EECS 6322 Week 10, Paper 1
Residual Flows for Invertible Generative Modeling

Shivani Sheth (Student #: 218011783)
shivs29@yorku.ca

I. Summary and contributions

The paper proposes a framework, known as Residual Flows, that uses a "Russian roulette" estimator to give a tractable unbiased estimate of the log density. It also reduces the memory requirements during training, and improves the invertible residual blocks by using different activation functions that prevent derivative saturation and generalize the Lipschitz condition to induced mixed norms.

Normalizing Flows use flow-based models to fit high-dimensional continuous data into a family of distributions. The models use the change of variables theorem that constructs flexible distributions which allow tractable exact sampling and efficient evaluation of its density. Thus, the models use invertibility to generate realistic images and can achieve a high density estimation performance. The existing flow based models use transformations which scale to high-dimensional data using specialized architectures such as coupling blocks, which impose a strong inductive bias that can hinder their application in other tasks, such as learning representations that are suitable for both generative and discriminative tasks. This paper proposes a model, which is based on residual networks that can be made invertible by simply enforcing a Lipschitz constraint, allowing to use a very successful discriminative deep network architecture for unsupervised flow-based modeling. It thus introduces Residual Flows, which is a flow-based generative model that produces an unbiased estimate of the log density and has memory-efficient backpropagation through the log density computation.

A few important topics in the domain include maximum likelihood estimation, change of variables theorem, and invertible residual networks (i-ResNets). The maximum likelihood estimation calculates the KL Divergence of the data given by: $\nabla_\theta D_{KL}(p_{data}||p_\theta) = \nabla_\theta E_{x\sim p_{data}(x)}[log\ p_\theta(x)] = E_{x\sim p_{data}(x)}[\nabla_\theta log\ p_\theta(x)]$*. The change of variables theorem captures the change in density of the transformed samples with an invertible transformation* f*, given by:* $log\ p(x) = log\ p(f(x)) + log|det\frac{df(x)}{dx}|$*. The residual networks include simple transformations of the form* $y = f(x) = x + g(x)$*, which is invertible by the Banach fixed point theorem if* g *is contractive. Thus, by applying i-ResNets to the change-of-variables theorem, we get the modified equation as* $log\ p(x) = log\ p(f(x)) + tr(\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}[J_g(x)]^k)$*, where* $J_g(x) = \frac{dg(x)}{dx}$*. This approach requires the decoupling of both the objective and estimation bias.*

The proposed architecture relies on randomization to derive an unbiased estimator that can be computed in finite time, since the evaluation of the exact log density function in the i-ResNets equation requires infinite time due to the power series. The unbiased estimator can be applied infinitely many times to obtain the "Russian roulette" estimator. This estimator has been taken as a base and has been modified to an unbiased log density estimator used by the proposed model with a constraint that $p(N)$ *must have support over all of the indices. Thus, the unbiased log density estimator is given by:* $log\ p(x) = log\ p(f(x)) + E_{n,v}[\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k} \frac{v^T[J_g(x)^k]v}{P(N\geq k)}]$*, where* $n \sim p(N)$ *and* $v \sim N(0, I)$*. Additionally, a Skilling-Hutchinson trace estimator has been used to estimate the trace of the matrices* J_g^k*. The variance of the Russian roulette estimator is small when the infinite series exhibits fast convergence, and* $p(N)$ *does not need to be tuned for variance reduction. This forms the core of Residual Flow since maximum likelihood training can be performed by back propagating through the unbiased log density estimator to obtain the unbiased gradients, which can allow for more expressive networks to be trained.*

The memory cost during training is $O(n.m)$ *where* n *is the number of computed terms and* m *is the number of residual blocks in the entire network, which can lead to large memory consumption for a large random sample of* n*. To reduce the memory consumption during training, the authors have applied the gradients as a power series derived from a Neumann series. Thus, the modified unbiased log-determinant gradient estimator with* $Lip(g) < 1$ *and* N *random variable with support over positive integer, reduces the memory requirement by a factor of* n*. This is useful while using an unbiased estimator since the memory remains constant regardless of the number of terms drawn from* $p(N)$*. The authors propose that the memory constraints can be further reduced by partially performing backpropagation during the forward pass. The derivative of the loss function can be split into a scalar and a vector term, out of which the vector term can be computed along with the forward pass, thus releasing the memory for the computation graph. Then, during backpropagation, this vector term can be multiplied by the scalar term, which reduces memory by a factor of* m *to* $O(1)$ *with negligible overhead.*

To avoid derivative saturation, the authors propose the LipSwish activation function given by: $LipSwish(z) = Swish(z)/1.1 = z\sigma(\beta z)/1.1$*, where* σ *is the sigmoid function. This activation function is a modification of the Swish function that exhibits a less than unity Lipschitz property. In*

the experiments used in the paper, β is parameterized to be strictly positive by passing it through softplus. The Residual Flow models were evaluated on datasets including MNIST, CelebA-HQ, and CIFAR-10. The models were trained with the standard batch size of 64 for MNIST and CIFAR-10, and a batch size of 3 per GPU for CelebA-HQ. When evaluated on the bits per dimension metric on the benchmark datasets MNIST, CIFAR-10, downsampled ImageNet, and CelebA-HQ, the Residual Flow model achieved competitive performance to state-of-the-art flow-based models on all datasets. In terms of the sample quality, the model was evaluated on the FID scores metric, where it improves on the i-ResNets and PixelCNN architectures, and outperforms the Glow model which has double the number of layers. For generating visually appealing images, the model performs reduced entropy sampling on CelebA-HQ 256, but the samples do not converge to the mode of the distribution.

An ablation study has also been performed on the model, which proves that even in cases where the Lipschitz constant and bias are relatively low, a significant improvement is obtained from using the unbiased estimator. It is also concluded that the LipSwish activation function not only converges much faster but also results in better performance as compared to ELU or softplus, especially in cases with high Lipschitz. Residual Flows was also experimented on a joint training of continuous and discrete data to learn a hybrid model including both a generative model and a classifier. A weighted maximum likelihood objective given by $E_{(x,y)\sim p_{data}}[\lambda log\ \ p(x) + log\ \ p(y|x)]$ was used, where λ is a scaling constant. The proposed model outperforms its counterparts on both pure classification and hybrid modeling, thus proving its efficiency as a hybrid model.

II. Strengths

The work proposed by the paper introduces stochasticity to construct tractable flow-based models that rely on exact log-determinant computations, which opens up a new design space of expressive but Lipschitz-constrained architectures. The proposed model also reduces the memory required during training by using an alternative infinite series for the gradient. The unbiased estimator as proposed by the authors using a "Russian roulette" method individually adds significantly to the performance of the model.

III. Weaknesses

The samples extracted from the reduced entropy sampling performed on the CelebA-HQ 256 by the Residual Flows model do not converge to the mode of the distribution, and hence the images generated by the model are not as visually appealing as the images generated by other state-of-the-art models. Since the $p(x)$ and $p(z)$ distributions are not the same for Residual Flows, temperature annealing also cannot be performed on the model to generate better looking images.

IV. Correctness

The claims and empirical methodology of the paper are correct and are supplemented by strong theoretical grounding and comparison studies on several datasets such as MNIST, CIFAR-10, downsampled ImageNet, and CelebA-HQ against various architectures. The theorems proposed by the paper for the unbiased estimator using a "Russian roulette" method, and to reduce the memory constraints of the model have been constructed by modifying the existing concepts, which have been proved to be efficient by the experiments conducted in the paper.

V. Clarity

The paper is well written in general with clear formatting of theoretical concepts along with the explanation of the model architecture. The theoretical proofs have also been elaborated which supplement the clarity of the work proposed by the authors. In addition, the illustrations and the tabular results given by the comparison between different architectures support the results discussed by the authors and aid in the understanding of the reader.

VI. Relation to prior work

Prior work in the field of 'Estimation of Infinite Series' include estimating limiting behavior of optimization problems, solving of stochastic differential equations, ray tracing for rendering paths of light, and using Chebyshev polynomials to estimate the spectral functions of symmetric matrices. These works differ from the estimated quantities proposed by the authors since the Jacobian computed in the power series proposed in this paper is not symmetric. Additionally, the prior work also made assumptions on p(N) in order for it to be applicable to general infinite series, while the work proposed in the paper made a different set of assumptions requiring only that p(N) has sufficient support. In the field of 'Memory-efficient Backpropagation', the prior works focus on the path-wise gradients from the output of the network, whereas the work proposed in the paper focuses on the gradients from the log-determinant term in the change of variables equation specifically for generative modeling.

VII. Reproducibility

Enough details in terms of theoretical concepts and their implementations have been provided by the authors to reproduce the major results of the work proposed by the authors. The code implementation of the work has also been provided by the authors in PyTorch which will supplement the reproducibility of the work.

VIII. Additional comments

A discussion on the applications of the model where the model would be best suited for tasks in various real-life domains would supplement the work proposed by the authors.