UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica

Harnessing adversarial examples

Machine Learning

Saverio Meucci

Introduction

For this project, a model for a multi-label problem is trained, using the MNIST dataset. Once the model is obtained, adversarial examples can be generated. An adversarial example is an example that is only slightly different from the original and correctly classified example that a model misclassify with a high confidence.

The objective of this projects is to harness adversarial examples to make train more robust model, by lower the classification error and the confidence associated with those misclassification.

Dataset

The MNIST dataset is a large dataset of handwritten digits, thus including ten mutually exclusive classes. It was creates from the samples of the NIST dataset; the originally black and white images were normalized to fit 28x28 pixels images and anti-alisiasing were used, which introduced grayscale levels, so that the pixel values are in the range [0, 255].

The MNIST dataset contains 60,000 training images (divided in two sets, train and validation, with 55,000 and 5,000 images resprectively) and 10,000 testing images.

For this project, from each sample the mean image of the dataset were subtracted.

Explaining adversarial examples

Generating adversarial examples

Let θ be the parameters of a model, x the input to the model, y the targets associated with x (for machine learning tasks that have targets) and $J(\theta, x, y)$ be the cost used to train the neural network. We can linearize the cost function around the current value of θ, obtaining an optimal max-norm constrained pertubation of ,

$$\eta = \epsilon * sign(\nabla_x J(\theta, x, y))$$

The adversarial examples can thus be obtained as follow,

$$x = x + \eta$$

Generating adversarial examples

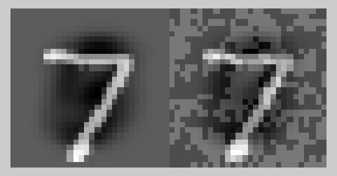


Figure : Example of adversarial generation.

CNN - Architecture

The net is based on LeNet and consists of eight layers, as shown in the following image.

layer	0	1	2	3	4	5	6	7	8
type	input	conv	mpool	conv	mpool	conv	relu	conv	softmxl
name	n/a	layer1	layer2	layer3	layer4	layer5	layer6	layer7	layer8
support	n/a	5	2	5	2	4	1	1	1
filt dim	n/a	1	n/a	20	n/a	50	n/a	500	n/a
num filts	n/a	20	n/a	50	n/a	500	n/a	10	n/a
stride	n/a	1	2	1	2	1	1	1	1
pad	n/a	0	0	0	0	0	0	0	0
rf size	n/a	5	6	14	16	28	28	28	28
rf offset	n/a	3	3.5	7.5	8.5	14.5	14.5	14.5	14.5
rf stride	n/a	1	2	2	4	4	4	4	4
data size	28	24	12	8	4	1	1	1	1
data depth	1	20	20	50	50	500	500	10	1
data num	1	1	1	1	1	1	1	1	1
data mem	3KB	45KB	11KB	12KB	3KB	2KB	2KB	40B	4B
param mem	n/a	2KB	0B	98KB	0B	2MB	0B	20KB	0B

CNN - Settings

- Number of epochs 30
- Learning rate initialized to 0.001
- Softmax log-loss as a cost function
- SGD with momentum initialized to 0.9
- Weight decay of 0.0005

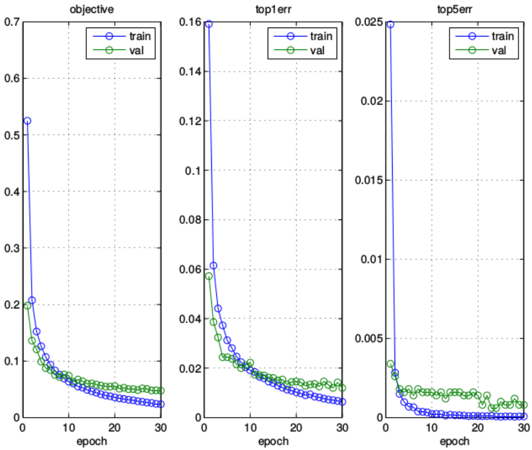
Standard Training

The net was trained in a standard way, using the 60,000 images of the train and validation sets.

The tests carried out on the trained model were of two kinds.

The first test was made using the 10,000 clean samples from the testing set; the second test was made using the adversarial examples, that is the same samples as the first test but with an added perturbation computed as shown previously.

Standard Training



Standard Training

	Clean	Adversarial
Correctly Predicted	98.47	3.36
Error	1.53	96.64
Confidence	98.59	93.82
Confidence Correctly Predicted	99.04	91.67
Confidence Error	69.95	93.89

As we can see from the results, the classification using the clean samples as test images worked great, with high prediction rate and confidence. The results for the adversarial test, however, behaved as expected, with a large percentage of misclassification with a high confidence.

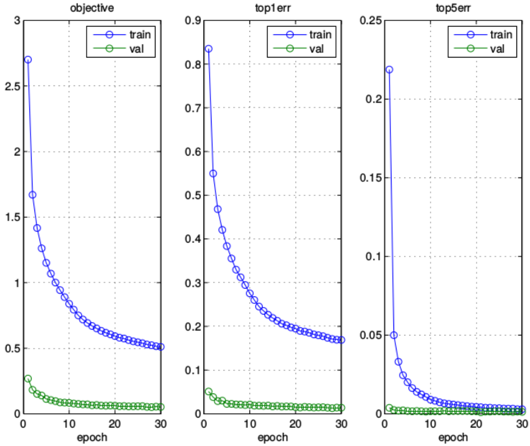
Mixed Training

For this case, adversarial examples were included in the training phase. At each epoch, the adversarial example of each images were computed, given the model at that particular epoch.

Both images, clean and adversarial, were used in the training process, similar to data augmentation. This way, both the clean example and the adversarial example participate in the optimization of the parameters of the model in order to decrease the loss.

Adversarial examples are thus generated dynamically at each epoch, so that the model is trained considering its blind-spots.

Mixed Training



Mixed Training

The tests were carried out as for the standard training.

	Clean	Adversarial
Correctly Predicted	98.21	85.28
Error	1.79	14.72
Confidence	97.89	86.30
Confidence Correctly Predicted	97.85	89.07
Confidence Error	66.91	70.22

Adversarial Training

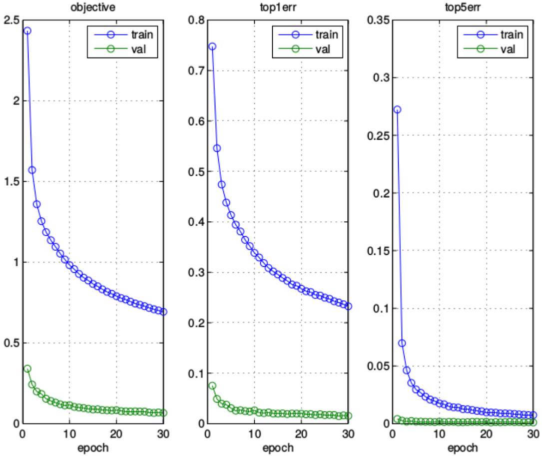
The method of training used for this case is inspired by the one proposed by [1].

In [1], the objective function is modified to include adversarial examples in the training phase, by doing a weighted sum of the loss for the clean and the adversarial examples:

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon \mathit{sign}(\nabla_x J(\theta, x, y)), y)$$

In the implemented method, the same principle is applied only to the computing of the gradients wrt the weights in the backpropagation, so that the direction chosen for the next step will also take into consideration adversarial examples.

Adversarial Training



Adversarial Training

The tests were carried out as for the standard and mixed training.

	Clean	Adversarial
Correctly Predicted	97.97	77.80
Error	2.03	22.20
Confidence	96.03	79.05
Confidence Correctly Predicted	96.69	82.55
Confidence Error	64.23	66.79

Adversarial example transfer - Error Rate

	Standard	Adversarial	Mixed
Standard	96.64	34.78	19.75
Adversarial	0.08	22.20	0.09
Mixed	0.07	0.10	14.72

Table : Error rate when exchanging adversarial examples between models.

Adversarial example transfer - Error Confidence

	Standard	Adversarial	Mixed
Standard	93.89	75.79	75.26
Adversarial	65.19	66.79	64.49
Mixed	69.70	68.05	70.22

Table : Error confidence when exchanging adversarial examples between models.

Conclusion

The results show that for the clean samples the classification for the mixed and adversarial model behaved as for the standard model.

For the adversarial examples, however, the results are much better than the standard approach. For both the mixed and adversarial model, the error rate dropped of $\sim 80\%$ and the confidence for the misclassification also dropped significantly, settling in a similar range as the clean samples for every model.

The results shows that the new models performed well even with adversarial examples generated by other models. Also, consistent with the results of [1], the error rate and the error confidence for the standard model dropped when using adversarial examples generated by the mixed or the adversarial models.

References

[1] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples", arXiv preprint arXiv:1412.6572, 2014