How Real is Real?
Quantitative and Qualitative comparison of GANs and supervised-learning classifiers.
CS229 Project Final report

Riccardo Verzeni (rverzeni), Jacqueline Yau (jyau)

Abstract

In this project we primarily explore how supervised learning classifiers, of increasing complexity, generalize on GANs synthetic MNIST images and secondarily how semi-supervised learning classifier performs on real images. The methods utilized encompass linear classifiers, Neural Networks and modified GANs. The supervised learning classifier seemed to generalize reasonably well on the synthetic images, while semi-supervised learning classifier performed worse than expected on real images.

Introduction

Generative Adversarial Networks (GANs) have been introduced in 2014 combining learning generative models with game theory. In the past four years, some implementations of this new unsupervised-learning model reached quite significant results. They generated qualitatively convincing replicas of the training dataset from random noise. But how convincing are those images for a classifier trained with supervised-learning techniques? Also, could GANs, opportunely modified, improve standard supervised-learning classifiers results? To answer these questions, we evaluated how well supervised learning classifiers (see Methods first subsection), trained on the standard MNIST dataset, generalized on images generated by pre-trained GANs[1]. We additionally picked the best performing classifier and converted it into a semi-supervised learning classifier modifying the standard GANs discriminator architecture as described in this OpenAI paper [5]. We then proceeded comparing the accuracy of the semi-supervised learning classifiers against the original MNIST dataset (see Methods second subsection). The input of our algorithms is an MNIST image, while the output is the label of the classified image.

Related work

Deep neural networks and in particular convolutional neural networks have been extensively used for solving the MNIST digit classification problem and are considered state-of-the-art [3][4]. Current GANs research aims to improve the results of such models in creating more convincing imitations of real images, but also explores how those models could be used for performing semi-supervised learning classification[5].

Dataset and Features

The real MNIST dataset has been downloaded using Keras APIs (60000 training examples, 10000 test examples), while the synthetic MNIST dataset (6336 unlabeled examples) has been generated using this Deep Convolutional Generative Adversarial Networks[1], which has been previously trained separately. The validation set splits 10% of the training set, so actual training of the supervised learning classifiers had 54000 training examples and 6000 validation examples.

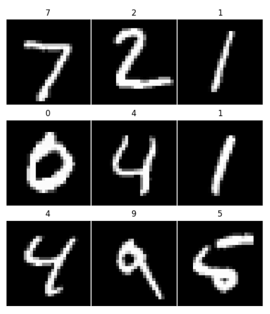

Figure 1: Real MNIST digits sample.

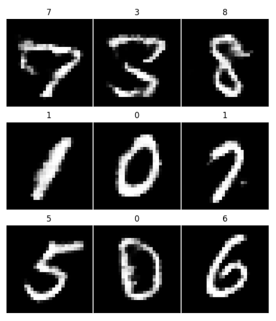

Figure 2: Synthetic MNIST digits sample, manually labelled.

We manually labeled 1000 of the synthetic images acting as ground truth creating a synthetic test dataset for the quantitative accuracy comparison. Thus, the two test datasets we used had 1000 real and 1000 synthetic examples respectively.

Each MNIST grayscale image has dimensions $1 \times 28 \times 28$. The input features are the pixels that compose the MNIST grayscale image. For what concerns the features learned by the classifiers, the neural networks derive new features in the hidden layers, and the convolutional ones extract additional features in the convolutions.

Methods

For consistency we used the cross entropy loss and Adam optimization for all classifiers. Specifically we used Keras sparse categorical cross entropy loss function, which is the cross entropy function for discrete distributions with the constraint that the output labels must be integers ranging in values of the classes.

$$\mathcal{H}(p,q) = -\sum_x p(x)\log q(x)$$

Softmax has been used as activation for all output layers,

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$$

in addition the semi-supervised classifier also uses the Sigmoid activation for its second output layer

$$g(x) = \frac{1}{1+e^{-x}}$$

Supervised-learning classifiers

We built and trained a total of 4 supervised-learning classifiers: a linear Softmax classifier, a 2 layers fully connected neural network (NN) (Fig. 3), a 4 layers (Fig. 4) and a 5 layers (Fig. 5) convolutional neural networks (CNN).

The fully connected NN takes in the input image with dimensions 1 x 28 x 28, then flattens it and sends it through ReLU activation at 256 neurons, and then the softmax layer at 10 neurons for classification.

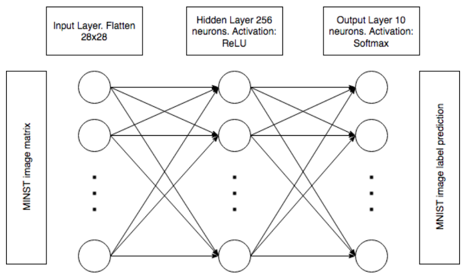
Input Layer. Flatten 28x28
Hidden Layer 256 neurons. Activation: ReLU
Output Layer 10 neurons. Activation: Softmax
MINST image matrix
MNIST image label prediction

Figure 3: Fully-connected NN classifier architecture.

The CNN 4-layer starts with the input image and convolves it with 32 filters with kernel size 5. Then, it goes through max pooling with pool size 2, and the output is flattened and sent through ReLU activation (128 neurons) before the softmax output layer (10 neurons). The CNN 5-layer is identical to the CNN 4-layer architecture, but it has an additional convolution layer, with 64 filters and kernel size 5, after the first one.

For building training and testing our models we used Tensorflow Keras high-level APIs[2]. The supervised learning classifiers trained on 54000 samples with 6000 validation. For comparing the accuracy of trained classifiers against synthetic images we tested them on 1000 real and the 1000 manually labeled synthetic MNIST images.

Semi-supervised learning classifier

The semi-supervised modified GANs discriminator has the same architecture as the CNN 5-layer, but with two output layers (Fig. 6): a 1 neuron Sigmoid output layer for the discriminator to determine whether an image is real or synthetic, and the 10+1 neuron Softmax output layer to classify the image between 0 and 9 or synthetic which is when the

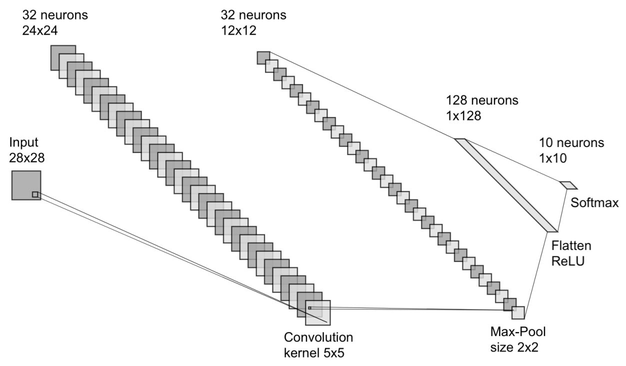

Figure 4: Convolutional NN 4-layer classifier architecture.

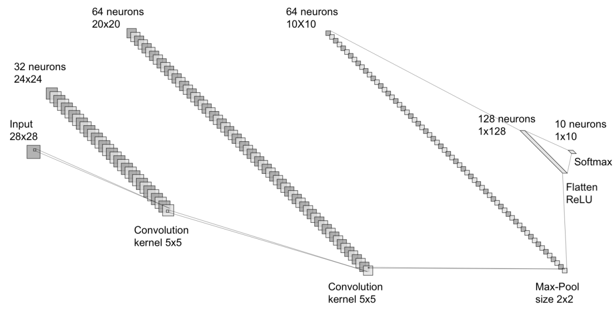

Figure 5: Convolutional NN 5-layer classifier architecture.

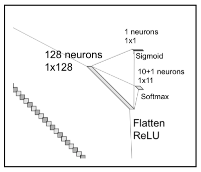

Figure 6: Particular of the architecture of the modified GANs discriminator used as semi-supervised classifier showing the two output layers.

additional "+1" neuron is activated. Due to time constraints, we used previously generated synthetic images as unlabeled data, instead of also building a generator and training it along with the modified discriminator as shown by Tim S. and All[1]. Since we only had around 6000 synthetic images we decided to train the semi-supervised modified GANs on two smaller datasets. The first train dataset was created combining 5400 labeled real images with 5400 unlabeled synthetic images, while the real images test dataset remained the same used for the other classifiers (1000 real images). For comparison reasons we then trained the same classifier on a 5400 labeled real images and 0 unlabeled synthetic images dataset, still testing the performances on the 1000 real images test dataset. Each of the output layers had its own respective loss function and were contributing equally in the optimization process.

Models	Epochs					
	5		50		100	
	Real	Syn	Real	Syn	Real	Syn
Lin softmax	0.919	0.923	0.921	0.921	0.916	0.913
FCNN	0.978	0.957	0.979	0.969	0.983	0.962
CNN-4l	0.986	0.971	0.989	0.970	0.989	0.974
CNN-5l	0.994	0.972	0.988	0.973	0.985	0.969

Table 1: Supervised learning classifier results on test accuracy for real and synthetic images

Models	Epochs		
	5	50	100
Lin softmax	0.923	0.936	0.937
FCNN	0.989	0.999	0.999
CNN-4l	0.995	1.000	1.000
CNN-5l	0.996	0.998	0.998

Table 2: Supervised learning classifier results on train accuracy on real images

Experiments and Results Discussion

Our primary quantitative metric of evaluation was accuracy. The accuracy was calculated as the number of correct classifications over the number of total number of classifications. To better understand which digits were most affected by misclassification we also calculated the confusion matrix for the best performing classifier against both the real and the synthetic test dataset (Fig. 9, 10). For our hyperparameters, we chose to train each model respectively for 5, 50, and 100 epochs to give us an idea on how each one of them would have performed for increasing periods of training time and at which point they would have started showing signs of overfitting. Since we had a large number of data samples we decided to dedicate a significant amount to the validation split: 6000 as it was constituting only the 10% of the total train dataset. The baseline used for classification was the accuracy of a random classifier, which for MNIST is ten percent.

Supervised learning classifiers

The test accuracy obtained by the various supervised learning classifiers over synthetic images was overall slightly worse but comparable with the real images, where the most complex convolutional NN (CNN-5l) performed best. Overall, the various supervised classifier seemed to generalize reasonably well on GANs synthetic MNIST images. We expected the CNN to perform better than any other classifier since it is considered state-of-the-art for the MNIST digit classification problem[4]. That has been indeed the case. The convolutional layers were able to extract meaningful features that were effective in identifying and then classifying the number in the MNIST image. The 5-layer CNN performed better than the 4-layer one since the extra convolution layer with 64 filters was able to find additional features that the first 32 filter convolution layer could not.

However, the CNN did overfit over long training periods (Fig. 8), and while we did not have enough time to implement regularization and test it, in future work we include what approach we would like to try to mitigate it.

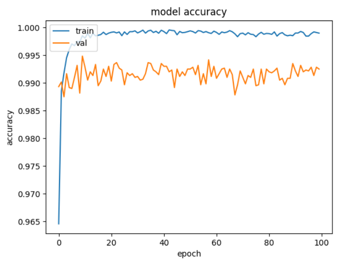

Figure 7: CNN-5l model accuracy

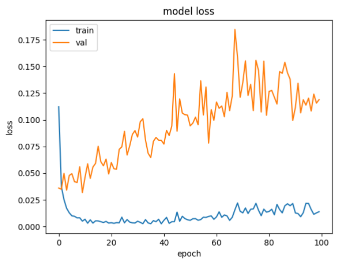

Figure 8: CNN-5l model loss

To analyze the classification results for our best performing model, the CNN 5 layer, we used a confusion matrix to see which examples it had the most trouble classifying and attention maps to observe which features the CNN focused on when classifying.

For what concerns the qualitative analysis: we investigated where in the real and synthetic images the CNN focused its attention on, using two types of attention visualization maps. The first type of attention map used is the saliency, which visualizes activation over the final dense layer (Softmax activation). Specifically, the saliency computes the gradient of the output category with respect to the input image to see how much the output classification changes with respect to the input image pixels. This highlights regions that contribute the most towards the output. (To properly visualize activation, the Softmax layer is swapped for a linear layer, as suggested by Keras-vis API because maximizing an output node can be done by minimizing other outputs.) The second type of attention map used is the class activation map, which instead of using the gradients with respect to output, uses the penultimate convolution layer output. It uses the nearest convolution layer to utilize spatial information lost in the dense layers. The attention maps are in the appendix (Fig. 11-14).

For each attention map, there are three different ways we modified the backpropagation gradients. 'Vanilla' is the case when there is no modification performed on the backpropagation gradients. The 'ReLU' modifier clips negative gradient values, and the 'guided' modifier modifies the backpropagation to only propagate positive gradients for positive activations. The 'guided' modifier seems to produce the clearest attention maps.

The confusion matrix describes the performance of the classifier model and displays how many and which examples were classified correctly or incorrectly. Analyzing the results from the confusion matrix of real images (Fig. 9), the CNN performs very well and there are not many errors. However, of the mistakes it seems that the number the CNN has most trouble classifying is 3. We observe that the CNN most frequently misclassifies 3 as 5. Comparing the results from the confusion matrix with the attention maps, we can observe that the features highlighted by the CNN to classify 5 mainly focuses on the middle, to identify where the curve starts. Since it seems like the CNN tries to classify 5 based on if there is a curve in the middle, that would explain why it would misclassify 3 as 5, since 3 has a second curve in the middle. And the one instance of a wrong prediction on 5, is a misclassification as 3.

In comparison to the real images, we observe the results for the confusion matrix on the synthetic images (Fig. 10) and see that overall the CNN 5 layer performed worse, and it especially had trouble classifying 1 and 6. It seems that the CNN most often misclassified the synthetic image 6 as 5 and 1 as 7. The synthetic 5 seems to be classified based on the top of the horizontal line and the bottom of curve. Since 6 has a bottom curve as well, that may be why 6 was incorrectly predicted to be 5. For the case of predicting 1, the saliency attention of synthetic 7 appears to concentrate on the slanted vertical line. Since 1 is a completely vertical line that is sometimes slanted, that would explain why 1 is misclassified as 7. Curiously, the classifier could predict 7 accurately, and did not misclassify 7 as 1.

Overall, based on the results of the confusion matrices, the CNN 5 layer performs very well on the real images but comparatively not as great on the synthetic ones, though total accuracy is still quite high. Interestingly, the attention maps showed that the CNN's focus features for identifying a real image may be different for the same number synthetic image. For example, the features to identify real image 5 and synthetic 5 are quite different.

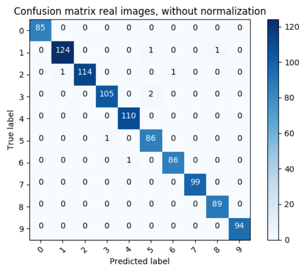

Figure 9: Confusion matrix for CNN-5l over real images of test dataset

Semi-supervised learning classifier

The test accuracy obtained by training the semi-supervised learning classifier on the combined labeled real / unlabeled synthetic images dataset was worse than the one obtained

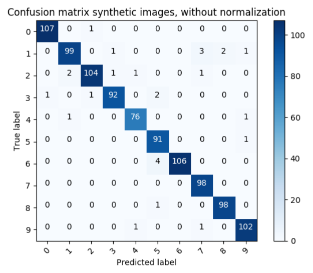

Figure 10: Confusion matrix for CNN-5l over synthetic images of test dataset

by training it on the fully labeled real images dataset (Table 3,4).

Models	Epochs		
	5	50	100
Semi-gan (all labeled)	0.973	0.979	0.977
Semi-gan (half/half)	0.857	0.932	0.933

Table 3: Semi-supervised modified GANs results on test accuracy for label classification of real images

Models	Epochs		
	5	50	100
Semi-gan (all labeled)	0.995	1.000	1.000
Semi-gan (half/half)	0.974	1.000	1.000

Table 4: Semi-supervised modified GANs results on train accuracy for label classification of real images

We expected that the addition of the unlabeled synthetic data would have improved the label classifier accuracy on real MNIST images instead of worsening it. The latter seemed to be the case. This is probably because, as mentioned in the Methods section due to time constraints, we used previously generated synthetic images instead of building a generator. As a result the features extracted by the unlabeled data might not have been as relevant as they would have been if coming from the same distribution of the real data that the generator would have reproduced.

Conclusion and Future Work

We trained four supervised learning classifiers on real MNIST images and tested them on real and synthetic images for classification. We also trained a semi-supervised learning modified GANs discriminator to classify real images as well as discern whether an image is real or synthetic. The CNN 5 layer performed the best on classifying the real and synthetic images. The CNN 5 layer performed better probably because it had an extra convolution layer with 64 filters, so it could extract features that the other models could not.

For future work, the goal would be to implement some approaches to solve issues noticed during error analysis.

For the supervised learning classifiers, since the CNN was overfitting pretty severely, the next step would be to add some regularization to help reduce the overfitting, such as a dropout layer. Another approach to try out is implementing a ResNet to see if it could perform even better than a CNN.

For the semi-supervised learning GAN, the next step would be to build a complete semi-supervised learning GANs classifier, such as the one suggested[1] and see if that would improve the results obtained in our partial attempt. Building the generator for the GANs would allow the adversarial training between the discriminator and the generator, which would hopefully improve the label classifier results. All the code used for this project is available at: `https://github.com/project-mnist-2018/cs229_project`

Appendix

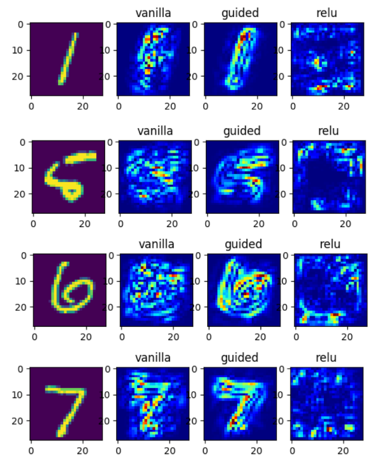

Figure 11: Saliency attention for CNN-5l on a real digit sample.

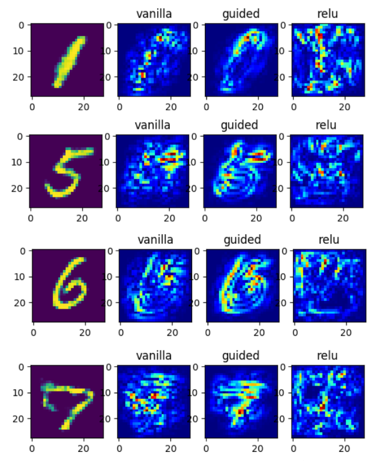

Figure 12: Saliency attention for CNN-5l on a synthetic digit sample.

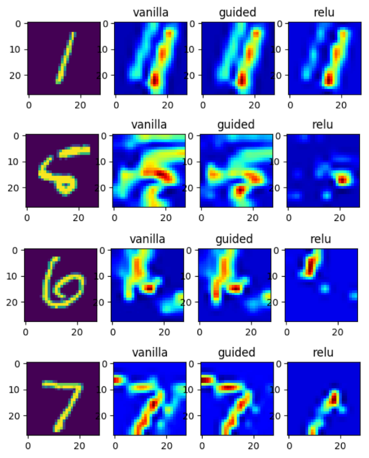

Figure 13: Class Activation attention Class Activation Attention for CNN-5l on a real digit sample.

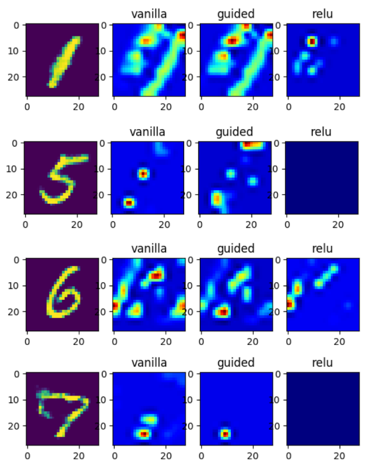

Figure 14: Class Activation attention Class Activation Attention for CNN-5l on a synthetic digit sample.

Contributions

Both team members worked on coding up the various classifier models with visualizations, running said models to gather results, labelling synthetic images, and writing up the proposal, milestone, poster, and final report.

References

[1] A tensorflow implementation of "deep convolutional generative adversarial networks". https://github.com/carpedm20/DCGAN-tensorflow.

[2] Tensorflow keras high-level apis. https://www.tensorflow.org/guide/keras.

[3] Ueli M. Dan C. and Jurgen S. Multi-column deep neural networks for image classification. in computer vision and pattern recognition (cvpr), 2012 ieee con- ference on, pages 36421–3649. ieee. 2012.

[4] Andrew Z. Karen S. Very deep convolutional networks for large-scale image recognition. arxiv preprint arxiv:1409.1556. 2014.

[5] Tim S., Ian G., Wojciech Z., Vicki C., Alec R., and Xi C. Improved techniques for training gans. 2016. https://arxiv.org/pdf/1606.03498.pdf.