ARE GENERATIVE CLASSIFIERS MORE ROBUST TO ADVERSARIAL ATTACKS?

Yingzhen Li
University of Cambridge, UK
yl494@cam.ac.uk

ABSTRACT

There is a rising interest in studying the robustness of deep neural network classifiers against adversaries, with both advanced attack and defence techniques being actively developed. However, most recent work focuses on *discriminative* classifiers which only models the conditional distribution of the labels given the inputs. In this abstract we propose *deep Bayes classifier* that improves the classical naive Bayes with deep generative models, and verifies its robustness against a number of existing attacks. Our initial results on MNIST suggest that deep Bayes classifiers might be more robust when compared with deep discriminative classifiers.

1 INTRODUCTION

Deep neural networks have been shown to be vulnerable to adversarial attacks Szegedy et al. (2013); Goodfellow et al. (2014). Since then, many researchers have proposed adversarial attack and defence mechanisms, and some notable developments include: Goodfellow et al. (2014); Moosavi Dezfooli et al. (2016); Papernot et al. (2016); Carlini & Wagner (2017a); Kurakin et al. (2016); Madry et al. (2018) for attacks, and Szegedy et al. (2013); Gu & Rigazio (2014); Grosse et al. (2017); Li & Gal (2017); Feinman et al. (2017); Louizos & Welling (2017); Song et al. (2018); Madry et al. (2018) for defences. These developments enable better understanding of the robustness of deep neural networks as *discriminative classifiers* against adversaries.

Surprisingly, much less recent work has investigated the robustness of *generative classifiers* against adversarial attacks for multi-class classification, where such classifiers explicitly model the conditional distribution of the inputs given labels. In formula, denote the random variables of the input and label as $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{y} \in \{\boldsymbol{y}_c | c = 1, ..., C\}$ where $\boldsymbol{y}_c$ denotes the one-hot encoding vector for class c. A generative classifier first builds a *conditional generative model* $p(\boldsymbol{x}|\boldsymbol{y})$, then, in prediction time, predicts the label of a test input $\boldsymbol{x}^*$ using Bayes' rule

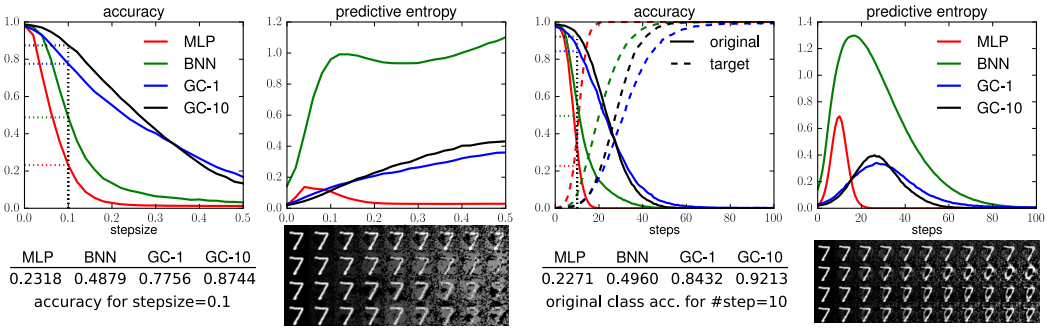
$$p(\boldsymbol{y}^*|\boldsymbol{x}^*) = \frac{p(\boldsymbol{x}^*|\boldsymbol{y}^*)p(\boldsymbol{y}^*)}{p(\boldsymbol{x}^*)}. \tag{1}$$

Perhaps the *naive Bayes* classifier is the most well-known generative classifier, which assumes a factorised distribution for the conditional generator, i.e. $p(\boldsymbol{x}|\boldsymbol{y}) = \prod_{d=1}^{D} p(x_d|\boldsymbol{y})$. However naive Bayes is less suitable for e.g. image and speech data, where the factorisation assumption is inappropriate. Fortunately, we can leverage the recent advances of generative modelling and apply a deep generative model for the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$. We refer to such generative classifiers that use deep generative models as *deep Bayes* classifiers.

As an example, we use a deep latent Gaussian model (Rezende et al., 2014) which reads

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y})p(\boldsymbol{z}), \quad p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I}), \quad p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}) = \prod_{d=1}^{D} p(\boldsymbol{x}_d|\boldsymbol{z}, \boldsymbol{y}), \tag{2}$$

where $p(\boldsymbol{x}_d|\boldsymbol{z}, \boldsymbol{y})$ can be Gaussian or Bernoulli distributions with parameters determined by a deep neural network talking both $\boldsymbol{z}$ and $\boldsymbol{y}$ as inputs. Importantly, this leads to a *non-factorised* conditional distribution $p(\boldsymbol{x}|\boldsymbol{y}) = \int_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y})p(\boldsymbol{z})d\boldsymbol{z}$. However this marginal likelihood is intractable, and instead we use the variational auto-encoder (VAE) algorithm (Kingma & Welling, 2013; Rezende et al., 2014) to train the conditional generative model, together with an inference network $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$



(a) Untargeted attack. (b) Targeted attack.

Figure 1: FGSM attacks on MNIST classifiers. The predictive entropy is defined by the entropy of the classifier's output probability vector. The BNN results are taken from Li & Gal (2017).

that is also conditioned on the label $\boldsymbol{y}$. After training the predicted class probability vector $\boldsymbol{y}^*$ for a future input $\boldsymbol{x}^*$ is (approximately) computed by Bayes' rule:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*) \approx \text{softmax}_{c=1}^{C}\left(\log \sum_{k=1}^{K} \frac{p(\boldsymbol{x}^*, \boldsymbol{z}_c^k, \boldsymbol{y}_c)}{q(\boldsymbol{z}_c^k|\boldsymbol{x}^*, \boldsymbol{y}_c)}\right), \quad \boldsymbol{z}_c^k \sim q(\boldsymbol{z}|\boldsymbol{x}^*, \boldsymbol{y}_c). \tag{3}$$

where $\text{softmax}_{c=1}^{C}$ denotes the softmax operator over the c axis. Therefore the output probability vector is computed in an analogous way to many deep discriminative classifiers that use softmax activation in the last layer, so that many existing attacks can be tested directly.

2 Initial Experimental Study

We carry out an initial test on the proposed generative classifier (GC) using VAE (3). All the attacks are taken from the CleverHans 2.0 library (Nicolas Papernot, 2017).

MLP experiments on MNIST. We follow Li & Gal (2017) and consider adversarial attacks on classifiers based on MLPs. Four models are tested: a normal discriminative classifier parameterised by an MLP, a Bayesian MLP network (BNN) trained with dropout rate 0.5 and tested with $K = 10$ times MC-dropout, and finally the deep Bayes classifier (trained with ℓ_2 loss) using $K = 1$ (GC-1) and $K = 10$ samples (GC-10), respectively.

We first consider the untargeted single-step FGSM attack (Goodfellow et al., 2014) and vary the stepsize between 0.0 and 0.5. In Figure 1(a) we show the classification accuracy on the adversarial examples and also the predictive entropy measure $\mathbb{H}[\boldsymbol{y}^*]$. It is clear that the deep Bayes classifiers are most robust, where increasing K also improves the test accuracy. The predictive entropy, used as a measure of uncertainty, also increases for the deep Bayes classifiers, which is as expected since the inputs are driven away from the data manifold.

We also apply the iterative version of targeted FGSM for 100 iterations with step-size 0.01. Results are shown in Figure 1(b). While this attack is more effective in terms of accuracy, again the deep Bayes models achieve the best robustness against it. Also, running this iterative attack produces a smooth interpolation between digits of the original and adversarial classes, and the predictive entropy of the classifier increases then decreases along the gradient descent path.

CNN experiments on MNIST. We also apply adversarial attacks to classifiers based on CNNs, and in this case we focus on the comparisons between discriminative classifiers and generative classifiers. The attack is the Carlini & Wagner ℓ_2 attack (CW-ℓ_2) (Carlini & Wagner, 2017a) with recommended parameters.[1] Since attacking generative classifiers take significantly longer computation time, in this initial experiment we sample 200 test images from the MNIST dataset, and craft

[1] https://github.com/carlini/nn_robust_attacks/blob/master/l2_attack.py

Table 1: Targeted CW-ℓ_2 attack on CNN-based models. Here error (adv) reports the error of the adversarial inputs to the original classes, and accuracy (adv) reports the accuracy to the target labels.

	accuracy (clean)	error (adv)	accuracy (adv)	distortion (adv)
CNN	100.00%	99.89%	99.89%	1.993
GC-1	98.77%	15.61%	2.65%	2.204
GC-10	99.15%	20.46%	5.86%	2.266

adversarial examples targeting the classes other than the ground-truth label. This results in 1,800 attempts in total. Results are reported in Table 1, where the average distortion in ℓ_2 distance is computed on the successful attacks. Presumably the best performance of GC-1 is due to the randomness of the classifier (3). However, we believe this randomness effect is largely removed in the GC-10 experiment, since no significant accuracy improvement is observed on clean inputs for $K > 10$. In summary, the results indicate that the deep Bayes classifier is significantly more robust to the CW-ℓ_2 attack than the CNN baseline.

Detecting adversarial attacks with conditional generative models. Given an input $\boldsymbol{x}$, a trained conditional generative model can produce a "reconstruction" $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y})$ of $\boldsymbol{x}$ conditioned on a given label $\boldsymbol{y}$ (by auto-encoding or optimisation). We conjecture that, if a classifier returns an incorrect label $\boldsymbol{y}^{\text{pred}} \neq \boldsymbol{y}^{\text{true}}$ on $\boldsymbol{x}$, then under some distance measure we can show that $d(\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}^{\text{true}})) < d(\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}^{\text{pred}}))$. Consequently, if an adversarial image of a cat is incorrectly labelled as "dog", then the "reconstructed" image will be close to an image of a dog, which is far away from the manifold of "cats" in an appropriately selected distance. We used ℓ_2 distance as the distance measure in the appendix experiments, and confirmed our conjecture on all the attacks tested.

3 Discussion

We have shown initial evidence that generative classifiers might be more robust to existing attacks than discriminative classifiers. The results are not conclusive as Carlini & Wagner (2017b) suggested that MNIST properties might not hold on e.g. Cifar-10. Future work will investigate deep Bayes classifiers based on auto-regressive models such as the PixelCNN (van den Oord et al., 2016b;a), and test deep Bayes classifiers on other natural image datasets such as Cifar-10 and SVHN.

Our positive results might be due to gradient masking (Papernot et al., 2017) and future work will investigate it in more detail. But we also note that many recent attacks are designed for discriminative classifiers, while many benchmark datasets have some anti-causal structures (Schölkopf et al., 2012). Consider MNIST as an example: a person first intends to write a digit ($\boldsymbol{y} \sim p_{\mathcal{D}}(\boldsymbol{y})$), then this intention causes a writing action producing an image of that digit ($\boldsymbol{x} \sim p_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{y})$). Therefore a deep Bayes classifier is more suitable to MNIST, and it will be more robust if the deep generative model is very powerful to approximate the data distribution $p_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{y})$.

We do not intend to claim that generative classifiers are robust to *all* possible attacks. Indeed, naive Bayes as a standard approach for spam filtering is fragile (Dalvi et al., 2004; Huang et al., 2011), and very recently Tabacof et al. (2016); Kos et al. (2017); Creswell et al. (2017) also designed attacks for (unconditional) VAE-type models. However, Dalvi et al. (2004) also showed that generative classifiers can be made more secure if aware of the attack strategy, and Biggio et al. (2011; 2014) further improved naive Bayes' robustness by modelling the conditional distribution of the adversarial inputs. This is similar to adversarial training of deep discriminative classifiers, and efficient ways for doing so with deep Bayes classifiers can be an interesting research direction.

In general, deep Bayes classifiers are less accurate than deep discriminative classifiers on classifying legitimate inputs. Also they are much more computationally expensive, limiting their applications to big neural networks and large-scale datasets such as the ImageNet. Still, a careful study of generative classifiers can inspire better designs of attack, defence and detection techniques for discriminative classifiers that use generative models as auxiliaries. Indeed Gu & Rigazio (2014); Song et al. (2018) proposed defence techniques by "purifying" adversarial inputs with auto-encoders/generative models, which moves the adversarial images towards the data manifold. Also the proposed detection method using conditional generative models has shown promising results.

ACKNOWLEDGMENTS

I thank John Bradshaw for discussions and feedback on this abstract.

REFERENCES

Battista Biggio, Giorgio Fumera, and Fabio Roli. Design of robust classifiers for adversarial environments. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 977–982. IEEE, 2011.

Battista Biggio, Giorgio Fumera, and Fabio Roli. Security evaluation of pattern classifiers under attack. *IEEE transactions on knowledge and data engineering*, 26(4):984–996, 2014.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017a.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017b.

Antonia Creswell, Anil A Bharath, and Biswa Sengupta. Latentpoison-adversarial attacks on the latent space. *arXiv preprint arXiv:1711.02879*, 2017.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108. ACM, 2004.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58. ACM, 2011.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. *arXiv preprint arXiv:1702.06832*, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *International Conference on Machine Learning*, pp. 2052–2061, 2017.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

Ian Goodfellow Reuben Feinman Fartash Faghri Alexander Matyasko Karen Hambardzumyan Yi-Lin Juang Alexey Kurakin Ryan Sheatsley Abhibhav Garg Yen-Chen Lin Nicolas Papernot, Nicholas Carlini. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519. ACM, 2017.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJUYGxbCW`.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Pedro Tabacof, Julia Tavares, and Eduardo Valle. Adversarial images for variational autoencoders. *arXiv preprint arXiv:1612.00155*, 2016.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pp. 1747–1756, 2016b.

A DETECTING ADVERSARIAL ATTACKS WITH CONDITIONAL GENERATIVE MODELS

We describe the detection algorithm in more detail, with the conditional VAE as an example. Given an input $\boldsymbol{x}$ and a label $\boldsymbol{y}$, we can "reconstruct" $\boldsymbol{x}$ by

$$\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}), \quad \hat{\boldsymbol{x}} \sim p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}).$$

Therefore $\hat{\boldsymbol{x}}$ depends on $\boldsymbol{x}$ and $\boldsymbol{y}$. In experiments we do not perform sampling and instead compute the reconstruction directly:

$$\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\mu}_{\boldsymbol{x}}(\boldsymbol{\mu}_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y}), \quad \boldsymbol{\mu}_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\boldsymbol{z}], \quad \boldsymbol{\mu}_{\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{y}) = \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{y})}[\boldsymbol{x}].$$

Other reconstruction methods apply, e.g. one can select a distance measure $d(\cdot, \cdot)$ and define

$$\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}) = \arg\min_{\hat{\boldsymbol{x}}} -\log p(\hat{\boldsymbol{x}}|\boldsymbol{y}) + \lambda d(\boldsymbol{x}, \hat{\boldsymbol{x}}).$$

This proposal is not investigated here and we leave it to future work.

Our conjecture is that, for an input-label pair $(\boldsymbol{x}, \boldsymbol{y})$ and its adversarial pair $(\boldsymbol{x}^{\text{adv}}, \boldsymbol{y}^{\text{adv}})$ (here $\boldsymbol{y} \neq \boldsymbol{y}^{\text{adv}}$), we can measure the distance between the input and the reconstruction, and have

$$d(\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y})) < d(\boldsymbol{x}^{\text{adv}}, \boldsymbol{r}(\boldsymbol{x}^{\text{adv}}, \boldsymbol{y}^{\text{adv}})).$$

Therefore, a simple detection method would first compute $\bar{d}_{\mathcal{D}} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[d(\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}))]$, then determine an input $\boldsymbol{x}^*$ as an adversarial example for a classifier F if $d(\boldsymbol{x}^*, \boldsymbol{r}(\boldsymbol{x}^*, F(\boldsymbol{x}^*))) > \bar{d}_{\mathcal{D}}$. It is also possible to have different threshold $\bar{d}_c$ for different classes, however this is not investigated here.

To verify the conjecture we perform detection tests on MNIST with a trained CNN classifier as the victim model and a conditional VAE as the generative model. The attacks in consideration are (untargeted) CW-ℓ_2 (Carlini & Wagner, 2017a) and DeepFool (Moosavi Dezfooli et al., 2016). Table 2 reports the average distortion of successful attacks, the ℓ_2 distance between inputs and reconstructed images, and the detection rate using the average distance $\bar{d}_{\mathcal{D}}$ as the threshold. It is clear that for both attacks $d(\boldsymbol{x}^*, \boldsymbol{r}(\boldsymbol{x}^*, F(\boldsymbol{x}^*))) > \bar{d}_{\mathcal{D}}$ in average, and the detection method is very effective. We also visualise the reconstructed images in Figure 2, where visually the reconstructions of the adversarial images look similar to images in the adversarial classes.

Table 2: Detection experiments with the conditional generative model.

attack	distortion	distance (clean)	distance (adv)	detection rate
DeepFool	1.745 ± 0.732	2.948 ± 0.828	4.930 ± 1.150	97.71%
CW-ℓ_2	1.370 ± 0.530	2.948 ± 0.828	4.995 ± 1.212	97.52%

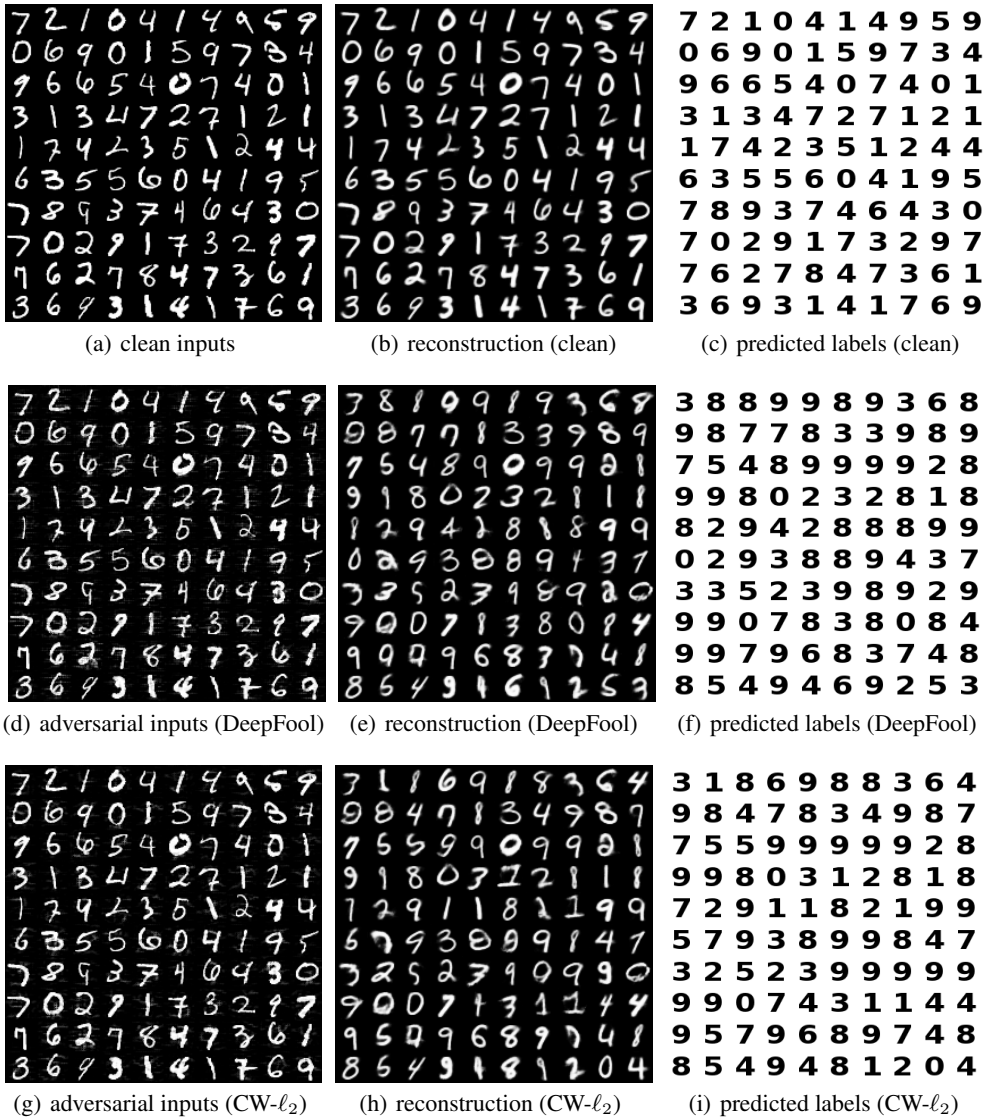
B MODEL ARCHITECTURES

MLP: The MLP has 3 hidden layers of 500 units. We use ReLU activations.

VAE-MLP: The decoder takes $(\boldsymbol{z}, \boldsymbol{y})$ as input and produces $\boldsymbol{x}$ using a two hidden-layer MLP with hidden layer size 500. The encoder has a symmetric architecture except that it takes $(\boldsymbol{x}, \boldsymbol{y})$ as inputs and return the mean and variance parameters of $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$. Here $\boldsymbol{z}$ is 32 dimensions.

CNN: We used 4 convolutional layers with filter size 3 and 128 channels, each followed by a max-pooling operation. Then the output is fed into a one hidden-layer MLP with 500 hidden units to produce the class probability vector.

VAE-CNN: The decoder takes $(\boldsymbol{z}, \boldsymbol{y})$ as input and produces $\boldsymbol{x}$ by a one hidden-layer MLP with 500 units, followed by a 3-layer deconvolutional neural network with filter size 3×3 and number of channels 64, 64, 1. The encoder has almost identical architecture, except that the convolutional part only takes $\boldsymbol{x}$ as inputs, and the MLP part takes $\boldsymbol{y}$ and the convolutional features of $\boldsymbol{x}$. The latent dimension is set to $\dim(\boldsymbol{z}) = 32$.



(a) clean inputs (b) reconstruction (clean) (c) predicted labels (clean)

(d) adversarial inputs (DeepFool) (e) reconstruction (DeepFool) (f) predicted labels (DeepFool)

(g) adversarial inputs (CW-ℓ_2) (h) reconstruction (CW-ℓ_2) (i) predicted labels (CW-ℓ_2)

Figure 2: Visualising the clean, adversarial and reconstructed images, as well as the labels on the clean/adversarial inputs. Many of the reconstructed images from the adversarial inputs are visually more close to the predicted labels on the adversarial images.

C PARAMETERS OF THE ATTACKS

CW-ℓ_2: as recommended by `https://github.com/carlini/nn_robust_attacks/blob/master/l2_attack.py`

DeepFool: as recommended by `https://github.com/tensorflow/cleverhans/blob/master/cleverhans/attacks.py#L1092`