Exploring uses of Normalizing Flows for Document Image Processing: Text Super-resolution and Binarization

Giorgos Sfikas, George Retsinas and Basilis Gatos   NCSR "Demokritos"

DEMOKRITOS

DAS 2022
May 22-25, La Rochelle

Introduction to the Normalizing flow framework

- $p_X(\cdot)$ is to be estimated given $X = \{x_1, x_2, \cdots, x_N\}$ [1, 2]

$$p_X(x) = p_U(f_\theta(x))|det\frac{\partial f_\theta}{\partial x}(x)|$$

- How? *Transform observations* so that they follow p_U..
- ..using a *diffeomorphism* $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^D$
- f_θ is defined as a neural network $f_\theta(x) = f^K \circ f^{K-1} \circ \cdots \circ f^1(x;\theta)$.
- learning = *finding the optimal network parameters that transform X so that it follows $p_U(\cdot)$*
- Framework is recastable to use in a supervised task [3]:

$$p_{Y|X}(y|x) = p_U(f_\theta(y|x))|det\frac{\partial f_\theta}{\partial x}(y|x)|,$$

with a *maximum likelihood objective*:

$$\arg\max_\theta \log \mathcal{N}(f_\theta(y|x)) + \sum_{k=1}^{K} \log |det\frac{f^k}{z^k}(z^k|x;\theta)|,$$

where we set $z^0 = u$, $z^K = y$, $z^k = f^k(z^{k-1}|x)\ \forall k \in [1, K]$, $p_U = \mathcal{N}$

Flow layers

Flow layers compose f_θ. They must be..

- expressible
- invertible
- cheap to compute (evaluating $det\frac{\partial f_\theta}{\partial x}(x)$ can be a serious bottleneck!)

E.g."Affine coupling" ($z^k = \{z^k_A, z^k_B\}$ a partition of z^k & f^k_θ is defined in terms of $f^k_{\theta,s}$,$f^k_{\theta,t}$

$$z^k = f^k_\theta(z^{k-1}|x) \implies \begin{cases} z^k_A = z^{k-1}_A \\ z^k_B = z^{k-1}_B \circ exp(f^k_{\theta,s}(z^{k-1}_A|x)) + f^k_{\theta,b}(z^{k-1}_A|x) \end{cases} \quad (1)$$

Note that f^k_θ is (easily!) invertible, while $f^k_{\theta,s}$,$f^k_{\theta,t}$ can be arbitrarily complex and difficult to invert [4].

Text SR and Binarization as Normalizing Flows

- Text Super-Resolution is cast as a supervised problem: Estimate HR image given LR image
- Likewise for binarization: Estimate the binarized image given the unprocessed image
- Results come as a probability density function of the output image given the input image
- Inference is performed by sampling:

$$y = f_\theta^{-1}(z|x), z \sim \mathcal{N}(0, \tau)$$

Figure 1: Binarization results: Original images and binarization results for different "temperatures" τ.

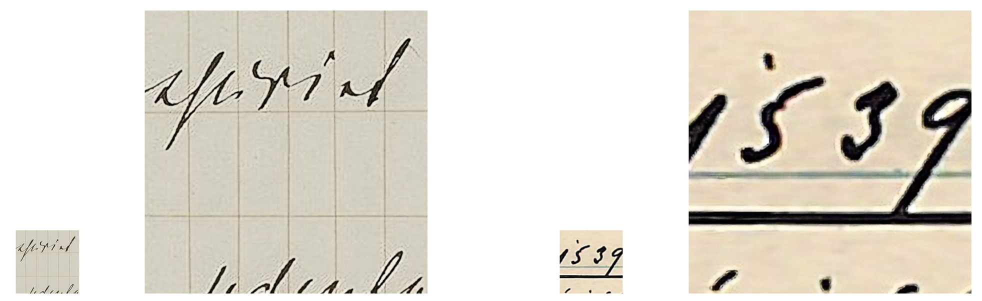

Figure 2: Super-resolution results: Original images and super-resolved images (τ=0.7).

Conclusion

- A probabilistic setting combined with neural networks is an attractive option!
- Multiple outputs per input are possible, and we know which one is the most likely
- Probabilistic framework lends easily to nice extensions (e.g. combine with a task-specific prior)
- Future work: More research on appropriate flow models for document image processing

References

[1] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research (JMLR)*, 22(57), 2021.

[2] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions in Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[3] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In *Proceedings of the IEEE European Conference in Computer Vision (ECCV)*, pages 715–732. Springer, 2020.

[4] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using RealNVP. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.