

Redes Generativas Adversarias

Aplicadas a Generación de Imágenes

F. Atenas¹ F. Sanhueza¹ C. Valenzuela¹

¹Departamento de Ingeniería Matemática
Universidad de Chile

8 de Julio, 2017

Problema

Nuestro trabajo consistió en el estudio del uso de redes adversariales generativas o GANs.

Estas son un modelo para generar datos de un proceso cuya densidad se encuentra implícita. Presenta ciertas ventajas respecto a otros modelos:

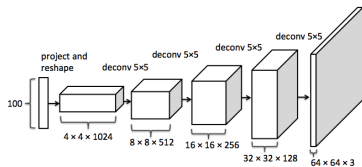
- 1 El procedimiento para generar una muestra se hace de forma paralela.
- 2 Pocos requerimientos de dimensionalidad y sobre la función G generadora.
- 3 El procedimiento es insesgado y asintóticamente consistente.
- 4 Generan muestras en un paso, a diferencia de algoritmos basados en cadenas de Markov.

Consta de dos funciones diferenciables G y D , que son representadas por redes neuronales convolucionales. G recibe un ruido aleatorio z y a partir de él genera una muestra. D recibe una muestra y debe determinar si esta fue generada por G o si es un dato real. Así ambas participan en un juego de suma cero definido por:

$$\max_G \min_D -\mathbb{E}_{x \sim p_{\text{datos}}}(\log(D(x))) - \mathbb{E}_{z \sim p_z}(\log(1 - D(G(z))))$$

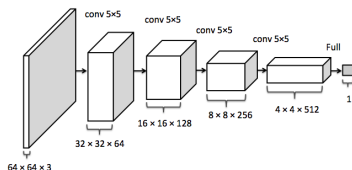
En el equilibrio de Nash de este juego resulta que G replica la distribución latente de manera perfecta.

Generador:
4 capas de convolución traspuesta.



Funciones de activación ReLU y de salida tanh.

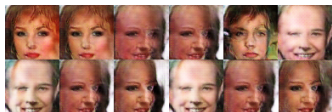
Discriminador:
4 capas de convolución.



Funciones de activación LReLU y salida sigmoide.

$$\text{Costos: } -\frac{1}{n} \sum_{j=1}^n (y_j \ln(\sigma(x_j)) + (1 - y_j) \ln(1 - \sigma(x_j)))$$

Resultados



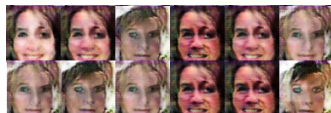
(a) $U(-1, 1)$, LReLU, 10 épocas.



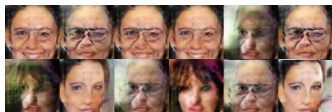
(b) $U(-1, 1)$, LReLU, 20 épocas.



(c) $\mathcal{N}(0, 1)$, LReLU, 10 épocas.



(d) $\mathcal{N}(0, 1)$, LReLU, 20 épocas.



(e) $U(-1, 1)$, ReLU, 10 épocas.



(f) $\mathcal{N}(0, 1)$, ReLU, 20 épocas.

Conclusiones

- Modelo posee una forma simple de muestreo de una distribución desconocida, sujeta a la capacidad del generador.
- Implementación flexible a modificaciones.
- El tiempo de entrenamiento está lejos del tiempo de convergencia del algoritmo.
- Generación de imágenes con atributos humanos, pero que pueden ser mejoradas para ser más realistas.
- Aún es un misterio porque las GANs prefieren generar imágenes con pocos modos diferentes.

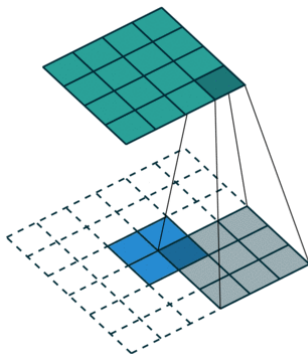
Referencias I

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative Adversarial Nets. *arXiv preprint arXiv:1406.2661v1*.
- [2] I. Goodfellow, (2017). Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160v4*.
- [3] A. Radford, L. Metz, and S. Chintala (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Network. *arXiv preprint arXiv:1511.06434v2*.
- [4] *Image Completion with Deep Learning in TensorFlow* (2017, 25 Junio). Recuperado de:
<http://bamos.github.io/2016/08/09/deep-completion/>

Referencias II

- [5] *Improving the way neural networks learn* (2017, 27 Junio). Recuperado de: http://neuralnetworksanddeeplearning.com/chap3.html#the_cross-entropy_cost_function.
- [6] Z. Liu, P. Luo, X. Wang, and X. Tang (2015). Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV)*.
- [7] *Convolution arithmetic tutorial* (2017,6 Julio). Recuperado de: http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html#no-zero-padding-unit-strides-transposed.

Figure: Representación Gráfica de la Convolución Traspuesta (Deconvolución). Figura obtenida de [7].



Derivada de la *sigmoid cross entropy*: Si $\zeta = \sum_j w_j u_j + b$ es la suma con pesos de los inputs u de una capa, entonces

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_u \frac{\sigma'(\zeta) u_j}{\sigma(\zeta)(1 - \sigma(\zeta))} (\sigma(\zeta) - y) = \frac{1}{n} \sum_u u_j (\sigma(\zeta) - y)$$

donde y es el label esperado (1 para imágenes reales y 0 para las falsas).