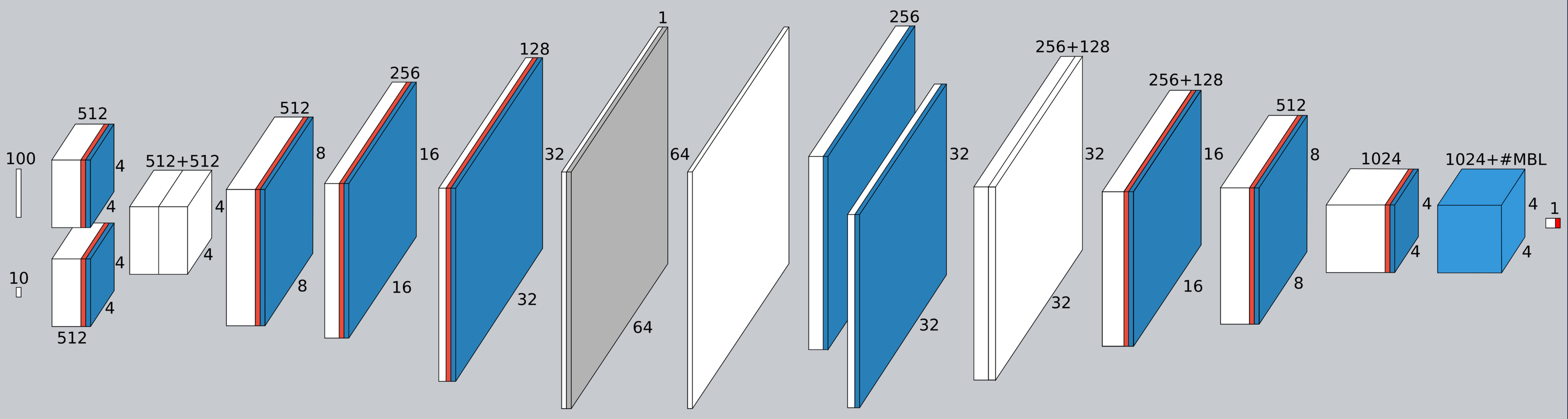
Semi-Supervised Learning using DCGANs

Anna Baumeister (w270) anna.baumeister@tum.de
Nicholas Gao (w348) nicholas.gao@tum.de
Matthias Mayer (w213) matthias.mayer@tum.de
Frederic Rackwitz (w515) rackwitz@in.tum.de
Yannick Spreen (w231) y.spreen@tum.de

We investigated several ideas to improve GAN training: Conditional GAN [HKang], Feature Matching and Mini-Batch Discrimination [OpenAI]. We applied those ideas in the training of DCGAN [Radford] on MNIST as well as CIFAR10 and evaluated the results visually (for MNIST) and by achievable results in semi-supervised learning (for CIFAR10). Trials on MNIST showed reasonable outcomes with all ideas, but espacially cGAN, and semi-supervised learning on CIFAR10 succeeded for all but cGAN with mostly similar accuracy.

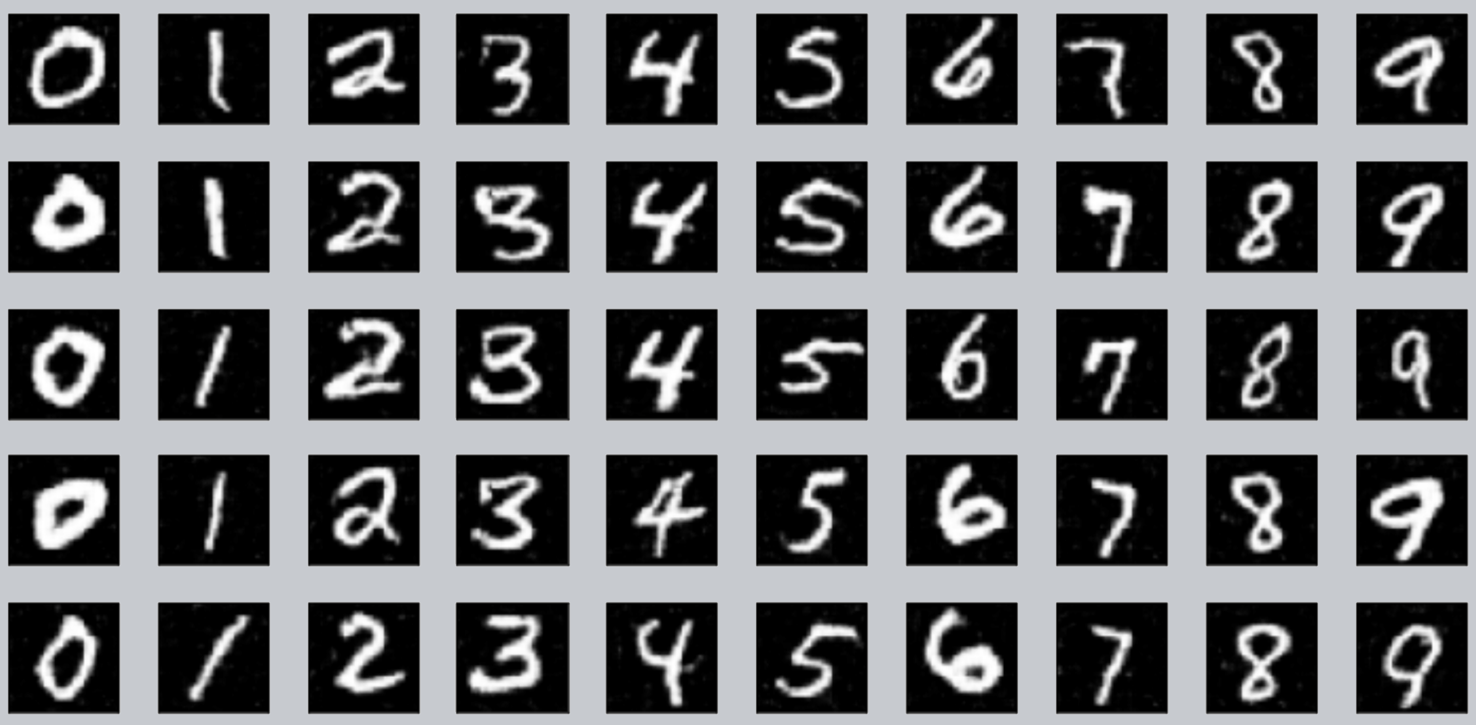
Generative Adversarial Networks

Generative Adversarial Networks (GANs) consist of two submodels: A discriminator trying to differentiate between real and fake images and a generator trying to produce convincing fakes. Both networks are trained simultaneously and compete to lower their respective losses based on the discriminator output. We use a Deep Convolutional GAN in which both the generator and discriminator are fully convolutional. The figure below shows our architecture.



Conditional GANs

In contrast to traditional GANs which try to model the data distribution $p(x)$, conditional GANs (cGANs) aim to condition $p(x)$ on targets y. This is realized by adding class labels as an input for both nets, allowing the generation of images that belong to a specific class. Furthermore, it prevents the generator from interpolating between multiple classes.
In contrast to training a single GAN for each class, cGANs allow feature sharing if the classes are correlated. Unfortunately, cGANs can only be trained fully supervised and often offer worse results in case of weakly correlated classes.



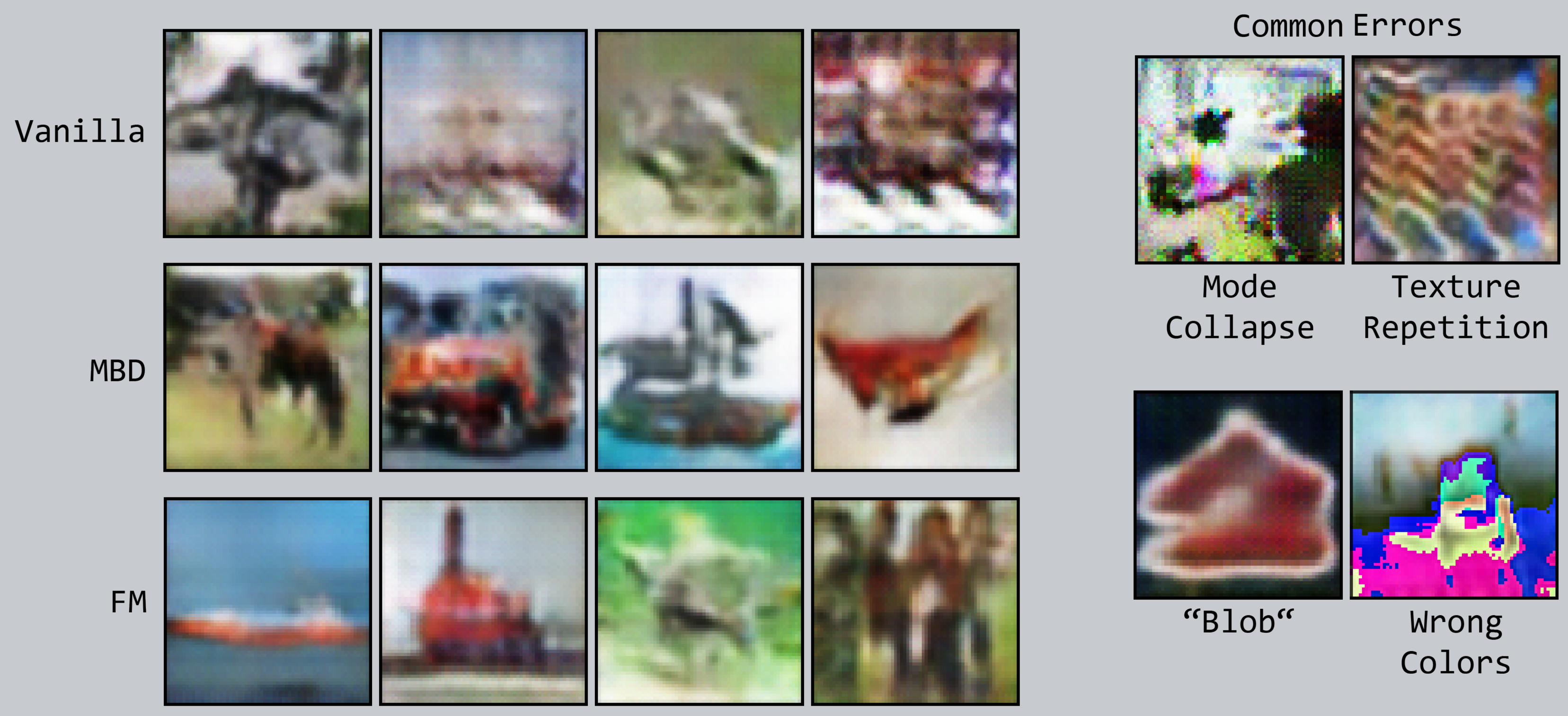
Feature Matching

Feature Matching introduces another loss function for the generator to minimize. In addition to maximizing the missclassification rate of its output, the generator tries to meet the expectation of a feature map $f(x)$ within the discriminator. With stochastic optimization, this results in the additional loss term $||\mathrm{Mean}(f(x)) - \mathrm{Mean}(f(G(z)))||_2$ for the generator. Here x is the feature map from a batch of real images, z is the random latent space variable for the generator and f is the last hidden feature map of the discriminator. Some examplary results are shown in the figure on the right-hand side.

Mini-Batch Discrimination

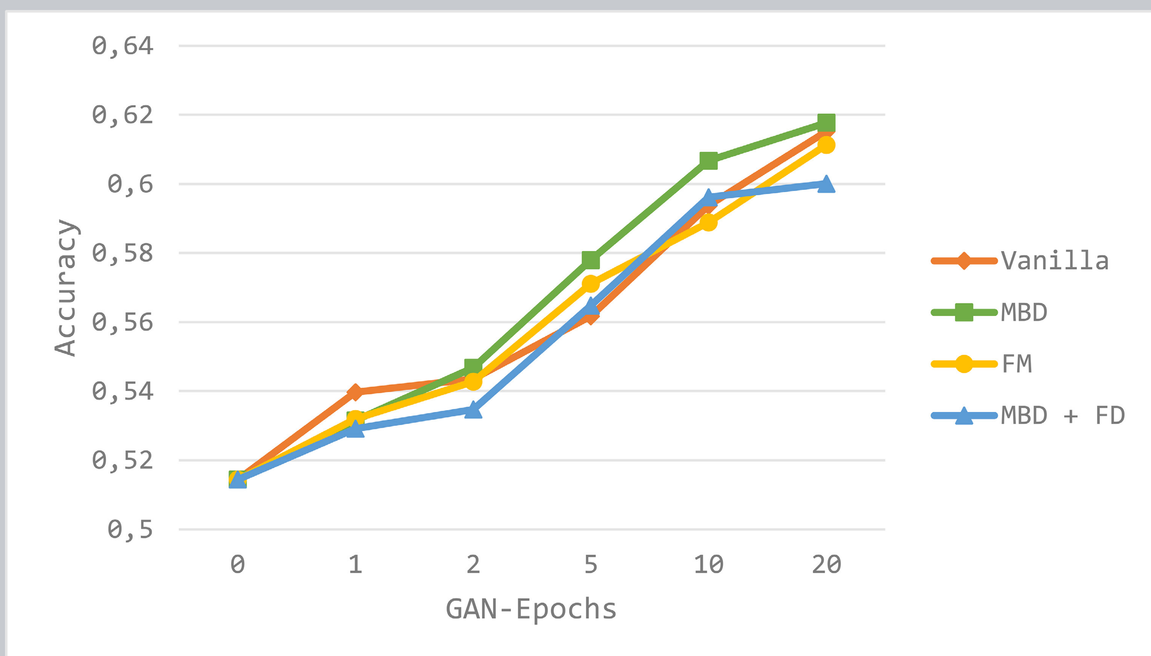
Mode collapse is a common failure case for GANs. When it happens, the generator emits images only from a single point of the distribution which achieves a good realness-score. Because the discriminator looks at each example from the batch in isolation, it has no way of judging whether the images produced by the generator have the correct amount of entropy. Mini-Batch Discrimination attempts to solve this issue by allowing the discriminator to look at multiple datapoints of one batch at the same time. This allows the net to be trained more steadily. Some examplary results are shown in the figure below.

CIFAR10 Results



Semi-Supervised Learning

One major reason for the recent success of deep CNNs is the availability of huge labeled datasets, such as ImageNet, which are expensive to build. Semi-supervised learning tries to establish an approach for big datasets that have not been labeled yet. We tried to achieve this with GANs. In theory, their discriminators should learn an effective feature representation of images during training without labels. The discriminator's first layers may therefore be reused for a classification task, after being learned on a labeled subset of the data. Results on CIFAR10 are shown below:



The net of the figure above was trained in two steps. First, n GAN-epochs were trained with the entire CIFAR10 set. Afterwards, a classifier was trained with the learned weights of the GAN. The classifier trained on 4000 (<10%) labeled samples of CIFAR, until it saturated. The figure shows that GAN-pretraining improves the semi-supervised performance.

Our trials showed that the impact of each enhancement greatly varies based on the dataset used. While cGANs can greatly improve the visual performance on MNIST, they have no positive impact on CIFAR10. For complex datasets, Mini-Batch Discrimination and Feature Matching improved the stability of the training by preventing mode collapse. Moreover, our tests showed that GANs increase the classification accuracy in semi-supervised classification tasks.

References

[OpenAI] SALIMANS, Tim, et al. Improved techniques for training gans. In: Advances in Neural Information Processing Systems. 2016. S. 2234-2242.
[HKang] Hyeonwoo Kang. pytorch-MNIST-CelebA-cGAN-cDCGAN. Online: https://web.archive.org/web/20180202171452/https://github.com/znxlwm/pytorch-MNIST-CelebA-cGAN-cDCGAN
[Radford] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.