Generating Realistic Facial Expressions with Bidirectional Conditional GAN

Dalia Sherman, Shir Mamia
Bar Ilan University, Ramat Gan, Israel

Abstract. We have developed a special GAN model which can generate face images according to a predefined emotion condition. We trained the model on the FER2013 dataset which consists of 35340 grayscale face images of 48X48 pixels, classified into six emotions: angry, fear, happy, sad, surprised, and neutral. For the first time Bidirectional Conditional GAN was applied to emotion-conditioned face generation, with promising results.

1 Introduction

Emotion detection is a capability that can assist in many areas such as health care and monitoring, emotional security, advertisement efficiency analysis, and is actively researched. The challenge is the shortage of large enough datasets of labeled facial expressions [1].

Generative Adversarial Networks (GANs) are able to generate high quality images such as handwriting fonts, face images, landscapes, art images, etc. [2] GAN models are able to learn the mapping from simple latent space to arbitrarily complex data distributions. However, we ought to understand the complex relationship between the latent space input to the generator and the generated images to control the types of images that are generated [3].

In the vanilla GAN model, there is no control on the types of data being generated. The conditional generative adversarial network (cGANs) involves the conditioning of class labels allowing the generated images targeted by a given label [4].

However, cGANs do not have the ability to learn inverse mapping. Therefore, one must use Bidirectional Generative Adversarial Networks (BiGANs) as an aid for learning it. BiGAN can perform the inverse mapping by adding to the generator G from the standard GAN, an encoder E which maps data x to latent representations z, as schematically shown in Fig. 1 [3].

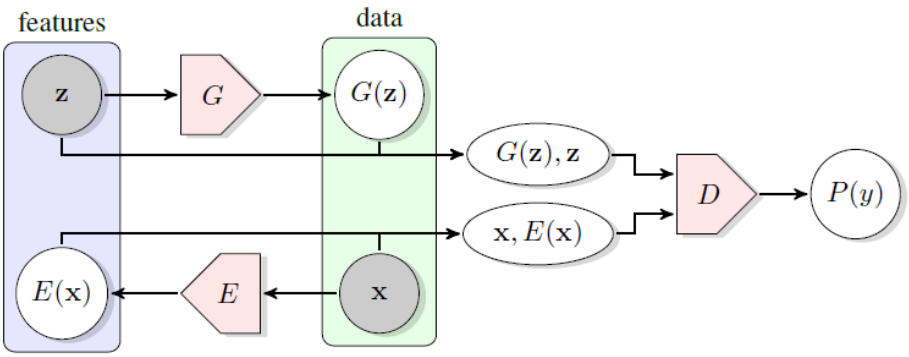

Fig. 1: The structure of Bidirectional Generative Adversarial Networks (BiGAN).

To merge the two components of conditioning and inverse mapping, we propose to use the Bidirectional Conditional GAN (BiCoGAN), which during the generation process breaks down the latent variables z and known information c. The model learns the inverse mappings to both z and c from data samples x, using the encoder that trained together with the generator and the discriminator, as schematically shown in Fig. 2 [5].

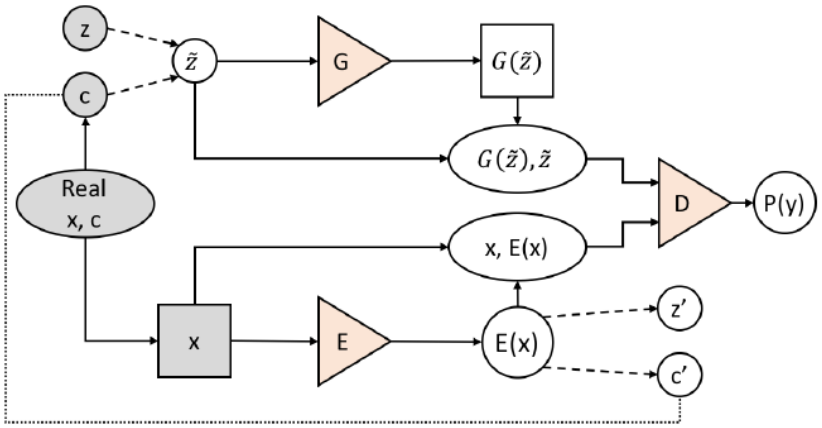

Fig. 2: Bidirectional Conditional Generative Adversarial Network.

BiCoGANs encode c more accurately and use z and c more effectively to generate samples than cGANs [5].

2 Method

Generative Adversarial Networks (GANs) technique can generate new data with the same statistics as the training set. We focused on following models:

Vanilla Generative Adversarial Networks (GANs)

The generator is aimed to create samples which are as close to real data as possible. So, the generator "fools" the discriminator. The discriminator's goal is to distinguish between real

samples and those generated by the generator as accurately as possible. The generator and the discriminator are trained with the following adversarial (minimax) objective
[3]:

$$V(D,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}[\log D(\mathbf{x})] + \underbrace{\mathbb{E}_{\mathbf{x}\sim p_G}[\log(1-D(\mathbf{x}))]}_{\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}[\log(1-D(G(\mathbf{z})))]}$$

Conditional GAN (cGAN)

Conditional GAN (cGAN) extends the vanilla GAN framework and enables the data to be conditioned on known auxiliary information e.g. object attributes, class labels, etc. Therefore, cGAN enables more control over the data generation process. we assume that some extra information y conditions both the generator and discriminator. The GAN conditioning is performed by creating an additional input layer which contains both the discriminator and generator. y is fed by this layer. $p_z(z)$ is defined as the prior input noise. $p_z(z)$ and yare combined in the generator in joint hidden representation. Flexibility in this hidden representation is enabled due to the adversarial training framework.

x and y are defined as the inputs of the discriminator's function [4].

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

Bidirectional Generative Adversarial Networks (BiGANs)

Bidirectional Generative Adversarial Networks (BiGANs) has developed as a method of learning this inverse mapping. The BiGAN model adds an encoder to the original framework (which is generator and discriminator). The encoder E maps data x to latent representations z. The discriminator is modified to incorporate not only in data space (x versus $G(z)$), but both data and latent space (tuples $(x, E(x))$ versus $(G(z), z)$) respectively. The latent component is the generator input z or the encoder output $E(x)$ [3]. The BiGAN training objective is defined by:

$$\min_{G,E} \max_D V(D,E,G)$$

where:

$$V(D,E,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\left[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(\cdot|\mathbf{x})}[\log D(\mathbf{x},\mathbf{z})]}_{\log D(\mathbf{x},E(\mathbf{x}))}\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\left[\underbrace{\mathbb{E}_{\mathbf{x}\sim p_G(\cdot|\mathbf{z})}[\log(1-D(\mathbf{x},\mathbf{z}))]}_{\log(1-D(G(\mathbf{z}),\mathbf{z}))}\right]$$

Bidirectional Conditional GAN (BiCoGAN)

For integrating two elements, we set a premise by the Bidirectional cGAN (BiCoGAN), that is efficiently dispatches z and c in the generation process and provides an encoder that learns inverse mapping, from x to z and c, trained concurrently with the generator and the discriminator. BiCoGANs has a clear upper hand over based cGANs, as far as encoding c precisely is concerned, as well as regarding exploiting z and c, and generating samples in a more

disentangled way. Therefore, the inventiveness of this model stems from its competence of conducting experiments of the synthesis of conditional facial expressions and we lay out an inventive approach Bidirectional cGAN for learning to convert an image from a domain (e.g. person's face images) conditioned on a given emotion of facial expression (e.g. happiness) to the same domain but conditioned on a different emotion of facial expression (e.g. sadness), without paired examples [5].
The distribution $p_{\tilde{z}}(\tilde{z})$ is mapped to pG by the generator $G(\tilde{z};\theta G)$ where $z\tilde{} = [z\ c]$. The generator is aimed to bring pG close to $pdata$. The distribution pdata is mapped to pE by the encoder $E(x;\theta E)$.The encoder is aimed to bring pE close to $p_{\tilde{z}}(\tilde{z})$.The discriminator decides whether the data is real or fake as $D(\tilde{z};\theta D)$ or $D(E(x),x;\theta D)$.The encoder is required to learn the inverse mapping of x to z and c while the generator is required to learn to merge both into the generation of data samples for fooling the discriminator. Practically, it is hard to achieve such optimality. In contrast to the intrinsic factors that are sampled randomly from a simple latent distribution, the extrinsic factors are more specialized in certain types of high-level information (e.g. object attributes, class labels, etc.). Therefore, their basic distribution is significantly harder to model. The extrinsic factor loss (EFL) is aimed to address this challenge. EFL is a mechanism that guides BiCoGANs to better encode extrinsic factors. In the BiCoGAN objective, the EFL does not have any explicit form due to the choice of the loss function which depends on c and on the dataset/domain. Best results are not achieved only by adding EFL to the BiCoGAN objective (for both encoding c and generating x that merges the knowledge of c). Because the training process has no information about the inherent difficulty of encoding c, the backpropagated gradients of the EFL (to the encoder) could be distorted (by those from the discriminator in the BiCoGAN framework). Hence, EFL is multiplied by an important weight γ which is called the EFL weight (EFLW).

The BiCoGAN objective is given by the following equation:

$$\min_{G,E}\max_{D} V(D,G,E) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(E(x),x)\right] + \gamma\ \mathbb{E}_{(x,c)\sim p_{data}(x,c)}\left[\mathrm{EFL}(c,E_c(x))\right] + \ \mathbb{E}_{z\sim p_{\tilde{z}}(\tilde{z})}\left[\log(1-D(\tilde{z},G(\tilde{z})))\right]$$

EFL weight (EFLW) is designed as a clipped exponentially increasing variable that is initialized by a small value: $\gamma = min(\alpha e^{pt}, \varphi)$.

α is defined as the initial value for γ and φ is its maximum value. The rate of exponential increase is controlled by p and the number of epochs (that the model has already been trained) is indicated by t [5].

3 Dataset

We applied the Bidirectional Conditional GAN model on the FER2013 dataset which consists of 35340 examples, 48x48 pixel grayscale images of faces, classified into six emotions: Angry, Fear, Happy, Sad, Surprised, Neutral.

4 Working Flow and Results

First, to understand how the GAN model works, we implemented a basic GAN model. The results we got weren't satisfying, therefore we tried a different GAN model that was also easier to convert to conditional GAN later.

From the conditional GAN we got the following results in Fig. 3:

Fig. 3: Results from applying Conditional GAN.

However, as we mentioned, our aim is to combine cGAN with BiGAN. Therefore, first we implemented cGAN and BiGAN separately but in a format that will be convenient to merge them later.
The results we got from the combined model after 18600 epochs are presented in Fig. 4:

Fig. 4: Results from applying BiCoGAN.

To improve the model, we also tried to implement Wasserstein GAN and change the discriminator model to a critic model. Unfortunately, when we tried to combine it with the current BiCoGAN model, it didn't yield good results. Without despair, we tried also to

implement another version of Wasserstein that uses gradient penalty loss. Also, in this case, the merging didn't improve the model.

After these attempts to improve the model, we decided to continue with the best model that worked so far – BiCoGAN. Also, we decided to reduce the number of emotions from 7 to 6 and to give up on "disgust", because the number of samplings was less than 10% compared to the rest.

To improve the results, we decided to experiment with the following types of data augmentations:

Operation Type	Value	Result compared to no aug (better, worse, same)
Affine_rotate	from -50 to 30	better
Crop	from 0 to 0.2	worse
Fliplr	1	same
GammaContrast	2	worse
Affine_scale	"x": (1.5, 1.0), "y": (1.5, 1.0)	worse
Affine_shear	from 0 to 0.2	same

After examining the results, we concluded that the "Affine_rotate" improved the results in the most noticeable way. After 16200 epochs we got as presented in Fig. 5:

Fig. 5: Results from applying BiCoGAN with "Affine_rotate" augmentation.

To sum up, we experimented with different models and different types of augmentations to generate the best conditional face images. We found that the best model was BiCoGAN with "Affine_rotate" augmentation.

You can find the full code on GitHub at this link.

References:

[1] G. Tesei, "Generating Realistic Facial Expressions through Conditional Cycle-Consistent Generative Adversarial Networks (CCycleGAN)," pp. 1–7.

[2] M. Chen, C. Li, K. Li, H. Zhang, and X. He, "Double Encoder Conditional GAN for Facial Expression Synthesis," *Chinese Control Conf. CCC*, vol. 2018-July, pp. 9286–9291, 2018, doi: 10.23919/ChiCC.2018.8483579.

[3] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial Feature Learning," no. 2016, pp. 1–18, 2016, [Online]. Available: http://arxiv.org/abs/1605.09782.

[4] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," pp. 1–7, 2014, [Online]. Available: http://arxiv.org/abs/1411.1784.

[5] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, "Bidirectional Conditional Generative Adversarial Networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11363 LNCS, pp. 216–232, 2019, doi: 10.1007/978-3-030-20893-6_14.