CAP6610 Machine Learning Project Proposal

Michael Francis Pérez
University of Florida
Computer and Information Science and Engineering Department
432 Newell Drive Gainesville FL 32611
michaelperez012@ufl.edu

Abstract

I propose to review and implement the paper "Generating Video with Scene Dynamics" (1), which proposes *VideoGAN* for video recognition and generation.

1 Introduction

Video generation is an important deep learning task which has applications in simulations and forecasting. (1) Similarly, video recognition (for example, action detection) has several applications in medicine, sociology, crime detection, and human-computer interaction. These tasks are interesting because the generative adversarial network (*GAN*) framework can learn each of these tasks jointly during training. In addition, although the *GAN* can generate images and voices with uncanny realism, the *GAN*'s fidelity for videos not as impressive. Video generation is a challenging task; a lot of work is left to be done before realistic videos can be generated in an unconstrained setting. (2)

2 Background

A new framework for estimating generative models via an adversarial process was developed in 2014, the *GAN* (3). Two models, a generative model G that captures the data distribution and a discriminative model D that estimates the probability that a sample $\mathbf{x}$ came from the training data rather than G, are trained jointly. G takes as input a noise vector z, sampled from a normal distribution $p_{noise}(z)$, and up-samples it into an image. D outputs a scalar probability that an input image $\mathbf{x}$ is from the real data distribution. D and G play a two-player mini-max game with value function:

$$\mathbb{E}_{x\sim p_{data}(x)}[logD(\mathbf{x})] + \mathbb{E}_{z\sim p_{noise}(z)}[log(1 - D(G(\mathbf{z}))] \tag{1}$$

The framework is termed "adversarial" because the generative model competes against an adversary, the discriminative model, that learns to distinguish between samples from the real and generated distributions. Competition causes both models to improve until the generated samples are indistinguishable from samples from the real data distribution, and the generator learns to fool the discriminator. The quality of generated images is assessed by fitting a Gaussian Parzen window to the images and then computing the log-likelihood on the test set. The MNIST (4) dataset contains images of 70,000 28×28 handwritten digits, while CIFAR-10 (5) consists of 60,000 32×32 color images from 10 classes. The log-likelihood estimates on MNIST and CIFAR-10 show competitive results to existing generative models, suggesting the *GAN* framework's viability.

To capture motion, videos can naturally be decomposed into a spatial component, which has information about the objects and scenes in a video, and a temporal component, which describes the movement of objects and the camera. (6) A two-stream CNN architecture (6) was used to establish a new state-of-the-art method for action recognition by combining a spatial and temporal recognition

stream by late fusion (7). The spatial stream operates on single frames from the input video, while the temporal stream operates on multi-frame optical flow. An optical flow is a set of displacement vector fields between two consecutive frames. The temporal stream input is formed by stacking the optical flow displacement fields between a sequence of consecutive frames. Instead of explicitly calculating optical flow before training, *VideoGAN* (1) learns motion features during training.

Three-dimensional (3D) CNNs and spatio-temporal convolutions were used to achieve a new state-of-the-art in video object recognition and scene classification, entitled Convolution 3D feature (*C3D*) (8). The authors (8) argue that only 3D convolutions preserve the temporal information of the input signals, because 2D convolution collapses the temporal information. The Net A very deep CNN architecture from (9) was adapted by replacing all 2D convolution and pooling operations with their 3D counterparts. Filter kernels of size $3 \times 3 \times 3$ that operate over space and time are used; 16-frame clips are used as input to the network. The UCF-101 dataset (10) contains 13,320 video clips labeled into 101 action classes. The primary evaluation is performed using UCF-101; the authors achieve an 11% improvement over the two-stream approach (6), which can be attributed to *C3D* modeling temporal signals better.

VideoGAN (1) is the first work to extensively investigate *GAN*'s for video. The authors design a one-stream architecture and a two-stream architecture for the generator G. The one-stream architecture uses spatio-temporal convolutions (8) to provide spatial and temporal invariance, and fractionally strided convolutions (11) to up-sample efficiently. This architecture is a variant of *DCGAN* (12) that is extended in time. The two-stream generator architecture models a static background and moving foreground according this expression:

$$G_2(z) = m(z) \odot f(z) + (1 - m(z)) \odot b(z), \tag{2}$$

where $0 \leq m(z) \leq 1$ is a mask that selects either the foreground $f(z)$ or the background $b(z)$ at each pixel and time step, and $\odot$ is element-wise multiplication. $f(z)$ is the same network as the one-stream architecture, $b(z)$ is similar to the generator in *DCGAN* (12), and $m(z)$ shares weights with $f(z)$ except for the last layer, which has one output channel. The generator outputs 64×64 pixel videos up to 32 frames long ($\sim$1 second). The discriminator network is a five-layer spatio-temporal CNN with kernels of size $4 \times 4 \times 4$. The discriminator architecture uses strided convolutions instead of fractionally strided convolutions in order to down-sample the image and the last layer outputs a binary classification (real or fake).

In *VideoGAN*, the discriminator and generator are trained via stochastic gradient descent. The Adam optimizer (13) is used with a fixed learning rate of 0.0002 and momentum of 0.5. The latent code $z \in \mathbb{R}^{100}$ is sampled from a normal distribution, and a batch size of 64 is used. After every layer in the generator other than the output layer, there is a batch normalization layer (14) and then a ReLU activation function. In the discriminator, batch normalization is used with leaky ReLU (15).

3 Proposal

In one experiment, the authors (1) evaluated *Video-GAN*'s performance classifying actions on UCF-101. 5,000 hours of unlabeled Flickr videos (16) are pre-processed, through stabilization of camera motion using SIFT and RANSAC, and normalization. The model is trained on the large unlabeled dataset; then, the discriminator is fine-tuned on a relatively small set of labeled videos. The discriminator's output function is modified to be a K-way softmax classifier instead of a binary classifier, and dropout (17) is used as the second-to-last layer to reduce overfitting. This fine-tuning procedure yields a classification accuracy of 52.1% on UCF-101.

I propose to implement *Video-GAN* then replicate two experiments to evaluate its performance. I will use the unlabeled Flickr videos datasets (publicly available at `http://www.cs.columbia.edu/~vondrick/tinyvideo/`) to train *Video-GAN* to generate samples for each dataset, then qualitatively evaluate their fidelity. I will also evaluate the discriminator's performance in classifying actions on the UCF-101 dataset, using the same fine-tuning procedure and hyperparameters from the paper. I will aim to reproduce the result (52.1% accuracy).

References

[1] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *CoRR*, vol. abs/1609.02612, 2016. [Online]. Available: http://arxiv.org/abs/1609.02612

[2] A. Clark, J. Donahue, and K. Simonyan, "Efficient video generation on complex datasets," *CoRR*, vol. abs/1907.06571, 2019. [Online]. Available: http://arxiv.org/abs/1907.06571

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, O. Sherjil, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[4] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[5] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," no. 0, 2009.

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *CoRR*, vol. abs/1406.2199, 2014. [Online]. Available: http://arxiv.org/abs/1406.2199

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[8] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[10] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[11] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," pp. 2528–2535, 2010.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.06434

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[15] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[16] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, "The new data and new challenges in multimedia research," *CoRR*, vol. abs/1503.01817, 2015. [Online]. Available: http://arxiv.org/abs/1503.01817

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html