Exploring Uses of Normalizing Flows for Document Image Processing: Text Super-Resolution and Binarization

Giorgos Sfikas[1,3,4], George Retsinas[2], and Basilis Gatos[3]

[1] CIL/IIT, NCSR "Demokritos", Greece
[2] School of ECE, NTUA, Greece
[3] Dpt. of CS and Engineering, University of Ioannina, Greece
[4] Dpt. of Surveying and Geoinformatics Engineering, Univ. of West Attica, Greece

Abstract. Normalizing flows are powerful models that elegantly combine invertible neural networks with probabilistic modeling. We explore uses of the normalizing flow framework for two document image processing tasks: Text Super-Resolution and Binarization.

Keywords: Normalizing Flows, Text Super-Resolution, Binarization

1 Introduction to Normalizing flows

In the normalizing flow (NF) framework [6], a probability density function $p_X(\cdot)$ is sought to be estimated given a finite set of samples $X = \{x_1, x_2, \cdots, x_N\}$ known to come from that distribution. The core adea is to express the available observed data in terms of a distribution $p_U(\cdot)$, that is termed the "base" distribution and is typically a standard isotropic Gaussian. A diffeomorphism (a smooth, bijective function) $f : \mathbb{R}^D \to \mathbb{R}^D$ is assumed to transform data X into images $\{f_\theta(x_1), f_\theta(x_2), \cdots, f_\theta(x_N)\}$, that are required to follow the (typically) Normal distribution $p_U(\cdot)$, and images and pre-images share the same dimensionality, denoted as D. θ is a set of parameters that define the transformation. The term "normalizing flow" stems from exactly this requirement; f_θ is responsible for creating data that are normally distributed, and in this sense it is "normalizing". Transformation function f_θ is defined as a neural network, and learning the data is performed by finding the optimal network parameters that transform X as required. Concerning notation, in what follows we will write $f_\theta(x)$ or $f(x;\theta)$ or simply f to refer to the same transformation.

Formally, we can write [1]:

$$p_X(x) = p_U(f_\theta(x))|det\frac{\partial f_\theta}{\partial x}(x)|, \quad (1)$$

where we use the change-of-variables formula between pdfs, θ are the parameters that define the transformation f, and $\partial f_\theta(x)/\partial x$ is the Jacobian matrix for f_θ. A very important constraint over f_θ is that it needs to be bijective. In practice, network f_θ needs to be structured so as to have both a Jacobian and an inverse

f_θ^{-1} that are easily computable. If network f_θ is defined as a composition $f_\theta(x) = f^K \circ f^{K-1} \circ \cdots \circ f^1(x;\theta)$, training the normalizing flow is tantamount to solving the following maximum likelihood problem:

$$\arg\max_\theta \log \mathcal{N}(f(x;\theta)) + \sum_{k=1}^{K} \log |det \frac{f^k}{z^k}(z^k;\theta)| \tag{2}$$

where we used $z^0 = u$, $z^K = x$, $z^k = f^k(z^{k-1})\ \forall k \in [1, K]$.

The standard formulation of Normalizing flows described above, fits the unsupervised setting of density estimation perfectly. For a supervised learning setting, where we have pairs of source $X = \{x_1, x_2, \cdots, x_N\}$ and target objects or labels $Y = \{y_1, y_2, \cdots, y_N\}$, this standard paradigm can be extended to a formulation of conditional Normalizing flows [6, 4]. Under this setting, transformation f is required to map from $y|x$ to $z|x$, i.e. now targets are mapped to a latent space by means of the normalizing flow, while all are conditioned on the source data x. It is then straightforward to rewrite the density of eq. 1 as a conditional density:

$$p_{Y|X}(y|x) = p_U(f_\theta(y|x))|det \frac{\partial f_\theta}{\partial x}(y|x)|, \tag{3}$$

and the maximum likelihood objective of eq. 2 in its conditional iteration as:

$$\arg\max_\theta \log \mathcal{N}(f(y|x;\theta)) + \sum_{k=1}^{K} \log |det \frac{f^k}{z^k}(z^k|x;\theta)|, \tag{4}$$

where we now set $z^0 = u$, $z^K = y$, $z^k = f^k(z^{k-1}|x)\ \forall k \in [1, K]$. Learning a model on data X, Y can hence be performed by optimizing eq. 4 given the available data and w.r.t. the transformation parameters θ. Transformation f is diffeomorphic thus differentiable by assumption, hence in practice we can choose to use any standard gradient-based optimizer (e.g. SGD, Adam).

Interestingly, flows have been shown to lead to state-of-the-art performance in a number of tasks, using only a Maximum Likelihood criterion to train [3, 4]. Other models often require multiple priors that entail requiring hyperparameters that weight the importance of each prior w.r.t. the likelihood term. These play often a critical role in the success of the architecture in practical applications. Further useful traits of NFs include: efficient and exact density evaluation; potential memory savings; an inherently probabilistic formulation, without many of the difficulties typically associated to probabilistic modeling and other generative models [3].

2 Formulation of Text Super-resolution and Binarization as Normalizing Flows

At a high-level, we follow the way the conditional architecture of SRFlow [4] is built, and we use the same way flow layers are grouped into a cascade of L levels.

Flow level are each related to a spatial resolution, in particular $H/2^l \times W/2^l$, where $H \times W$ stands for the initial resolution. A level can broken down into K groups of flow layers ("flow-steps" [4]). In turn, each flow-step is made up of the following four flow layers: actnorm, 1×1 convolution, affine injector and conditional affine coupling. For our super-resolution application we use a number of levels $L = 3$, and for the binarization application we use a single level $L = 1$, hypothesizing that the binarization problem is less complex / demanding than super-resolution. We use patches sized 160×160 pixels for our experiments. In super-resolution, we sub-sample the training patches to 40×40 to create low-res / high-res pairs. We use a pre-trained RRDB backbone in both cases. Inference is performed as a process of sampling from the learned density, conditioned on the input, i.e. the low-res image or the non-binarized image respectively. In figures 1 we show 2 we show visual results. Regarding the employed datasets for training and testing, we have used the DIBCO binarization competition datasets [7] and the new "PIOP-DAS" dataset [8].

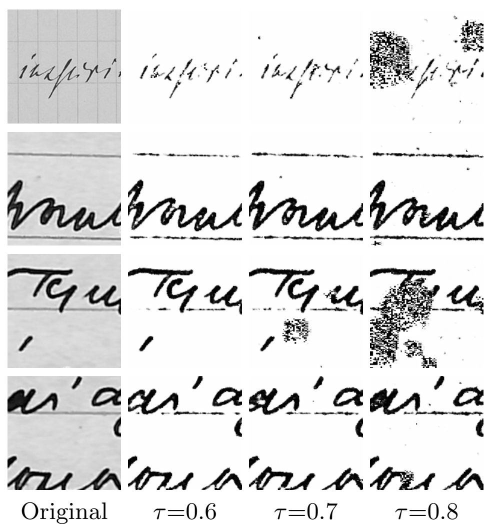

Fig. 1. Binarization results: Original images and binarization results for different "temperature" hyperparameter values τ.

3 Future work

After obtaining the reported first very preliminary though somewhat promising results, we plan to continue our research on NFs along the following axes: First, setup sets of experiments on both considered problems, evaluate numerically the results, and compare to state-of-the-art methods. Concerning super-resolution, consider integrating with a shape-based approach for the prior, leading to an extra loss term (e.g. [2], or the recent [5]). Also, test more challenging SR upsampling scales. We also envisage using SR combined with binarization, in a

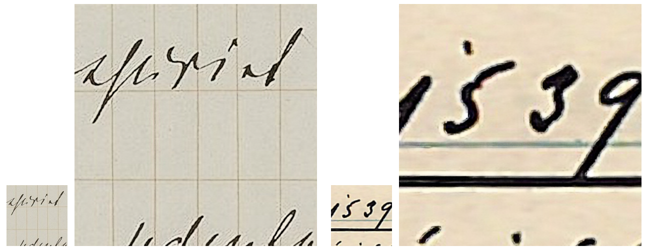

Fig. 2. Super-resolution results: Original images and super-resolved images (τ=0.7).

scenario where a binarization components may aid in avoiding to super-resolve areas that are unimportant (background) or noisy (jpeg artifacts), or aid in properly evaluating the result (by disregarding background from SR result evaluation).

Acknowledgments

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls "RESEARCH - CREATE - INNOVATE" (project *Culdile* - code T1E*Δ*K-03785), and "OPEN INNOVATION IN CULTURE" (project *Bessarion* - T6YB*Π*-00214).

References

1. Dinh, L., Krueger, D., Bengio, Y.: NICE: non-linear independent components estimation. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2015)
2. Giotis, A.P., Sfikas, G., Nikou, C., Gatos, B.: Shape-based word spotting in handwritten document images. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR). pp. 561–565. IEEE (2015)
3. Kingma, D., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NIPS (2018)
4. Lugmayr, A., Danelljan, M., Van Gool, L., Timofte, R.: SRFlow: Learning the super-resolution space with normalizing flow. In: ECCV. pp. 715–732. Springer (2020)
5. Nakaune, S., Lizuka, S., Fukui, K.: Skeleton-aware text image super-resolution. In: BMVC (2021)
6. Papamakarios, G., Nalisnick, E., Rezende, D., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. JMLR **22**(57) (2021)
7. Pratikakis, I., Zagoris, K., Kaddas, P., Gatos, B.: ICFHR2018 competition on handwritten document image binarization contest. In: ICFHR. pp. 1–1 (2018)
8. Sfikas, G., Retsinas, G., Giotis, A.P., Gatos, B., Nikou, C.: Keyword spotting with quaternionic ResNet: Application to spotting in Greek manuscripts. In: Proceedings of the International Workshop on Document Analysis Systems (DAS) (2022)