LEARNING PRIORS FOR ADVERSARIAL AUTOENCODERS

A PREPRINT

Belozerova Polina
Skoltech
bel.pol.4@gmail.com

Safin Alexander
Skoltech
safinsam@yandex.ru

Pavlovskaia Natalia
Skoltech
ya-ne-bo@yandex.ru

October 25, 2018

ABSTRACT

We took here several attempts to create a generative model. The main source of inspiration is [1]. The results are presented for MNIST and CIFAR-10 datasets. We haven't achieve the same quality as in the original paper, but learnt a lot.

***K*eywords** AAE · GAN

1 Introduction

1.1 Goal

Our goal is to reproduce the "Learning priors for adversarial autoencoders" [1]

1.2 Description

Most deep latent factor models choose simple priors for simplicity, tractability or not knowing what prior to use. Recent studies show that the choice of the prior may have a profound effect on the expressiveness of the model, especially when its generative network has limited capacity. In this paper, we propose to learn a proper prior from data for adversarial autoencoders (AAEs). We introduce the notion of code generators to transform manually selected simple priors into ones that can better characterize the data distribution. Experimental results show that the proposed model can generate better image quality and learn better disentangled representations than AAEs in both supervised and unsupervised settings. Lastly, we present its ability to do cross-domain translation in a text-to-image synthesis task.

2 Algorithms

2.1 Adversarial Autoencoder

"Adversarial Autoencoders" [2] Sasha put here the algorithm description

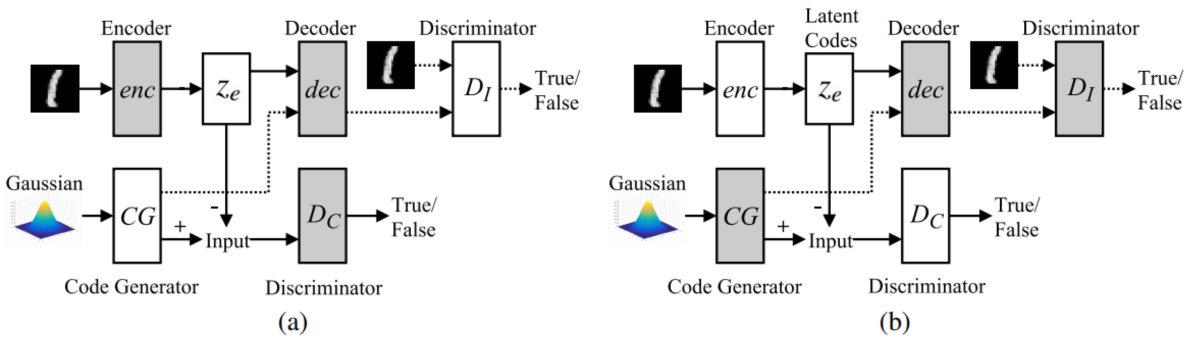
2.2 Original paper algorithm scheme

2.3 Original paper algorithm pseudo code

2.4 Additional algorithm

Is inspired by

"Autoencoding beyond pixels using a learned similarity metric"[3]



Alternation of training phases: (a) the AAE phase and (b) the prior improvement phase.

Figure 1: Original paper algorithm scheme

Algorithm 1 Training algorithm for our method.

$\theta_{enc}, \theta_{dec}, \theta_{CG}, \theta_{D_I}, \theta_{D_C}, \theta_Q \longleftarrow$ Initialize network parameters
Repeat (for each epochs E_i)
  Repeat (for each mini-batch x_j)
    // AAE phase
    $z \sim p(z)$
    If conditional variables s exist then
      $z_c \leftarrow CG(z, s)$
    Else
      $z_c \leftarrow CG(z)$
    End If

    $\mathcal{L}^C_{GAN} \leftarrow \log(D_C(z_c)) + \log(1 - D_C(enc(x)))$
    $x_{rec} \leftarrow dec(enc(x))$
    $\mathcal{L}_{rec} \leftarrow \frac{1}{N}\|\mathcal{F}(x) - \mathcal{F}(x_{rec})\|_2$

    // Update network parameters for AAE phase
    $\theta_{D_C} \leftarrow \theta_{D_C} - \nabla_{\theta_{D_C}}(\mathcal{L}^C_{GAN})$
    $\theta_{enc} \leftarrow \theta_{enc} - \nabla_{\theta_{enc}}(-\mathcal{L}^C_{GAN} + \mathcal{L}_{rec})$
    $\theta_{dec} \leftarrow \theta_{dec} - \nabla_{\theta_{dec}}(\lambda * \mathcal{L}_{rec})$

    // Prior improvement phase
    $z \sim p(z)$
    If conditional variables s exist then
      $z_c \leftarrow CG(z, s)$
    Else
      $z_c \leftarrow CG(z)$
    End If

    $x_{noise} \leftarrow dec(z_c)$
    $x_{rec} \leftarrow dec(enc(x_j))$
    $\mathcal{L}^I_{GAN} \leftarrow \log(D_I(x_j)) + \log(1 - D_I(x_{noise})) + \log(1 - D_I(x_{rec}))$

    // Update network parameters for prior improvement phase
    $\theta_{D_I} \leftarrow \theta_{D_I} - \nabla_{\theta_{D_I}}(\mathcal{L}^I_{GAN})$
    If conditional variables s exist then
      $\theta_{dec} \leftarrow \theta_{dec} - \nabla_{\theta_{dec}}(-\mathcal{L}^I_{GAN} + I(s; dec(z_c))$
      $\theta_Q \leftarrow \theta_Q - \nabla_{\theta_Q}(I(s; dec(z_c)))$
    Else
      $\theta_{dec} \leftarrow \theta_{dec} - \nabla_{\theta_{dec}}(-\mathcal{L}^I_{GAN})$
    End If
  Until all mini-batches are seen
Until terminate

Figure 2: Original paper algorithm pseudo code

2.5 Additional algorithm pseudo code

3 Data

- MNIST
- CIFAR-10

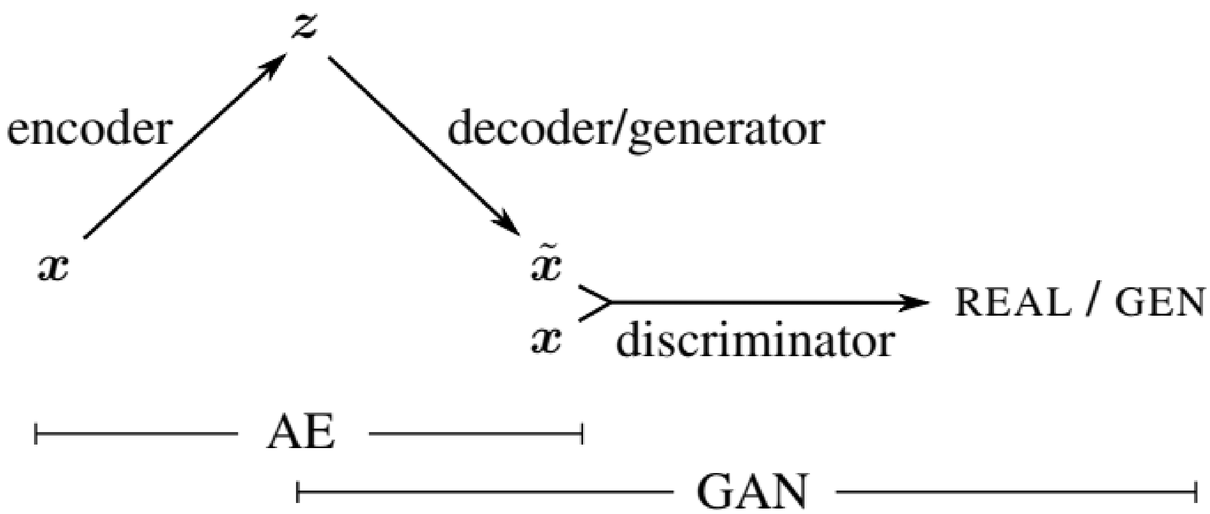


Figure 3: Additional algorithm scheme

Algorithm 1 Training the VAE/GAN model

$\boldsymbol{\theta}_{\mathrm{Enc}}, \boldsymbol{\theta}_{\mathrm{Dec}}, \boldsymbol{\theta}_{\mathrm{Dis}} \leftarrow$ initialize network parameters
repeat
  $\boldsymbol{X} \leftarrow$ random mini-batch from dataset
  $\boldsymbol{Z} \leftarrow \mathrm{Enc}(\boldsymbol{X})$
  $\mathcal{L}_{\text{prior}} \leftarrow D_{\mathrm{KL}}(q(\boldsymbol{Z}|\boldsymbol{X}) \| p(\boldsymbol{Z}))$
  $\tilde{\boldsymbol{X}} \leftarrow \mathrm{Dec}(\boldsymbol{Z})$
  $\mathcal{L}_{\text{llike}}^{\mathrm{Dis}_l} \leftarrow -\mathbb{E}_{q(\boldsymbol{Z}|\boldsymbol{X})}\left[p(\mathrm{Dis}_l(\boldsymbol{X})|\boldsymbol{Z})\right]$
  $\boldsymbol{Z}_p \leftarrow$ samples from prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{X}_p \leftarrow \mathrm{Dec}(\boldsymbol{Z}_p)$
  $\mathcal{L}_{\mathrm{GAN}} \leftarrow \log(\mathrm{Dis}(\boldsymbol{X})) + \log(1 - \mathrm{Dis}(\tilde{\boldsymbol{X}})) + \log(1 - \mathrm{Dis}(\boldsymbol{X}_p))$
  // Update parameters according to gradients
  $\boldsymbol{\theta}_{\mathrm{Enc}} \stackrel{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\mathrm{Enc}}}(\mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\mathrm{Dis}_l})$
  $\boldsymbol{\theta}_{\mathrm{Dec}} \stackrel{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\mathrm{Dec}}}(\gamma\mathcal{L}_{\text{llike}}^{\mathrm{Dis}_l} - \mathcal{L}_{\mathrm{GAN}})$
  $\boldsymbol{\theta}_{\mathrm{Dis}} \stackrel{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\mathrm{Dis}}}\mathcal{L}_{\mathrm{GAN}}$
until deadline

Figure 4: Additional algorithm pseudo code

4 Problems and solutions

Problems

- Different formulae in the text and the pseudo code in the original paper
- 1 epoch even for MNIST takes about 17 minutes
- We had no time to try supervised setting

Solutions

- Using other papers and do updates according to the our own understanding
- Several updates of encoder-decoder
- Adjusting learning rates

5 Results

5.1 Results for AAE

Sasha put here your pictures

5.2 Results evolution for original paper. MNIST.

The most enchanting thing here is evolution with epochs

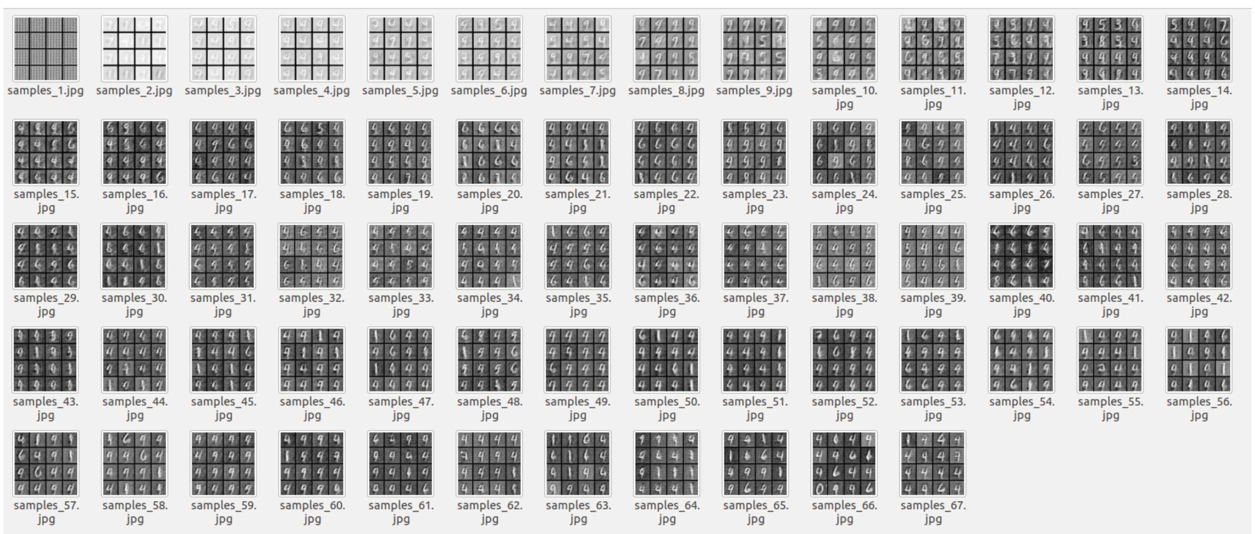


Figure 5: Results evolution for original paper. MNIST

5.3 Results for original paper. MNIST, 67 epoch.

5.4 Results evolution for original paper. CIFAR.

The results are not good at all. Here the latent dimension is 8 the same as for MNIST.

5.5 Results for original paper. CIFAR, 55 epoch.

5.6 Results for additional algorithm

5.7 Conclusions

- The adversarial training is difficult
- There are a lot of different schemes of generative models combining GAN and VAE

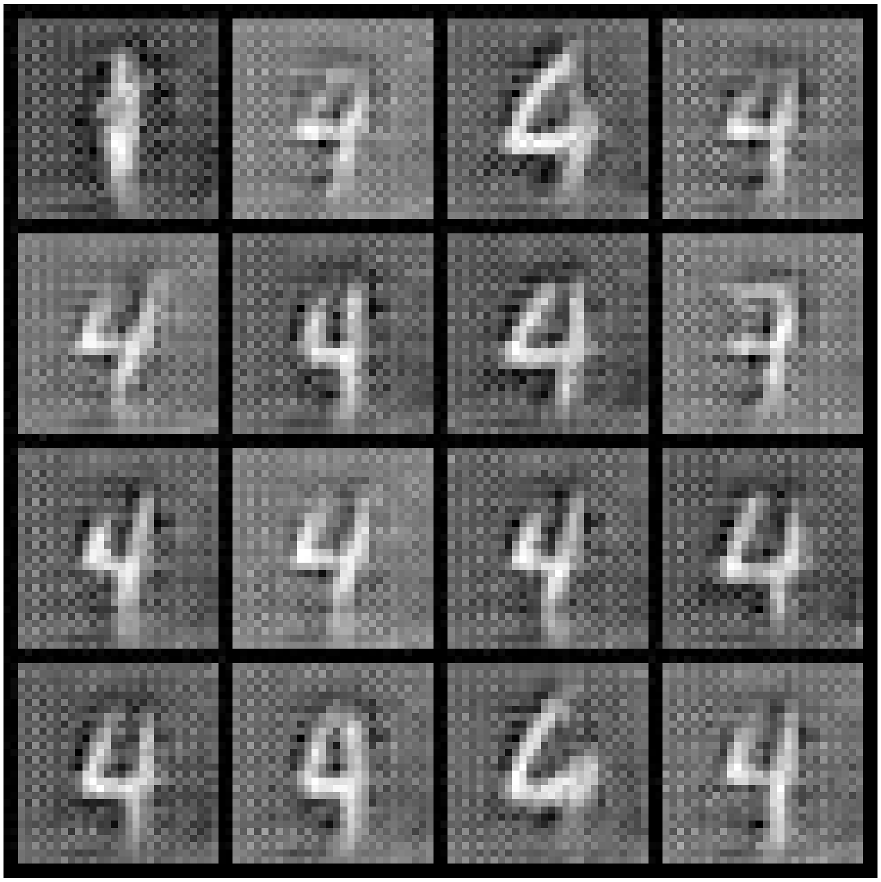


Figure 6: Results for original paper. MNIST, 67 epoch

5.8 Contributions

- Belozerova Polina: reading papers
- Safin Alexander: reading papers, implement the classical AAE, experiments for MNIST, corresponding parts inthe presentation and report
- Pavlovskaia Natalia: reading papers, implement the original paper, implement the additional algorithm, experiments for MNIST and CIFAR-10, corresponding parts inthe presentation and report

References

[1] Hui-Po Wang. Learning priors for adversarial autoencoders. 2018.

[2] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015.

[3] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

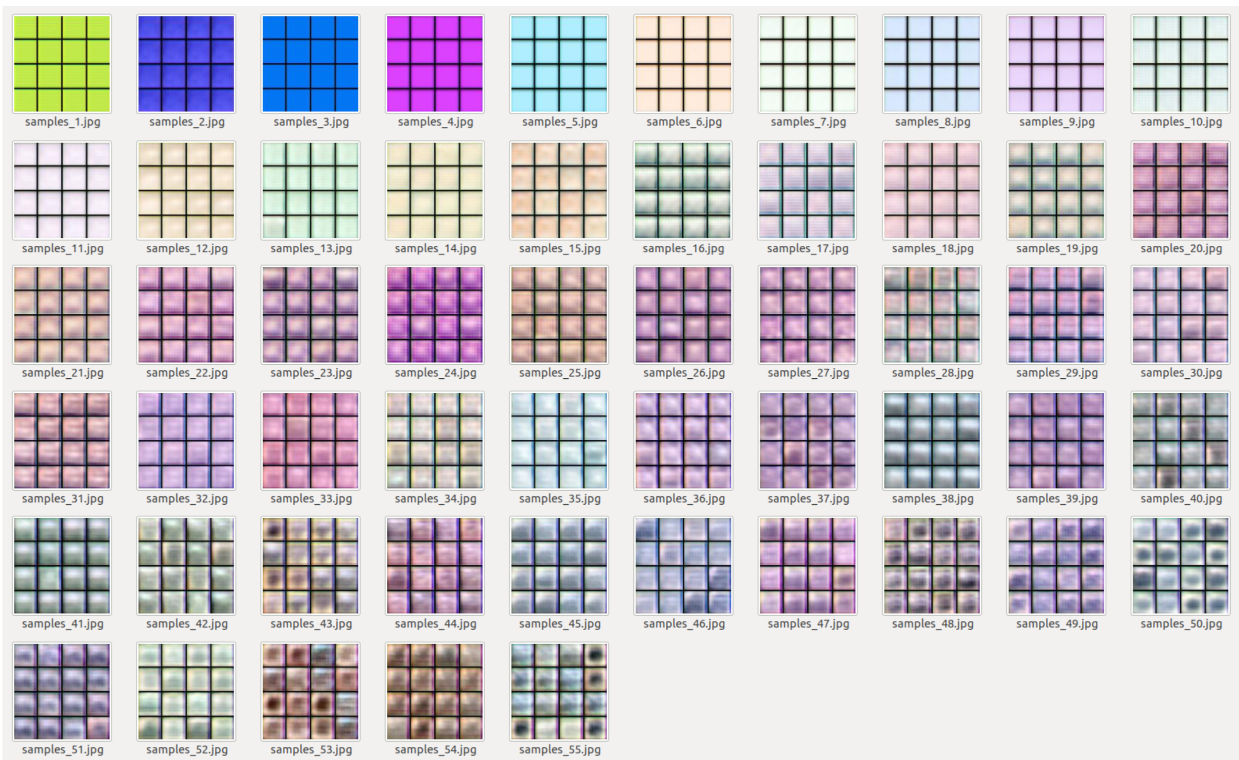


Figure 7: Results evolution for original paper. CIFAR

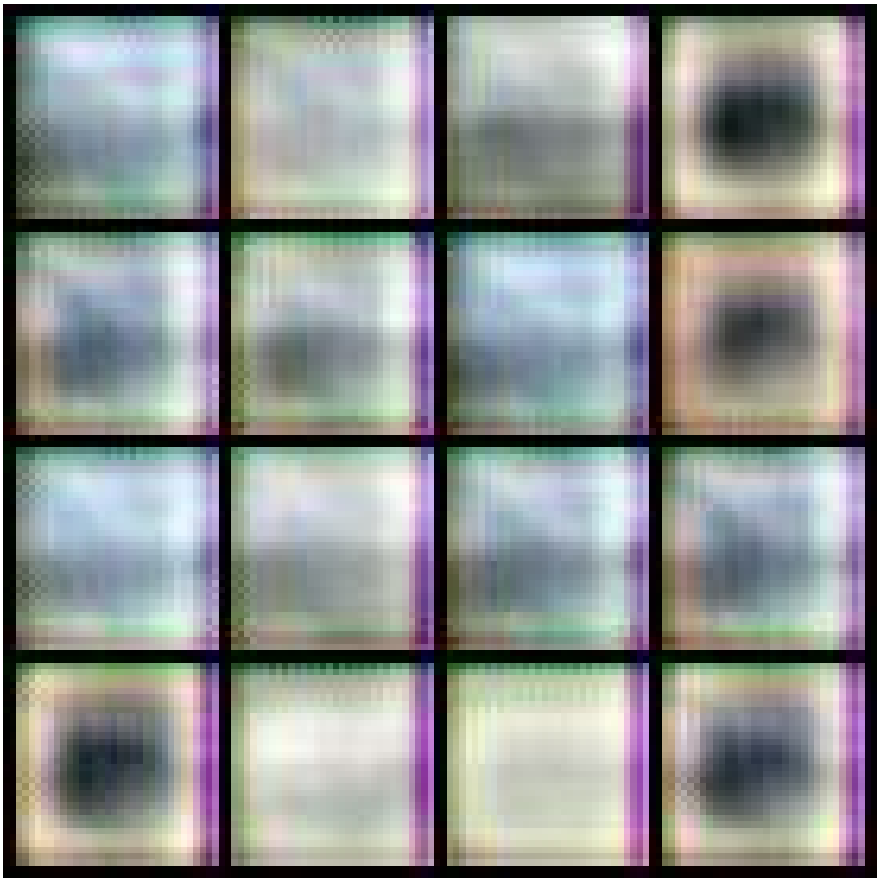


Figure 8: Results for original paper. CIFAR, 55 epoch

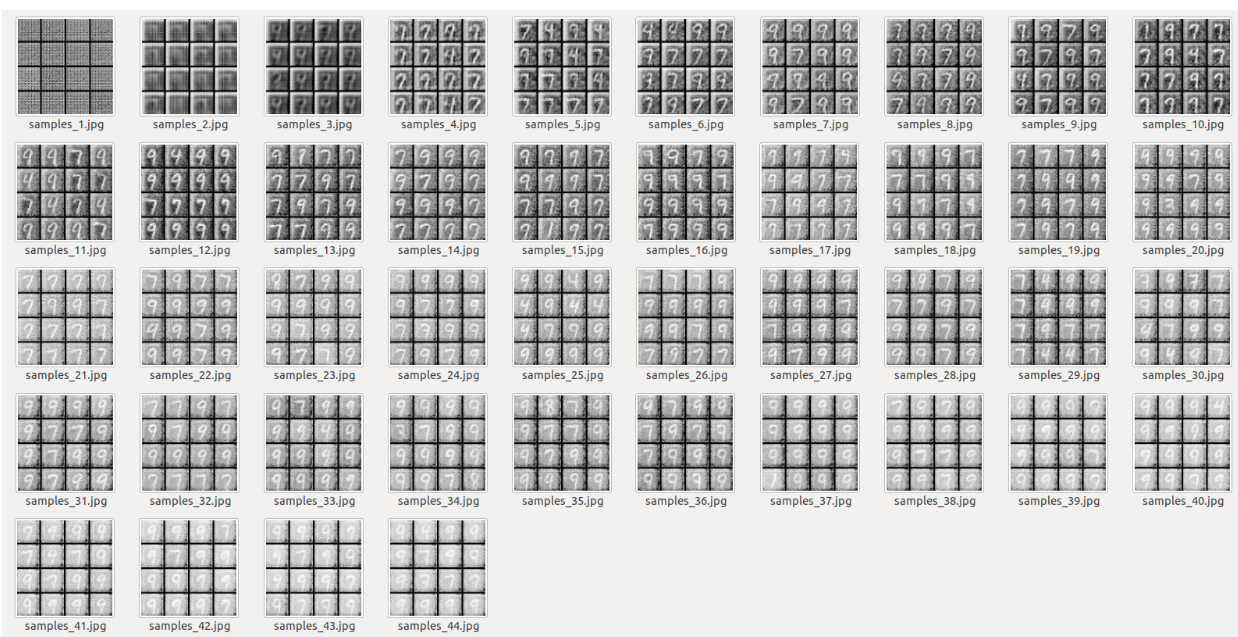


Figure 9: Results evolution for additional algorithm. MNIST