined by Wang et al. (2014), tailored made for learning a ranking for image information retrieval. Here we demonstrate using various datasets that our model learns a better representation than that of its immediate competitor, the Siamese network. We also discuss future possible usage as a framework for unsupervised learning.

1 INTRODUCTION

For the past few years, deep learning models have been used extensively to solve various machine learning tasks. One of the underlying assumptions is that deep, hierarchical models such as convolutional networks create useful representation of data (Bengio (2009); Hinton (2007)), which can then be used to distinguish between available classes. This quality is in contrast with traditional approaches requiring engineered features extracted from data and then used in separate learning schemes. Features extracted by deep networks were also shown to provide useful representation (Zeiler & Fergus (2013a); Sermanet et al. (2013)) which can be, in turn, successfully used for other tasks (Razavian et al. (2014)).

Despite their importance, these representations and their corresponding induced metrics are often treated as side effects of the classification task, rather than being explicitly sought. There are also many interesting open question regarding the intermediate representations and their role in disentangling and explaining the data (Bengio (2013)). Notable exceptions where explicit metric learning is preformed are the *Siamese Network* variants (Bromley et al. (1993); Chopra et al. (2005); Hadsell et al. (2006)), in which a contrastive loss over the metric induced by the representation is used to train the network to distinguish between similar and dissimilar *pairs* of examples. A contrastive loss favours a small distance between pairs of examples labeled as similar, and large distances for pairs labeled dissimilar. However, the representations learned by these models provide sub-par results when used as features for classification, compared with other deep learning models including ours. Siamese networks are also sensitive to calibration in the sense that the notion of similarity vs dissimilarity requires context. For example, a person might be deemed similar to another person when a dataset of random objects is provided, but might be deemed dissimilar with respect to the same other person when we wish to distinguish between two individuals in a set of individuals only. In our model, such a calibration is not required. In fact, in our experiments here, we have experienced hands on the difficulty in using Siamese networks.

We follow a similar task to that of Chechik et al. (2010). For a set of samples $\mathbb{P}$ and a chosen rough similarity measure $r(x, x')$ given through a training oracle (e.g how close are two images of objects semantically) we wish to learn a similarity function $S(x, x')$ induced by a normed metric. Unlike

Chechik et al. (2010)'s work, our labels are of the form $r(x, x_1) > r(x, x_2)$ for triplets x, x_1, x_2 of objects. Accordingly, we try to fit a metric embedding and a corresponding similarity function satisfying:

$$S(x, x_1) > S(x, x_2), \ \ \forall x, x_1, x_2 \in \mathbb{P} \ \ \text{for which } r(x, x_1) > r(x, x_2).$$

In our experiment, we try to find a metric embedding of a multi-class labeled dataset. We will always take x_1 to be of the same class as x and x_2 of a different class, although in general more complicated choices could be made. Accordingly, we will use the notation x^+ and x^- instead of x_1, x_2. We focus on finding an L_2 embedding, by learning a function $F(x)$ for which $S(x, x') = \|F(x) - F(x')\|_2$. Inspired from the recent success of deep learning, we will use a deep network as our embedding function $F(x)$.

We call our approach a *triplet network*. A similar approach was proposed in Wang et al. (2014) for the purpose of learning a ranking function for image retrieval. Compared with the single application proposed in Wang et al. (2014), we make a comprehensive study of the triplet architecture which is, as we shall argue below, interesting in and of itself. In fact, we shall demonstrate below that the triplet approach is a strong competitor to the Siamese approach, its most obvious competitor.

2 THE TRIPLET NETWORK

A *Triplet network* (inspired by "Siamese network") is comprised of 3 instances of the same feed-forward network (with shared parameters). When fed with 3 samples, the network outputs 2 intermediate values - the L_2 distances between the embedded representation of two of its inputs from the representation of the third. If we will denote the 3 inputs as x, x^+ and x^-, and the embedded representation of the network as $Net(x)$, the one before last layer will be the vector:

$$TripletNet(x, x^-, x^+) = \begin{bmatrix} \|Net(x) - Net(x^-)\|_2 \\ \|Net(x) - Net(x^+)\|_2 \end{bmatrix} \in \mathbb{R}^2_+ \ .$$

In words, this encodes the pair of distances between each of x^+ and x^- against the *reference* x.

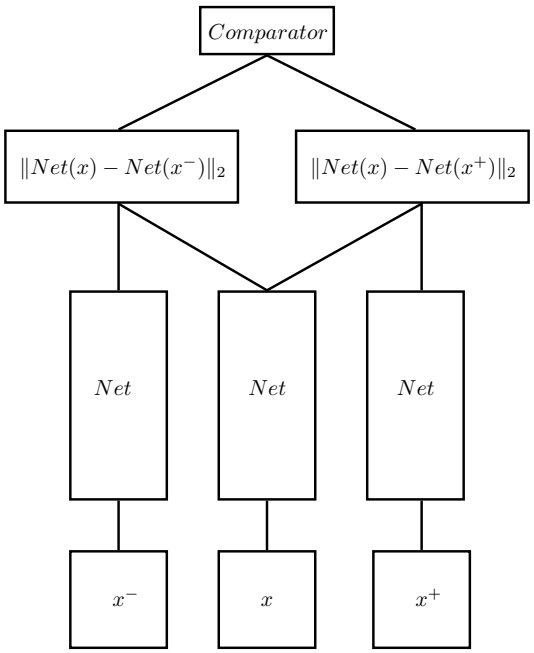

Figure 1: Triplet network structure

2.1 TRAINING

Training is preformed by feeding the network with samples where, as explained above, x and x^+ are of the same class, and x^- is of different class. The network architecture allows the task to be expressed as a 2-class classification problem, where the objective is to correctly classify which of x^+ and x^- is of the same class as x. We stress that in a more general setting, where the objective might be to learn a metric embedding, the label determines which example is *closer* to x. Here we simply interpret "closeness" as "sharing the same label". In order to output a comparison operator from the model, a SoftMax function is applied on both outputs - effectively creating a ratio measure. Similarly to traditional convolutional-networks, training is done by simple SGD on a negative-log-likelihood loss with regard to the 2-class problem. We later examined that better results are achieved when the loss function is replaced by a simple MSE on the soft-max result, compared to the $(0, 1)$ vector, so that the loss is

$$Loss(d_+, d_-) = \|(d_+, d_- - 1)\|_2^2 = const \cdot d_+^2$$

where

$$d_+ = \frac{e^{\|Net(x)-Net(x^+)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}}$$

and

$$d_- = \frac{e^{\|Net(x)-Net(x^-)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}} \ .$$

We note that $Loss(d_+, d_-) \rightarrow 0$ iff $\frac{\|Net(x)-Net(x^+)\|}{\|Net(x)-Net(x^-)\|} \rightarrow 0$, which is the required objective. By using the same shared parameters network, we allow the back-propagation algorithm to update the model with regard to all three samples simultaneously.

3 TESTS AND RESULTS

The Triplet network was implemented and trained using the Torch7 environment (Collobert et al. (2011)).

3.1 DATASETS

We experimented with 4 datasets. The first is *Cifar10* (Krizhevsky & Hinton (2009)), consisting of 60000 32x32 color images of 10 classes (of which 50000 are used for training only, and 10000 for test only). The second dataset is the original *MNIST* (LeCun et al. (1998)) consisting of 60000 28x28 gray-scale images of handwritten digits 0-9, and a corresponding set of 10000 test images. The third is the *Street-View-House-Numbers (SVHN)* of Netzer et al. consisting of 600000 32x32 color images of house-number digits 0-9. The fourth dataset is *STL10* of Coates et al. (2011), similar to Cifar10 and consisting of 10 object classes, only with 5000 training images (instead of 50000 in Cifar) and a bigger 96x96 image size.

It is important to note that no data augmentation or whitening was applied, and the only preprocessing was a global normalization to zero mean and unit variance. Each training instance (for all four datasets) was a uniformly sampled set of 3 images, 2 of which are of the same class (x and x^+), and the third (x^-) of a different class. Each training epoch consisted of 640000 such instances (randomly chosen each epoch), and a fixed set of 64000 instances used for test. We emphasize that each test instance involves 3 images from the set of test images which was excluded from training.

3.2 THE EMBEDDING NET

For Cifar10 and SVHN we used a convolutional network, consisting of 3 convolutional and 2x2 max-pooling layers, followed by a fourth convolutional layer. A *ReLU* non-linearity is applied between two consecutive layers. Network configuration (ordered from input to output) consists of filter sizes {5,3,3,2}, and feature map dimensions {3,64,128,256,128} where a 128 vector is the final embedded representation of the network. Usually in convolutional networks, a subsequent fully-connected layer is used for classification. In our net this layer is removed, as we are interested in a feature embedding only.

The network for STL10 is identical, only with stride=3 for the first layer, to allow the bigger input size. The network used for MNIST was a smaller version consisting of smaller feature map sizes $\{1,32,64,128\}$.

3.3 RESULTS

Training on all datasets was done by SGD, with initial learning-rate of 0.5 and a learning rate decay regime. We used a momentum value of 0.9. We also used the dropout regularization technique with $p = 0.5$ to avoid over-fitting. After training on each dataset for 10-30 epochs, the network reached a fixed error over the triplet comparisons. We then used the embedding network to extract features from the full dataset, and trained a simple 1-layer network model on the full 10-class classification task (using only training set representations). The test set was then measured for accuracy. These results (Figure 2) are comparable to state-of-the-art results with deep learning models, without using any artificial data augmentation (Zeiler & Fergus (2013b); Goodfellow et al. (2013); Lin et al. (2013)). Noteworthy is the STL10 dataset, in which the TripletNet achieved the best known result for non-augmented data. We conjecture that data augmentation techniques (such as translations, mirroring and noising) may provide similar benefits to those described in previous works.

We also note that similar results are achieved when the embedded representations are classified using a linear SVM model or KNN classification with up to 0.5% deviance from the results in Figure 2. Another side-affect noticed, is that the representation seems to be sparse - about 25% non-zero values. This is very helpful when used later as features for classification both computationally and with respect to accuracy, as each class is characterised by only a few non zero elements.

Dataset	TripletNet	SiameseNet	Best known result (with no data augmentation)
Mnist	99.54%$\pm$0.08	97.9%$\pm$0.1	99.61% Mairal et al. (2014); Lee et al. (2014)
Cifar10	87.1%	-	90.22% Lee et al. (2014)
SVHN	95.37%	-	98.18% Lee et al. (2014)
STL10	70.67%	-	67.9% Lin & Kung (2014)

Figure 2: Classification accuracy (no data augmentation)

3.4 2D VISUALIZATION OF FEATURES

In order to examine our main premise, which is that the network embeds the images into a representation with meaningful properties, we use PCA to project the embedding into 2d euclidean space which can be easily visualized (figures 5 4 5). We can see a significant clustering by semantic meaning, confirming that the network is useful in embedding images into the euclidean space according to their content. Similarity between objects can be easily found by measuring the distance between their embedding and, as shown in the results, can reach high classification accuracy using a simple subsequent linear classifier.

3.5 COMPARISON WITH PERFORMANCE OF THE SIAMESE NETWORK

The Siamese network is the most obvious competitor for our approach. Our implementation of the Siamese network consisted of the same embedding network, but with the use of a contrastive loss between a pair of samples, instead of three (as explained in Chopra et al. (2005)). The generated features were then used for classification using a similar linear model as was used for the TripletNet method. We measured lower accuracy on the MNIST dataset compared to results gained using the TripletNet representations 2.

We have tried a similar comparison for the other three datasets, but unfortunately could not obtain any meaningful result using a Siamese network. We conjecture that this might be related to the problem of context described above, and leave the resolution of this conjecture to future work.

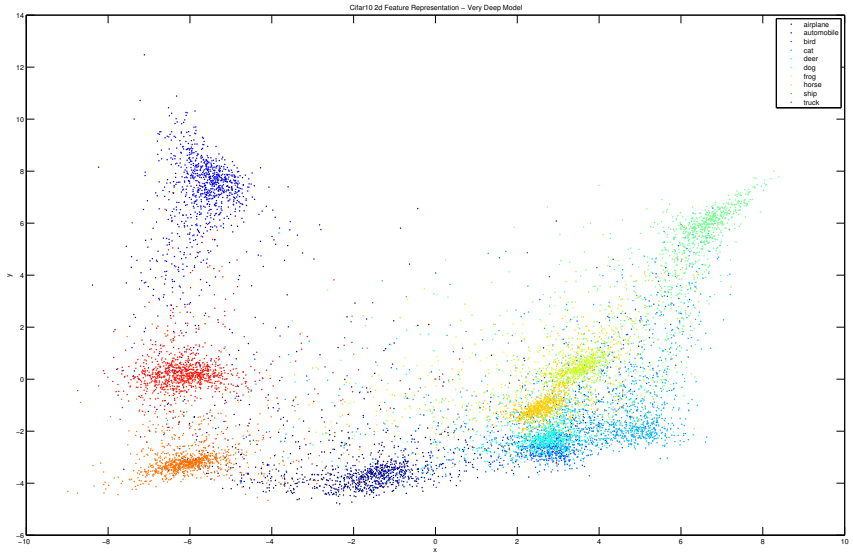

Figure 3: CIFAR10 - Euclidean representation of embedded test data, projected onto top two singular vectors

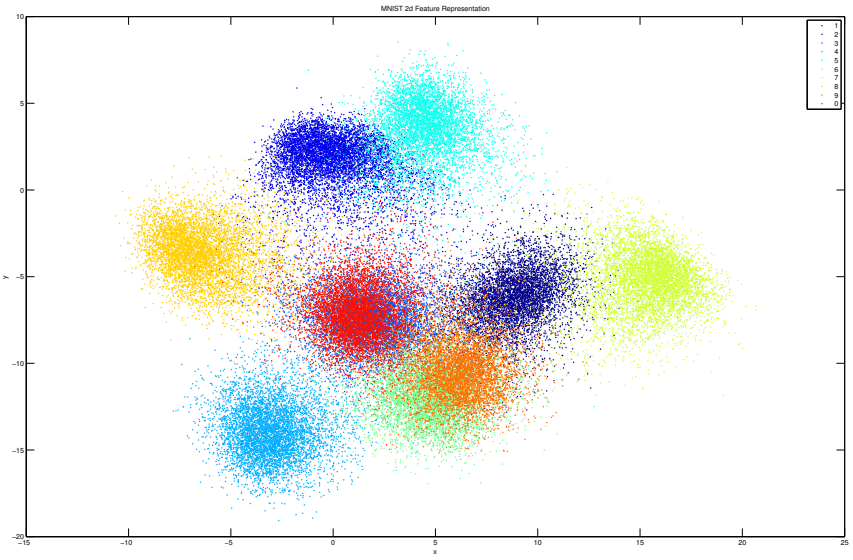

Figure 4: MNIST - Euclidean representation of embedded test data, projected onto top two singular vectors

4 FUTURE WORK

As the Triplet net model allows learning by comparisons of samples instead of direct data labels, usage as an unsupervised learning model is possible. Future investigations can be performed in several scenarios:

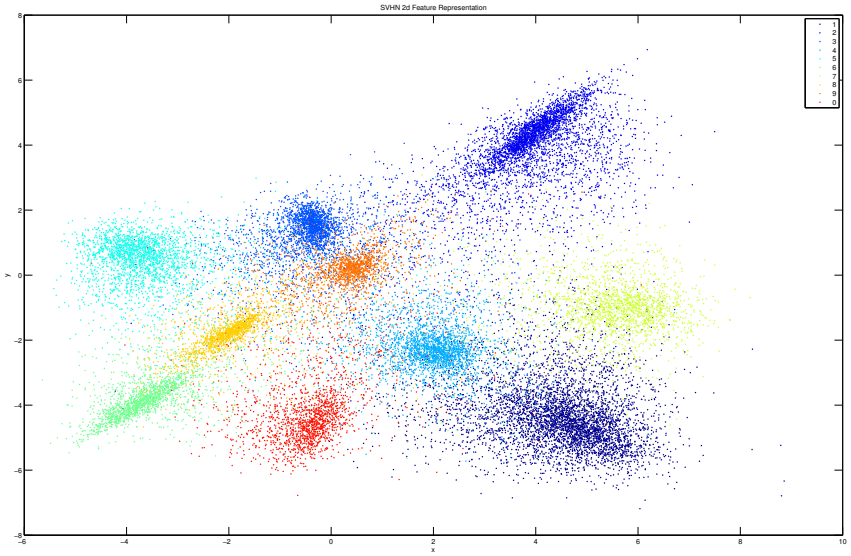

Figure 5: SVHN - Euclidean representation of embedded test data, projected onto top two singular vectors

- **Using spatial information.** Objects and image patches that are spatially near are also expected to be similar from a semantic perspective. Therefore, we could use geometric distance between patches of the same image as a rough similarity oracle $r(x, x')$, in an unsupervised setting.
- **Using temporal information.** The same is applicable to time domain, where two consecutive video frames are expected to describe the same object, while a frame taken 10 minutes later is less likely to do so. Our Triplet net may provide a better embedding and improve on past attempts in solving classification tasks in an unsupervised environment, such as that of (Mobahi et al. (2009)).

It is also well known that humans tend to be better at accurately providing comparative labels. Our framework can be used in a crowd sourcing learning environment. This can be compared with Tamuz et al. (2011), who used a different approach. Furthermore, it may be easier to collect data trainable on a Triplet network, as comparisons over similarity measures are much easier to attain (pictures taken at the same location, shared annotations, etc).

5 Conclusions

In this work we introduced the *Triplet network* model, a tool that uses a deep network to learn useful representation explicitly. The results shown on various datasets provide evidence that the representations that were learned are useful to classification in a way that is comparable with a network that was trained explicitly to classify samples. We believe that enhancement to the embedding network such as Network-in-Network model (Lin et al. (2013)), Inception models (Szegedy et al. (2014)) and others can benefit the Triplet net similarly to the way they benefited other classification tasks. Considering the fact that this method requires to know only that two out of three images are sampled from the same class, rather than knowing what that class is, we think this should be inquired further, and may provide us insights to the way deep networks learn in general. We have also shown how this model learns using only comparative measures instead of labels, which we can use in the future to leverage new data sources for which clear out labels are not known or do not make sense (e.g hierarchical labels).

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan-Z GPU used for this research.

REFERENCES

Bengio, Yoshua. Learning Deep Architectures for AI, 2009. ISSN 1935-8237.

Bengio, Yoshua. Deep learning of representations: Looking forward. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7978 LNAI, pp. 1–37, 2013. ISBN 9783642395925. doi: 10.1007/978-3-642-39593-2_1.

Bromley, Jane, Bentz, James W, Bottou, Léon, Guyon, Isabelle, LeCun, Yann, Moore, Cliff, Säckinger, Eduard, and Shah, Roopak. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

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

Chopra, Sumit, Hadsell, Raia, and LeCun, Yann. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 539–546, 2005. ISBN 0769523722. doi: 10.1109/CVPR.2005.202.

Coates, Adam, Ng, Andrew Y, and Lee, Honglak. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

Collobert, Ronan, Kavukcuoglu, Koray, and Farabet, Clément. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Hadsell, Raia, Chopra, Sumit, and LeCun, Yann. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pp. 1735–1742. IEEE, 2006.

Hinton, Geoffrey E. Learning multiple layers of representation, 2007. ISSN 13646613.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, Chen-Yu, Xie, Saining, Gallagher, Patrick, Zhang, Zhengyou, and Tu, Zhuowen. Deeply-supervised nets. *arXiv preprint arXiv:1409.5185*, 2014.

Lin, Min, Chen, Qiang, and Yan, Shuicheng. Network in network. *CoRR*, abs/1312.4400, 2013. URL `http://arxiv.org/abs/1312.4400`.

Lin, Tsung-Han and Kung, HT. Stable and efficient representation learning with nonnegativity constraints. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1323–1331, 2014.

Mairal, Julien, Koniusz, Piotr, Harchaoui, Zaid, and Schmid, Cordelia. Convolutional kernel networks. In *Advances in Neural Information Processing Systems*, pp. 2627–2635, 2014.

Mobahi, Hossein, Collobert, Ronan, and Weston, Jason. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 737–744. ACM, 2009.

Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. Reading digits in natural images with unsupervised feature learning.

Razavian, Ali Sharif, Azizpour, Hossein, Sullivan, Josephine, and Carlsson, Stefan. CNN Features off-the-shelf: an Astounding Baseline for Recognition. *Arxiv*, 2014. URL http://arxiv.org/abs/1403.6382.

Sermanet, Pierre, Eigen, David, Zhang, Xiang, Mathieu, Michael, Fergus, Rob, and LeCun, Yann. OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. *arXiv preprint arXiv:1312.6229*, pp. 1–15, 2013. URL http://arxiv.org/abs/1312.6229.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

Tamuz, Omer, Liu, Ce, Belongie, Serge, Shamir, Ohad, and Kalai, Adam. Adaptively learning the crowd kernel. In Getoor, Lise and Scheffer, Tobias (eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pp. 673–680, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Wang, Jiang, Song, Yang, Leung, Thomas, Rosenberg, Chuck, Wang, Jingbin, Philbin, James, Chen, Bo, and Wu, Ying. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

Zeiler, Matthew D and Fergus, Rob. Visualizing and Understanding Convolutional Networks. *arXiv preprint arXiv:1311.2901*, 2013a. URL http://arxiv.org/abs/1311.2901.

Zeiler, Matthew D and Fergus, Rob. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013b.