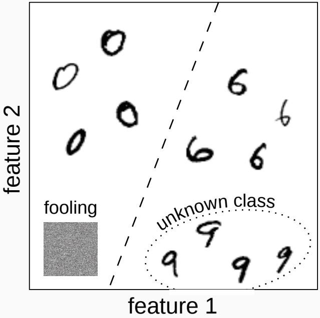
Denoising Autoencoders for Overgeneralization in Neural Networks

Giacomo Spigler

The Biorobotics Institute, Scuola Superiore Sant'Anna

Overgeneralization and "Fooling"

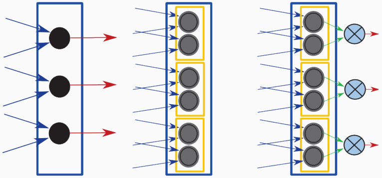
- Overgeneralization: classifying inputs not belonging to any training class as one of the training classes
  - Open set recognition: training on a *limited* number of classes, testing on a *larger* number of classes
- Fooling [Nguyen et al., 2015]: inputs that are unrecognizable to humans get classified as one of the training classes with high confidence



Previous Work

- Positive vs negative training samples
- Threshold on the outputs of a classifier
- Confidence score based on k-Nearest-Neighbor
- Open set recognition: 1-vs-Set Machine, Weibull-SVM, OpenMax
  - Special case '1-class recognition': 1-class SVM.

- COOL (**C**ompetitive **O**vercomplete **O**utput **L**ayer: each output unit is replaced with ω ones competing with oneanother via a softmax activation. Confidence score = product of the output of all ω units for the same class.



(a) Conventional output layer (b) COOL during training (c) COOL during testing

Image from [Kardan and Stanley, 2017]

Proposed Solution

Proposed Solution – Motivation

- Identify data points that belong to the data distribution $p(x)$
- Problem: $p(\mathbf{x})$ is hard to model!
- Solution: it may suffice to identify points that are close to the local maxima of the data distribution

- [Bengio et al., 2013] and [Alain and Bengio, 2014] showed that denosing and contractive autoencoders implicitly learn aspects of the underlying data distribution. Specifically, their reconstruction error approximates the gradient of the log-density of the data

$$\frac{\partial \log p(\mathbf{x})}{\partial \mathbf{x}} \propto r(\mathbf{x}) - \mathbf{x}$$

for small corruption noise $\sigma \to 0$, $r(\mathbf{x}) = Dec(Enc(\mathbf{x}))$.

Proposed Solution – Confidence Score

- Critical points of $p(\mathbf{x}) \Leftrightarrow$ small gradient of the log-density $\Leftrightarrow$ small reconstruction error

Why

Those are points that the network can reconstruct well, and that it has thus hopefully experienced during training, or has managed to generalize to in a good way.

- We can use this insight to design a confidence score for the data points. For example,

$$\tilde{c}(\mathbf{x}) = \exp\left(-\frac{\alpha}{D}\|r(\mathbf{x}) - \mathbf{x}\|_2\right)$$

$\mathbf{x} \in \mathbb{R}^D$, α controls the sensitivity of the function to outliers

Proposed Solution – Local Maxima of $p(\mathbf{x})$

Problem

This approach cannot discriminate between local minima, maxima or saddle points, and may thus assign a high confidence score to points not belonging to the target distribution.

- Solution: approximate the Hessian of the log-density from the Jacobian of the reconstruction function [Alain and Bengio, 2014]

$$\frac{\partial^2 \log p(\mathbf{x})}{\partial \mathbf{x}^2} \propto \frac{\partial r(\mathbf{x})}{\partial \mathbf{x}} - I$$

- Then scale the computed confidence by a function $\Gamma(\mathbf{x})$ that favours small or negative curvature of the log-density. Here we propose

$$\Gamma(\mathbf{x}) = \begin{cases} 1 & \text{if } \gamma(\mathbf{x}) \leq 0 \\ \exp(-\beta\gamma(\mathbf{x})) & \text{if } \gamma(\mathbf{x}) > 0 \end{cases}$$

$$\gamma(\mathbf{x}) = \frac{1}{D} \sum_i \left(\frac{\partial r_i(\mathbf{x})}{\partial x_i} - 1 \right)$$

Proposed Solution

- The confidence score can be then modified as

$$\tilde{c}(\mathbf{x}) = \exp\left(-\frac{\alpha}{D}\|r(\mathbf{x}) - \mathbf{x}\|_2\right) \Gamma(\mathbf{x})$$

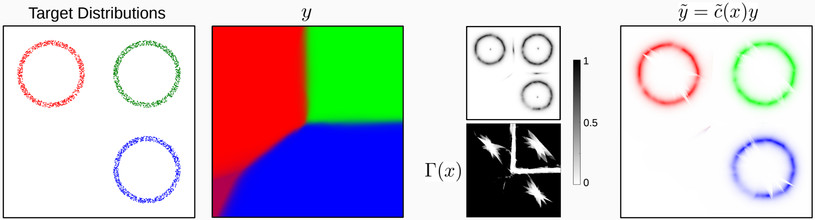
- The score is high for small reconstruction errors, that is for points within regions of small gradient of the log-density of the data. $\Gamma(\mathbf{x})$ further selects regions with small or negative curvature, restricting high values of $\tilde{c}(\mathbf{x})$ only near its maxima
- A classifier can be modified by scaling its predicted outputs by $\tilde{c}(\mathbf{x})$

$$\tilde{\mathbf{y}} = \tilde{c}(\mathbf{x})\mathbf{y}$$

Results

Results – 2D Toy Problem

- 3 target classes (rings with $thickness = 0.1$, $r_{inner} = 0.6$, $centers = \{(-1, 1), (1, 1), (1, -1)\}$)
- Predictions of a classifier y over the whole input space, along with confidence scores and scaled outputs $\tilde{y}$



Results – Fooling 1

- Fooling Generator Network (FGN): input samples produced from a single feedforward layer with sigmoid activation and random (fixed) input (e.g., for MNIST, single layer with 784 inputs and 784 outputs)
- Fooling is attempted by gradient descent on the parameters of the FGN to minimize the cross-entropy between the output of the network to be fooled and the desired target output class

- Network architectures for all the results:
  - Baseline: $\{Conv2D(1 \rightarrow 32, 5 \times 5),\ Max(2 \times 2),\ ReLU,\ Conv2D(32 \rightarrow 64, 5 \times 5),\ Max(2{\times}2),\ ReLU,\ FC(64 \rightarrow 400),\ ReLU,\ FC(400 \rightarrow 10),\ softmax\}$
  - COOL: same, with $10 \times \omega$ outputs
  - dAE (ours): same, with a symmetric decoder attached to the last hidden layer

Results – Fooling 2

Table 1: MNIST

Model	Accuracy			Fooling Rate (Avg Steps)	
	0%	90%	99%	90%	99%
CNN	99.35%	99.23%	99%	100% (63.5)	99% (187.1)
COOL	99.33%	98.1%	93.54%	34.5% (238.8)	4.5% (313.4)
dAE sym	98.98%	98.11%	96.8%	**0% (-)**	**0% (-)**
dAE asym	99.14%	98.41%		**0% (-)**	

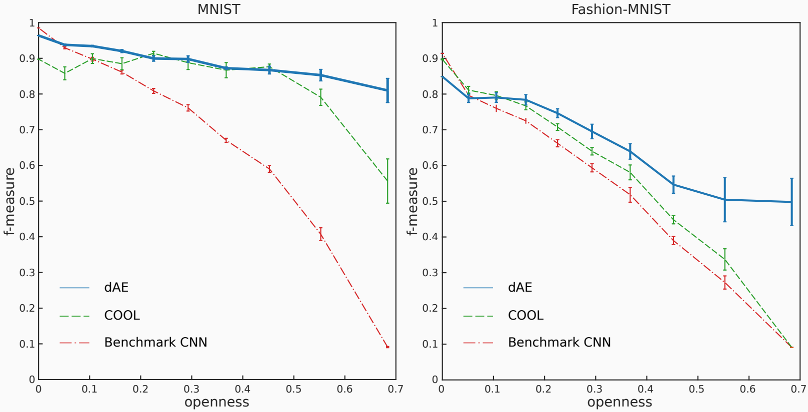
Table 2: Fashion-MNIST

Model	Accuracy			Fooling Rate (Avg Steps)	
	0%	90%	99%	90%	99%
CNN	91.65%	90.91%	89.27%	100% (113.0)	30.5% (902.0)
COOL	91.23%	87%	65.3%	**0%** (-)	**0%** (-)
dAE sym	91.59%	77.8%	64.87%	**0%** (-)	**0%** (-)

Results – Open Set Recognition

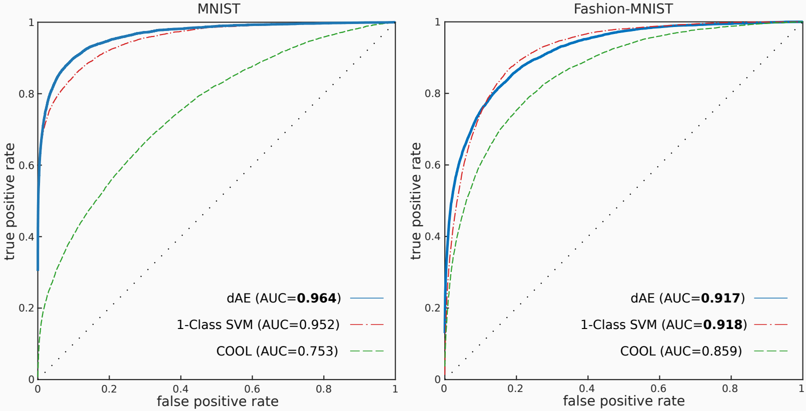
Threshold at 99% (MNIST) and 90% (Fashion-MNIST).

num_training_classes $\in \{1, 2, \ldots, 10\}$.



$$\text{openness} = 1 - \sqrt{\frac{\text{num\_training\_classes}}{\textit{num\_total\_classes}}} \qquad F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Results – 1-Class Recognition



Conclusions

Conclusions and Future Work

- Overgeneralization is a problem in discriminative models in machine learning
- *We proposed to use information about the data distribution implicitly learnt by denoising autoencoders to compute a confidence score for novel inputs*
- Applications in novelty and outliers detection
- Potential issues:
  - The dAE may not manage to learn a model of the data
  - Clutter and images with multiple objects
- Future work:
  - Recurrent attention model to deal with clutter (i.e., only reconstruct part of the input)
  - Replacing the dAE with an EBGAN discriminator

Questions?

Alain, G. and Bengio, Y. (2014).
What regularized auto-encoders learn from the data-generating distribution.
The Journal of Machine Learning Research, 15(1):3563–3593.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013).
Generalized denoising auto-encoders as generative models.
In Advances in Neural Information Processing Systems, pages 899–907.

Kardan, N. and Stanley, K. O. (2017).
Mitigating fooling with competitive overcomplete output layer neural networks.
In Neural Networks (IJCNN), 2017 International Joint Conference on, pages 518–525. IEEE.

Nguyen, A., Yosinski, J., and Clune, J. (2015).
Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.
In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 427–436.

Supplementary Slides – Clutter 1

- SVHN dataset for cluttered digit recognition
- Reconstructing parts of an image or individual objects may be easier than modelling all possible compositions of objects.
- ~~Reconstruction error low iff all image reconstructed~~ $\rightarrow$ only within an attention mask

$$\|\mathbf{a}\left(r(\mathbf{x}) - \mathbf{x}\right)\|_2$$

Supplementary Slides – Clutter 2

- Using a recurrent network to produce the attention mask yields an interesting result: the gradient of the log-density of the whole image is the sum of the gradients of the log-density for the relevant objects within, so the confidence score proposed here can be simply approximated with the sum of the masked reconstruction errors (over all objects / features in the input, minus clutter). For images composed by independent objects, the likelihood of the image can be approximated by the product of the likelihood of the objects

$$p(\mathbf{x}_{\text{whole}}) = \prod_i p(\mathbf{x}_{\text{object}_i})$$

$$\frac{\partial \log p(\mathbf{x}_{\text{whole}})}{\partial \mathbf{x}} = \sum_i \frac{\partial \log p(\mathbf{x}_{\text{object}_i})}{\partial \mathbf{x}} \approx \sum_i \mathbf{a}_i \left(r(\mathbf{x}) - \mathbf{x} \right)$$

- MSc student project?

Supplementary Slides – Open Set Recognition

- Fashion-MNIST, open set recognition, with classification threshold $\tau = 99\%$