UNIVERSITÉ CÔTE D'AZUR

MSc. DSAI

Variational autoencoders for image deconvolution in fluorescence microscopy

Internship report

Presented and defended on August 26, 2022

par

Sai MUTTAVARAPU

Academic tutor: Laure Blanc-Féraud and Luca Calatroni

I3S and Inria

i3s sophia antipolis

Inria

Acknowledgements

Firstly, I would like to express my indebtedness appreciation to my internship coordinators Dr. Laure BLANC-FERAUD, and Dr. Luca CALATRONI for giving me permission for internship, guidance, advice and encouragement in the overall preparation of this report. The time to time meetings and availability for updates has been a major contribution in my work. Also, I would like to thank the PhD student Vasiliki STERGIOPOULOU for helping me in all possible ways during this period. The internship would not have been possible without the immense support and constant guidance regarding the project and report from the mentioned supporters.

Furthermore, I want to express my gratitude to my Academic Supervisor, Prof. Michel Riveill, Professor of Computer Science UCA and Computer scientist in the MAASAI team shared by INRIA and the I3S Laboratory.

Table des matières

Abstract

Recently, the progress of deep learning approaches has been enormous and has become very useful in all research areas, such as biomedical image reconstruction. Improving the spatial resolution of images acquired by standard microscopes is a rather challenging task. Due to the physical barriers imposed by light diffraction phenomena, images acquired in fluorescence microscopy setups are typically blurry and corrupted by electronic and photon-counting noise. Removing noise and blur from observed noisy and blurry images is the motivation in this report. By formulating the problem as an inverse image reconstruction problem the task is thus to reconstruct the desired image from the noisy and blurry data by using the generative approaches such as variational autoencoders (VAE)[1]. The architecture of the variational autoencoder is helpful to estimate the distribution of the unknown, noise- and blur-free data which we use to draw samples which are compared to the given measured data. For this sake and for the particular application of covariance-based fluorescence microscopy recently considered in [2], we propose a new approach which is able to deblur and denoise the given input, up to a certain extent and we call it "Div-Blurring". We detail the modelling considered and show some possible hybrid variations where model-based priors are combined with fully data driven models to guarantee meaningful solutions.

Keywords : Deep learning, biomedical imaging, generative approaches, variational autoencoders, image reconstruction.

Institute and Team

This research is conducted in in the MORPHEME team, a joint research team between Inria, CNRS and Université Côte d'Azur (UCA), affiliated with Inria SAM, Computer Science+Signals+Systems Laboratory (I3S) and Institute of Biology Valrose (iBV)

INSTITUTES : Inria, I3S and iBV

Inria (National Institute for Research in Digital Science and Technology) : A French national research institution focusing on computer science and applied mathematics.

I3S (Computer Science Signals Systems Laboratory) : The I3S laboratory is one of the largest information and communication science laboratories in the French Riviera. it was one of first ones to settle down on Sophia Antipolis Science and Technology Park. It consists of a little less than 300 people.

iBV (Institute of Biology Valrose) : Made up of 28 research teams, bringing together around 300 people from all over the world. The iBV teams study fundamental questions in life sciences and health such as the biology of organ development, the molecular pathology of cancer, diabetes, obesity, reproductive biology, etc.

TEAM : MORPHEME

The research team MORPHEME was created in 2010, and in 2013 it got the status of Equipe-Projet Commune (EPC). It was renewed in 2017.

Objectives : Characterize and model the morphological properties of biological structures from the cell to the supra-cellular scale with the intent of providing a better understanding of the development of normal tissues and a characterization at the supracellular level of pathologies such as the Fragile X syndrome, Alzheimer or diabetes.

Motivation : The understanding of morphological and topological aspets in mesoscopic structures have a key influence on the functional behavior of organs and living entities.

Research Axes : Imaging, Feature extraction, Interpretation/Classification, Modeling.

1

Introduction

Biomedical image reconstruction is one of the most important and fundamental components of biomedical imaging, as in some cases the acquired images are not clear enough to extract with high precision every detail we are interested in.

An important application of biomedical imaging is the visualization of cells, molecules and other related biological samples using microscopes. Microscopy imaging can help to see objects that are impossible to see with the human eyes. Fluorescence microscopy is a specific class of techniques where the objects are labelled with fluorescent dyes, i.e. photoactivable proteins which will emit the light with different wavelength when we illuminate the light on specimen with a specific wavelength. That emitted light is then captured by a detector or camera. By this approach it is thus possible to capture the cell structures of interest in the specimen.

Samples acquired by means of fluorescence microscopy techniques are blurred due to the diffraction of light. This process can be modelled mathematically as a convolution of the object with a certain blurring source specific of the microscope called Point Spread Function (PSF). This can also be seen visually in Figure 1.1 [3].

The observed images acquired by the microscope are the sum of a blurred version of the desired sample and noise perturbations. This is expressed in the Figure 1.2 with microtubules data. The noise can be Gaussian or Poisson, in this report we considered Gaussian noise only. The goal of mathematical image reconstruction in this context is thus to enhance the given images so as to extract with high precision small details of the

micro organisms from the observed images.

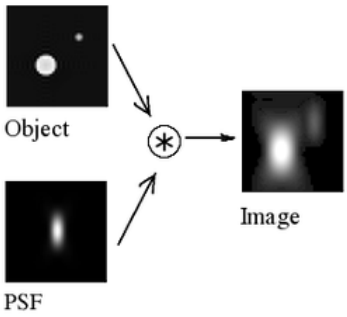

FIGURE 1.1 – Image formation in a confocal microscope. The acquired distribution arises from the convolution of the real light sources with the PSF[3].

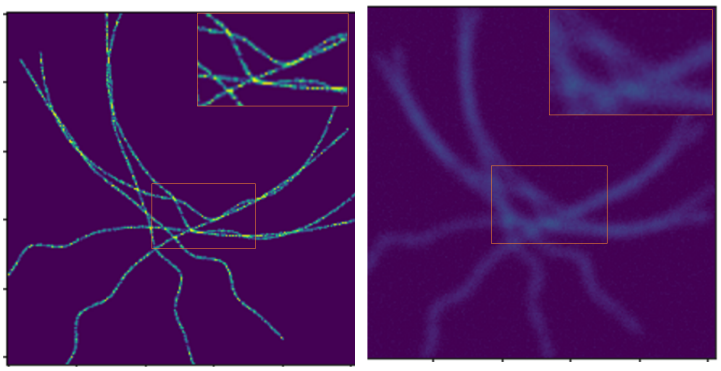

FIGURE 1.2 – Simulated images of microtubules. LEFT : The structure of True microtubule (the true image). RIGHT : Imaged microtubule, with blur and noise version by the PSF (the observed image).

We will now explain the general inverse problems framework along with some basic principle of fluorescence microscopy and deep learning techniques in this Chapter-1. Then, in Chapter-2, we will explain a method based on Variational Auto Encoders (VAE) called DivNoising which helps to denoise the images. Our proposed approach "DivBlurring" (Diversity DeBlurring) is mentioned in Chapter-3 along with some proposed model-driven improvement favouring specific properties of the solution by means of tailored regularisations. The results of DivBlurring are presented in Chapter-4 along with some more details on the VAE architecture used to train the deep learning model.

1.1 Inverse problems

Estimating learnable parameters or data from the inadequate observations can be referred as solving a inverse problem [4]. The observed data often contain incomplete information which is not sufficient to proceed in biomedical research. Generally this happens when capturing the data by measurement devices with some physical limitations. In such cases the inverse problems can help to estimate the enhanced version of given data.

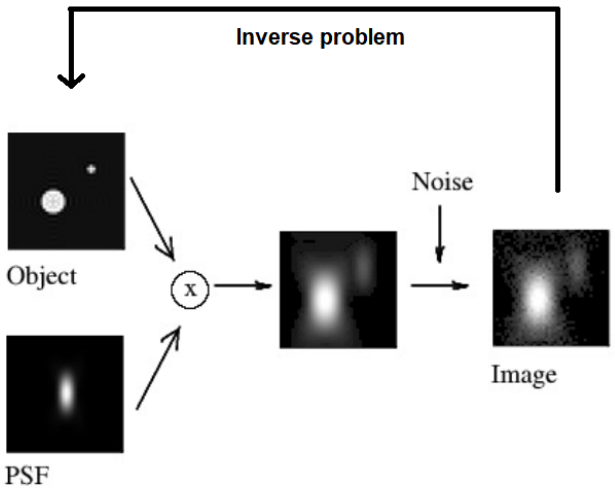

FIGURE 1.3 – the obtained image is the result of blurring and noising degradation processes from the original ground truth object[3].

From a mathematical point of view, we denote by $\mathbf{y} \in Y$ the observed noisy and blurred image, by $\mathbf{x} \in X$ the ground truth image and by $A : X \to Y$ the forward operator describing the physical degradation model (for instance, the convolution process) associating $\mathbf{x}$ to $\mathbf{y}$. The goal of an inverse problem is to find $\mathbf{x}$ for the given observation $\mathbf{y}$.

In its general form, an inverse problem can be expressed as following

$$\mathbf{y} = A\mathbf{x} + \mathbf{n} \tag{1.1}$$

where :

$\mathbf{y}$ (observation) = image corrupted by physical means : optics, radar, laser, IR, magnetic

field, X rays, ultrasounds.
A (the operator) : operator which links the observations to the ground truth.
$\mathbf{x}$ (ground truth) : the (unknown) image (we want to reconstruct).
n (noise) : random part in the observation process describing noise.

A standard approach used to solve the inverse problem in Eq. (1.1) is to consider the least-square minimisation. In order to incorporate further properties on the desired solution, it is also possible adding an additional term called the regularizer (also called penalty) so as to obtain the following regularised approach :

$$\hat{\mathbf{x}} \in \arg\min_{x} \|A\mathbf{x} - \mathbf{y}\|_2^2 + R(\mathbf{x}) \tag{1.2}$$

where $\|A\mathbf{x} - \mathbf{y}\|_2^2$ is a fidelity term and $R(\mathbf{x})$ is regularization term.
For the full derivation of $\arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{y}\|_2^2$ refer to appendix A.

A popular approach in standard inverse problems is to enforce some sparsity property of the solution $\mathbf{x}$. A known model is the Least Absolute Shrinkage and Selection Operator (LASSO) which can enhance the accuracy of prediction and interpretability of the resulting statistical model by performing both variable selection and regularization. So, the equation reads :

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda||\mathbf{x}||_1 \tag{1.3}$$

The parameter λ is a weighting parameter balancing regularisation with data fit.

One of well-known minimization algorithm for solving numerically the minimization problem in Eq. (1.3) is (proximal) Gradient descent [5], well suits here. Though this method is very efficient and simple but convergence can be slow.

1.2 Fluorescence microscopy

In this section we explain the basic principle of, how the observed images are captured through the fluorescence microscope. There are multiple vibrational energy levels for the atoms in the specimen. Normally the atoms are in the ground state but the fluorescent molecule can also be excited to the higher vibrational state when the light illuminates it. If the illumination light has a suitable wavelength then the acquisition of the photons will be most efficient. After a certain period the electrons will fall back into the ground

state. The absorbed photons will have higher energy levels compared to the photons which are emitted. This means that excitation light has a smaller wavelength compared to the emitted light. This is the basic principle of fluorescence microscopy. This process is depicted in the Figure 1.4 and a schema of the emission wavelengths is reported in Figure 1.5.

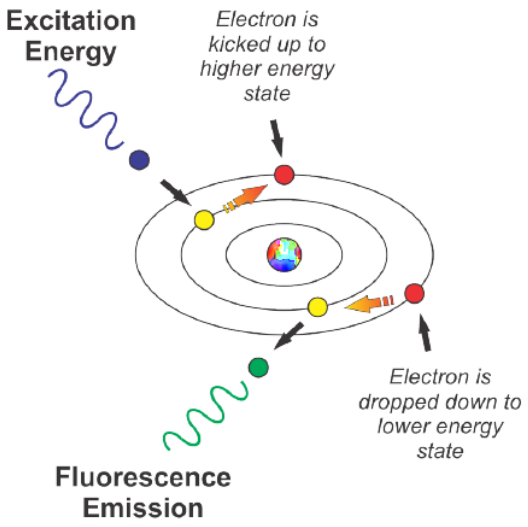

FIGURE 1.4 – The energy levels of an electron when it is excited and when it emits energy. Referred from [6].

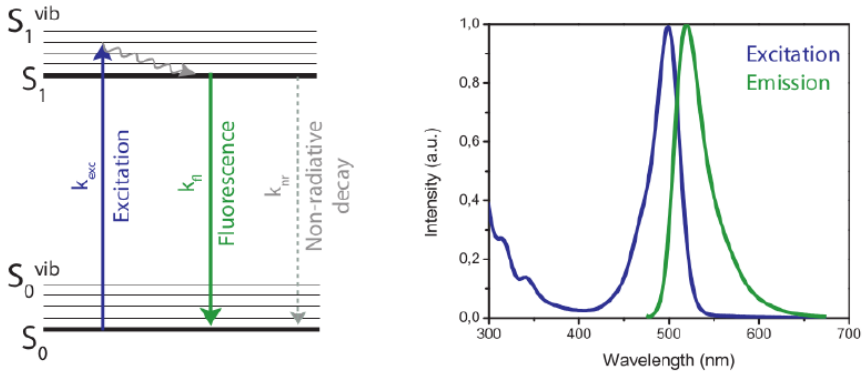

FIGURE 1.5 – Jablonski diagram illustrating the energy state transitions (left). The excitation and emission profile of a commonly used fluorescent dye Alexa 488 (right).Taken from [7].

The working principle of fluorescence microscopy is presented in Figure 1.6. The light

coming from the light source is filtered by the excitation filter. After that, the filtered light falls into the dichroic mirror. This is the mirror used to reflect the fraction of light that has the higher wavelength and let passes through a lower level of wavelength. In this way, light can be projected on the specimen and used the fluorophores. Later, Photons or light is emitted with an emissions wavelength (in green in Figure 1.6). As the emitted light has a higher wavelength, this light will pass through a dichroic mirror. This light later again will be filtered through an emission filter. Finally, the reflected light will be captured by the detector. The captured image will be subject to blur caused by the light diffraction observed in the acquisition process and by electronic and photon noise.

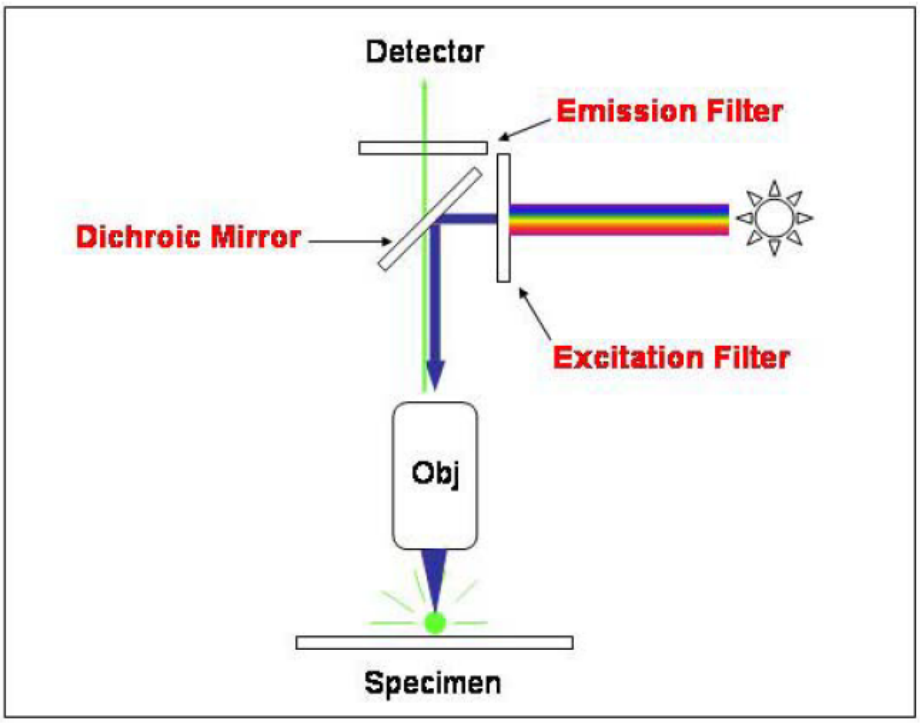

FIGURE 1.6 – The depiction of fluorescence microscopy working procedure[8].

To model mathematically the physical model causing blur in the observations, we fix here the forward model to be the discrete convolution of $\mathbf{x}$ with a convolution kernel $\mathbf{A}$ which, for simplicity, we will assume to be Gaussian. Convolution can be represented by the matrix multiplication with a matrix A. So, A will be referred to the convolution matrix. Figure 1.7 can visualizes the action of the convolution on signal in 1-dimension. Along with the convolution, there will be a added noise in the observed images.

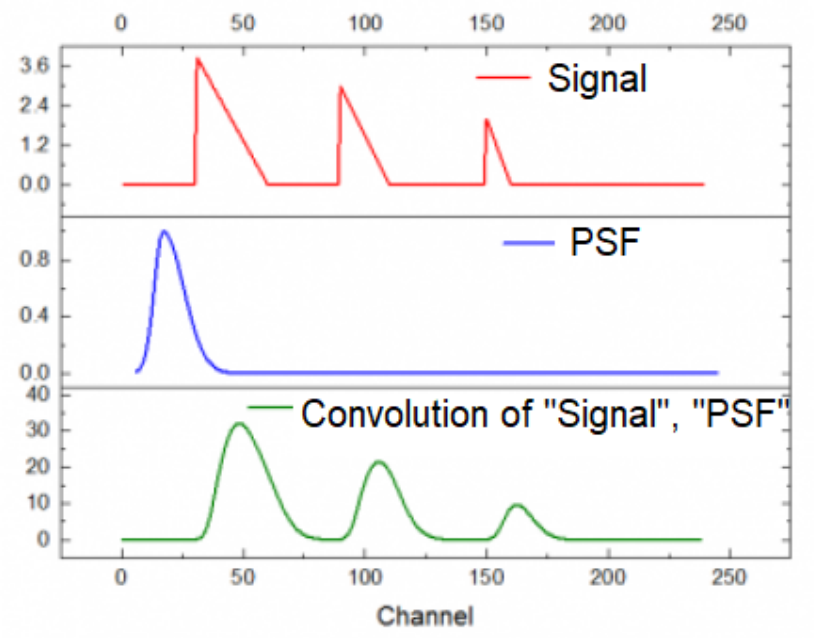

FIGURE 1.7 – The convolution of 1-dimensional signal (in red) with PSF (in blue) results convolved signal (in green)[8].

Physical methods for super resolution :

There are three outperforming approaches, Single Molecule Localization Microscopy (SMLM)[9], STimulation Emission Depletion (STED)[10] and Structured Illumination Microscopy (SIM)[11] . In SMLM only a few emitters are activated in each of thousands acquired images. The great advantage of having only a few elements activated is to localize them easily. For this purpose, usual statistical methods allow to find their precise location. In STED, the PSF will be shrinked with the help of another laser in the physical system that depletes the molecules in the donut-shape area around the focal point. In SIM, the aim is to exploit the interference pattern on the sample to derive more precise information on the location of the fluorescent markers.

Model based methods for super resolution :

There are super-resolution techniques that do not require any particular equipment but a series of images from a confocal, standard wide-field or other microscope. The interest of these approaches, generally more recent, is to find a super-resolved image without expensive devices and specific molecules.

The goal of these method is to find the precise position of the fluorescent molecules from the blurred, noisy and low resolution images available. These techniques aim at solving an ill-posed inverse problem that someone can formulate using the physical modelling of the microscope. They exploit the fact that the emitters are spatially and temporally independent by using second order statistics and reformulating the physical model in the covariance domain. Two of these methods are :

COvariance-based L_0 super-Resolution Microscopy with intensity Estimation (COLORME)[2] : This method is based on the estimation of the temporal auto-covariance matrix of the fine grid pixels over time, in order to precisely localize the fluorescent molecules. It is based on the hypothesis of the non-correlation of the emitters. It allows a good spatial resolution and has the advantage of reconstructing not only the support of the signal but also the intensity and background.

SPARsity-based super-resolution COrrelation Microscopy (SPARCOM)[12] : This method , similarly to COL0RME, assumes the absence of correlation between emitters, and solves an inverse problem in the covariance domain. There is another version of this method LSPARCOM[13] which more than SPARCOM finds automatically some hyper-parameters as well as the kernel describing the PSF of the microscope.

1.3 Generative models

In the machine learning community, generative approaches attracted the attention, due to their non-standard capability of providing an estimate of the distribution of the given data to generate the new samples. The most popular generative approaches are variational autoencoders (VAE) and Generative adversarial networks(GAN). Considering the simple architecture and efficiency, VAE has chosen in this work.

1.3.1 Variational Autoencoders

Variational autoencoders (VAE) are an extended version of autoencoders (AE). Typically the autoencoders consists of two main networks, one is the encoder and another one is the decoder. Autoencoder will propagate the given data into lower dimensional space

(encoding) with the help of encoder and then from lower dimensional space, the data will be reconstructed into the original space (decoding) with the help of decoder.

A variational autoencoder is being an autoencoder where regularisation is added to avoid over fitting which may happen in the attempt of generating the latent space of the given input with all required good properties. VAEs were introduced by Kingma & Welling (2014)[1]. VAEs can produce the distribution of given data $\mathbf{x}$ using a latent variable $\mathbf{z}$ with a fixed (a unit normal distribution) prior $p(\mathbf{z})$ which mathematically represents as follows :

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}. \tag{1.4}$$

The two networks map data from one representation to another, an encoder network $f_\phi(\mathbf{x})$, maps the observed image to a distribution $q_\phi(\mathbf{z}|\mathbf{x})$ in latent space and a decoder network $g_\theta(\mathbf{z})$ maps it to a distribution $p_\theta(\mathbf{x}|\mathbf{z})$ in image space. So, we can consider that ϕ and θ, are the parameters of the encoder network and decoder network, respectively. We can observe that the decoder alone (together with a suitable prior $p(\mathbf{z})$) is good enough to describe the generative model in Eq. 1.4 (the generative model is independent to encoder parameter ϕ). It is usually modelled to factorize over pixels [14].

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{N} p_\theta(x_i|\mathbf{z}) \tag{1.5}$$

where $p_\theta(x_i|\mathbf{z})$ is a normal distribution. The encoder distribution is modelled factorizing over the dimensions of the latent space. In the training of VAE, optimizing the parameters θ and ϕ are optimized to make sure that Eq. 1.5 suits the distribution of given training data $\mathbf{x}$. In Kingma et al[1]. it is shown that this requirement can be achieved with the help of the encoder by jointly optimizing ϕ and θ to minimize the loss :

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \mathcal{L}^R_{\phi,\theta}(\mathbf{x}) + \mathcal{L}^{KL}_{\phi}(\mathbf{x}) \tag{1.6}$$

where $\mathcal{L}^R_{\phi,\theta}(\mathbf{x})$ is reconstruction loss and it defined as follows :

$$\mathcal{L}^R_{\phi,\theta}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-logp_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\sum_{i=1}^{N} -logp_\theta(\mathbf{x_i}|\mathbf{z})] \tag{1.7}$$

and $\mathcal{L}^{KL}_{\phi}(\mathbf{x})$ is the KL divergence loss $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$. Considering sample from the posterior $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ using $\mathbf{z} = \mu + \sigma \bigodot \epsilon$ where $\epsilon \sim N(0, I)$ and $\bigodot$ the element-wise product, the KL divergence is approximated as follows :

$$\mathcal{L}^{KL}_{\phi}(\mathbf{x}) \simeq -\frac{1}{2}\sum_{j=1}^{J}(1 + log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2) \tag{1.8}$$

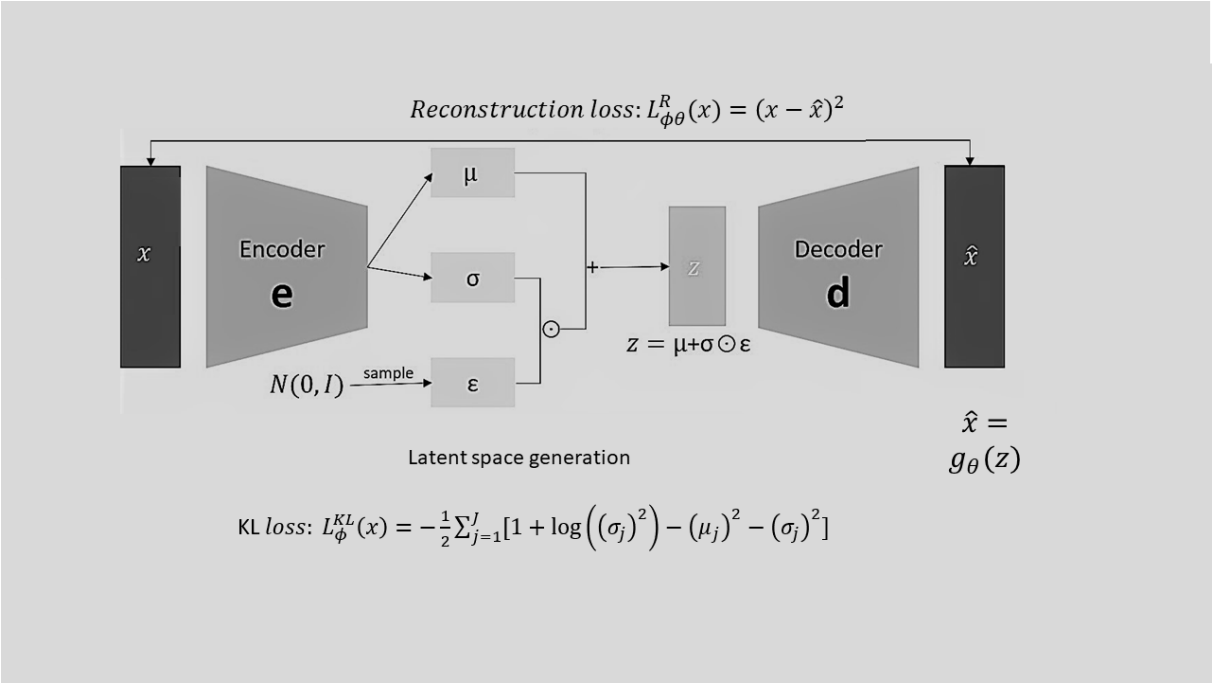

FIGURE 1.8 – The full architecture of a variational autoencoder, where μ and σ are the outputs of encoder part and ϵ is random variable drawn by a normal distribution. $\mathbf{z}$ is the latent space for the given data $\mathbf{x}$ and $\hat{\mathbf{x}}$ is output from the generator g_θ[15].

Where σ and μ are the outcomes of encoder and j represents dimension of the latent vector z. J is the total dimension of latent space (latent variable dimension). Refer the appendix B for the full derivation of $\mathcal{L}^{KL}_\phi(\mathbf{x})$[16].

Where as $\mathcal{L}^{KL}_\phi(\mathbf{x})$ can be computed analytically, the expected value defining $\mathcal{L}^{R}_{\phi,\theta}(\mathbf{x})$ can be approximated by drawing a single or n number of samples from the distribution $q_\phi(\mathbf{z}|\mathbf{x})$. We can take MMSE in the case of taking n number of samples. In this case, it required to follow the reparametrization trick for efficient gradient computations.

Reparametrization trick : In VAE the samples are drawn from the random node z. As we can not back-propagate through a random node, the VAE authors introduced a new parameter ϵ (a random parameter with a certain p.d.f.) which can help to backpropagate through z so that the the randomness moved to ϵ.

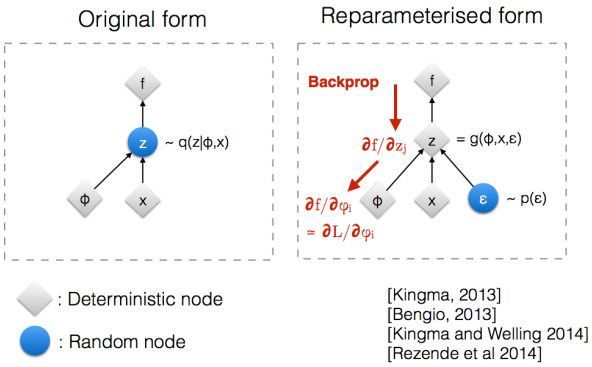

FIGURE 1.9 – Reparametrization trick in variational autoencoders. In order to backpropagate, the randomness is moved from z to ϵ. Taken from [17].

2

Related work

2.1 Deep Learning Based Denoising

In recent years, the usage of deep learning to denoise the images has increased drastically and there are nowadays many methods which learns a direct mapping from noisy image to clean images. These methods are outperformed in comparison to the classical denoising methods which often makes use of sophisticated diffusion and non-local filtering schemes. Some of the most famous methods based on deep learning are the ones of (Zhang et al., 2017a)[18] and (Weigert et al., 2018)[19]. One more interesting contribution by Ulyanov et al. (2018)[20] which uses Deep Image Prior, and consists of an untrained network to-be-trained for each new given image with no need of a training set. However, in this approach, training has to be stopped after a suitable but a priori unknown number of training steps.

2.2 DivNoising : A VAE model for image denoising

An interesting work done by Prakash et al.[14], proposes a denoising approach that is based on Variational Autoencoders (VAEs). The method is called DivNoising and is fully unsupervised by explicitly including the noise model into the decoder. In the DivNoising, the authors assume that images $\mathbf{y}$ have been created from a clean signal $\mathbf{x}$ via a known noise model $\mathbf{NM}$, i.e., $\mathbf{y} \sim p_{NM}(\mathbf{y}|\mathbf{x})$. which was assumed to be Gaussian :

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \tag{2.1}$$

Where $\mathbf{n} \sim N(0, \sigma^2 Id)$ and σ^2 is known.

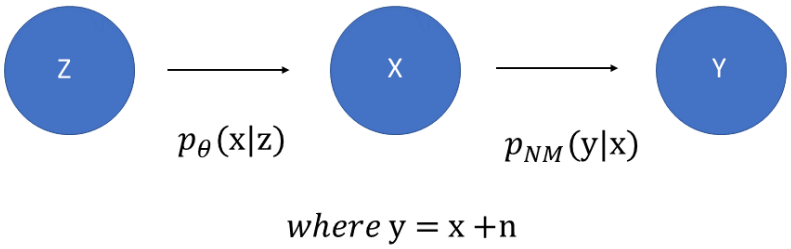

FIGURE 2.1 – DivNoise schema for the decoder, where $p_\theta(\mathbf{x}|\mathbf{z})$ generates new samples with the help of decoder part and $p_{NM}(\mathbf{y}|\mathbf{x})$ is the noise model. The observed image $\mathbf{y}$ is obtained by just adding noise $\mathbf{n}$ to the ground truth $\mathbf{x}$.

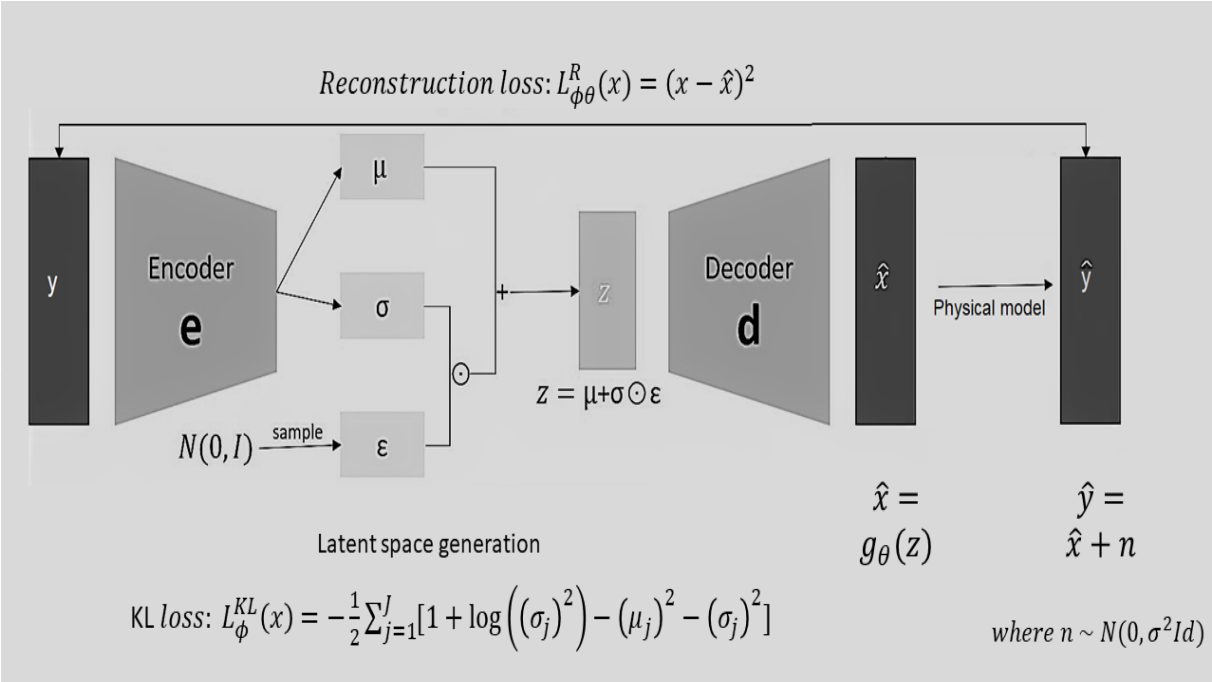

FIGURE 2.2 – The DivNoising method with the extended version of variational architecture for denoising. $\hat{\mathbf{y}} = \hat{\mathbf{x}} + \mathbf{n}$

To integrate this within the VAE architecture, the authors replace the generic normal distribution over pixel intensities in Eq. 2.1 with a known noise model $p_{NM}(\mathbf{y}|\mathbf{x})$. They get $p_\theta(\mathbf{y}|\mathbf{z}) = p_{NM}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} p_{NM}(y_i|x_i)$, with the decoder network now predicting the signal $g_\theta(\mathbf{z}) = \mathbf{x}$. Together with $p(\mathbf{z})$ and the noise model, the decoder now describes a

full joint model for all three variables[14] :

$$p_\theta(\mathbf{z}, \mathbf{y}, \mathbf{x}) = p(\mathbf{y}, \mathbf{x}|\mathbf{z})p(\mathbf{z}) = p(\mathbf{y}|\mathbf{x}, \mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) = p_{NM}(\mathbf{y}|\mathbf{x})p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{2.2}$$

where $p(\mathbf{y}|\mathbf{x}, \mathbf{z}) = p_{NM}(\mathbf{y}|\mathbf{x})$. For a given $\mathbf{z}$, as for standard VAEs, the decoder describes a distribution $p(\mathbf{y}|\mathbf{z})$ over noisy images. The corresponding clean signal $\mathbf{x}$, in contrast, is deterministically defined. Hence, $p_\theta(\mathbf{x}|\mathbf{z})$ is a Dirac distribution centered at $g_\theta(\mathbf{z})$[14].

Therefore, the reconstruction loss after also factorizing over pixels, becomes :

$$\mathcal{L}^R_{\phi,\theta}(\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\sum_{i=1}^{N} -log p_{NM}(y_i|\mathbf{x} = g_\theta(\mathbf{z}))].$$

Apart from this modification, the DivNoising approach follows the standard VAE training procedure. The only modification consists in how the decoder distribution is modeled. The authors assume that the training procedure still produces a model describing the distribution of the training data, while making sure that the encoder distribution well approximates the distribution of the latent variable given the image.

3

Proposed work

3.1 DivBlurring : A VAE model for image deblurring

In order to further add the modelling of blur to the one of the noise in the DivNoising approach, we introduced convolutional operator appearing in the fidelity term to the final step of the decoder architecture. So, the schema of the decoder will be the following :

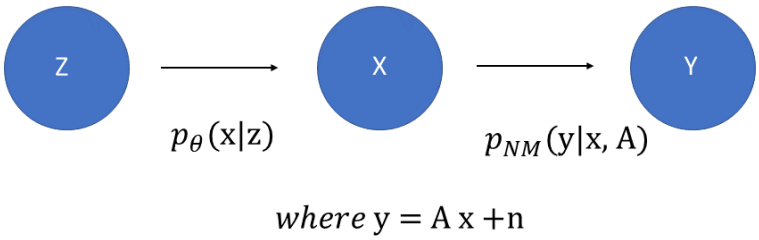

FIGURE 3.1 – DivBlurring schema for the decoder, where $p_\theta(\mathbf{x}|\mathbf{z})$ generates the new samples with help of decoder network and $p_{NM}(\mathbf{y}|\mathbf{x}, \mathbf{A})$ is the image model. The observed image y is obtained by blurring using the blurring kernel **A** and adding noise **n** to the ground truth image **x**.

We recall the image formation model :

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{3.1}$$

where **y** is the observed noisy and blurred data, **A** is the matrix associated to the blur kernel, **n** is Gaussian noise and **x** is the "clean" image we want to find. We can also

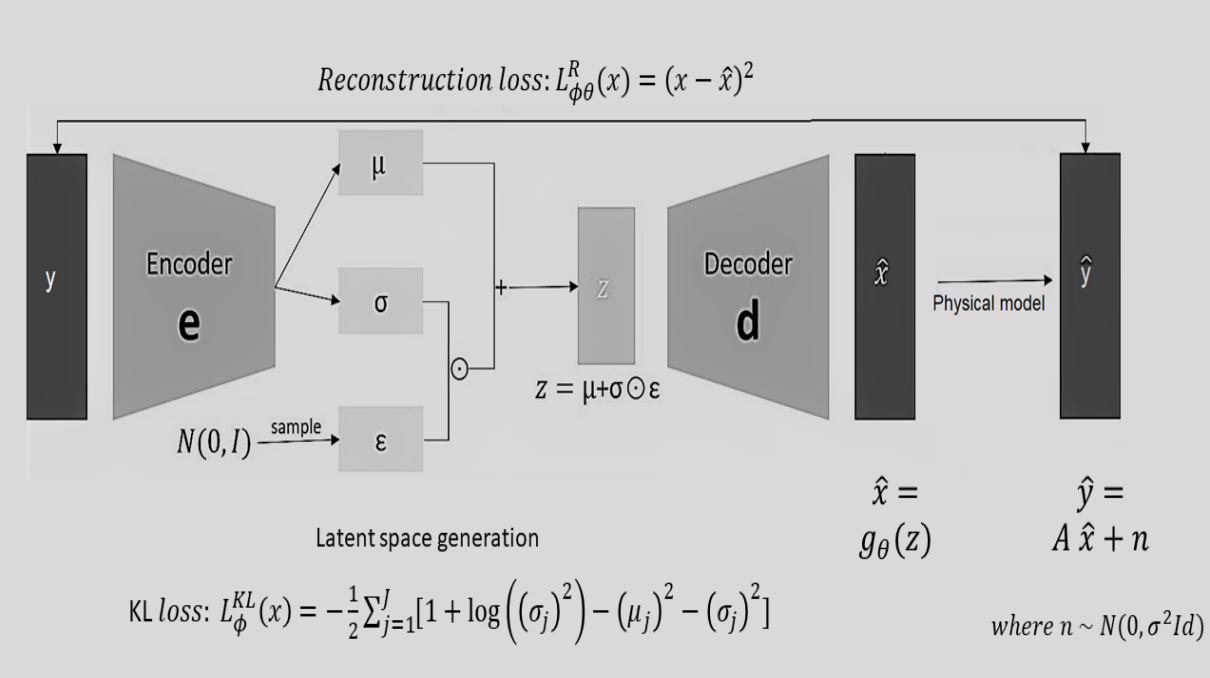

FIGURE 3.2 – The DivBlurring method with the extended version of variational architecture for denoising and deblurring. With the new notation $\hat{\mathbf{y}}$ is the output of physical model. **Note :** $\hat{\mathbf{y}} = A\hat{\mathbf{x}} + \mathbf{n}$

write : $\mathbf{y} \sim p_{NM}(\mathbf{y}|\mathbf{x}, \mathbf{A})$. So in order to incorporate the knowledge that we have from the acquisition model, we will assume for the decoder :

$$p_\theta(\mathbf{z}, \mathbf{y}, \mathbf{x}) = p_{NM}(\mathbf{y}|\mathbf{x}, \mathbf{A}) p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \tag{3.2}$$

and the loss function :

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \mathcal{L}^R_{\phi,\theta}(\mathbf{y}) + \mathcal{L}^{KL}_\phi(\mathbf{y})$$

where

$$\mathcal{L}^R_{\phi,\theta}(\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\sum_{i=1}^{N} -log p_{NM}(y_i|\mathbf{x} = g_\theta(\mathbf{z}), \mathbf{A})]$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[-log p_{NM}(\mathbf{y}|\mathbf{x} = g_\theta(\mathbf{z}), \mathbf{A})]$$

which, recalling the image observation model can be expressed therefore

$$p_{NM}(\mathbf{y}|\mathbf{x} = g_\theta(\mathbf{z}), \mathbf{A}) = \frac{1}{Z} e^{-\frac{||\mathbf{A}\mathbf{g}_\theta(\mathbf{z}) - \mathbf{y}||_2^2}{2\sigma^2}}$$

where Z is a normalisation constant. Hence we find the expression :

$$\mathcal{L}^{R}_{\phi,\theta}(\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[-log[\frac{1}{Z}e^{-\frac{||\mathbf{A}g_\theta(\mathbf{z})-\mathbf{y}||_2^2}{2\sigma^2}}]]$$

$$= const + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}\frac{||\mathbf{A}g_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2}$$

and as we will try to minimize the reconstruction loss, we can get rid of the constant. So, finally the reconstruction loss is given by :

$$\mathcal{L}^{R}_{\phi,\theta}(\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}\frac{||Ag_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2}$$

On the other side, the KL loss will be the same as in the general VAEs :

$$\mathcal{L}^{KL}_{\phi}(\mathbf{y}) = D_{kl}(q_\theta(\mathbf{z}|x_i)||p(\mathbf{z}))$$

$$= -\frac{1}{2}\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)$$

where j represents dimension of the latent vector $\mathbf{z}$, with J being the total dimension of latent space(latent variable dimension).

Therefore the total loss function is :

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \frac{||Ag_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2} + [-\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)] \tag{3.3}$$

Exceptional Case : Posterior collapse in the prior [21]

In some cases, based on the choice of the hyperparameters (learning rate, batch size, etc) and data complexity, the impact of the given input will have no impact on the produced latent space. In such cases the KL divergence loss will be very close to 0. This situation is refereed to as posterior collapse into the prior. In this cases, the model ignore to learn the latent variables. There are multiple approaches, to overcome the posterior collapse, namely Beta-vae[21] and δ-vae[22].

3.2 Best match regularizers (penalties)

In order to avoid the under-fitting and over-fitting, the regularization term helps to penalize the loss function. As there are many possible ways to penalize the loss function

we choose some specific and suitable regularizers which, moreover, given the biological context we are working in, enforce physically meaningful properties to the solution.

3.2.1 DivBlurring with ℓ_1-norm regularization penalty

The first penalty we chose is the ℓ_1-norm, which is given by the sum of the absolute values of the magnitude of the signal and promote sparsity to the signal. Therefore, the loss function will have the following form :

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \frac{||\mathbf{A}g_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2} + [-\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)] + \lambda||g_\theta(\mathbf{z})||_1^1$$

where $\lambda > 0$ is a regularization hyper-parameter that controls the importance of the regularization term.

3.2.2 DivBlurring with ℓ_2-norm regularization penalty

The second regularization term that we used is the ℓ_2-norm which is also called "Ridge regression" and is defined as sum of the squared magnitudes along with coefficient to the loss function is given by :

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \frac{||\mathbf{A}g_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2} + [-\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)] + \lambda||g_\theta(\mathbf{z})||_2^2$$

where $\lambda > 0$ a regularization hyper-parameter..

3.2.3 DivBlurring with Positivity constrain

As we are looking for biological samples associated to photon counts we introduce a regularization which enforces the solution to be non-negative. The regularization term we are going to use promotes all signal values to be positive by penalizing the negative ones. The loos function will be : .

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \frac{||Ag_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2} + [-\sum_{j=1}^{J}(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)] + \lambda \sum [Q(g_\theta(\mathbf{z})_i)^2]$$

where $Q(g_\theta(\mathbf{z})_i)$ is 0 when $g_\theta(\mathbf{z})_i >= 0$ otherwise it is equal to $g_\theta(\mathbf{z})_i$ and λ is regularization parameter.

3.2.4 DivBlurring with Positivity constrain + ℓ_1-norm regularization penalty

We also combined different regularisation models to improve the results even further. On the observation of pixel ranges, we observe that there are negative values present in the predicted output. So, we mainly taken into consideration a positivity constrain along with that added ℓ_1 norms as follows :

$$\mathcal{L}_{\phi,\theta}(\mathbf{y}) = \frac{||Ag_\theta(\mathbf{z}) - \mathbf{y}||^2}{2\sigma^2} + [-\sum_{j=1}^{J}(1+log(\sigma_j^2)-\mu_j^2-\sigma_j^2)] + \kappa \sum Q(g_\theta(\mathbf{z})_i)^2 + \lambda \sum ||g_\theta(\mathbf{z})_i||_1^1$$

where $Q(g_\theta(\mathbf{z})_i)$ is 0 when $g_\theta(\mathbf{z})_i >= 0$ otherwise it is equal to $g_\theta(\mathbf{z})_i$, $||g_\theta(\mathbf{z})_i||_1^1 = \sum_{i=1}^{n} |g_\theta(\mathbf{z})_i|$, n the number of pixels and κ , λ are regularization parameter.

4

Experimental results

4.1 Architecture

Convolutional architectures are very efficient to create such deep learning models. We chose simple 2 depth VAE architecture for each network (encoder and decoder) design. Depth 2 describes that 2 down sampling layers for the encoder network and 2 up sampling layers for the decoder network. In both networks we designed 3 X 3 convolutions with padding 1 followed by activation function. ReLU is our choice for activation function. Later we max pooled with 2 X 2 maxpooling layer. Finally, in total 36κ parameters got assigned for the both networks by considering input data shape (1 X 256 X 256). After encoding, the dimension of resultant bottleneck for the latent space is chosen 64. The designed network is trained with 32 batch size, 0.001 as the initial learning rate and for the iteration of 150 epochs. The full architecture is presented in the Figure 4.1. To estimate the signal $\hat{\mathbf{x}}$, we averaged of 100 sample's estimates for the better results which is similar to DivNoising[14].

For the code implementation, the popular deep learning library pytorch-lightning has been used. To run the experiments, utilised the in-house server "CALCUL" with high performing configuration with 320GB with two 2 FULL GPU's. Practically speaking, the training and testing of each network is taking around 3h using this high configuration we have access to.

For training the model we used the synthetic data of filaments with noise and blur. The data were generated by using the MatLab, version-R2022a. The main advantage of using the synthetic data is that we can compare the results with the ground truth so that

a quantitative assessment can be computed. We experimented on two types of data sets, DataSet-1 (low noise and high blur) which is more near to realistic data and DataSet-2 (high noise and high blur) which is a exceptional case to estimate the potential of DivBlurring. In the first scenario the PSF used to blur the data have a full width at half maximum (FWHM) equal to 3.8 and in the other one equal to 3.0, while in both cases the size of the pixels is equal to 25nm. Each data set has 7000 images with different noise realisations.

Initially we report the original ground truth and noisy and blur data of tubulin synthetic data. Note that we did not use the ground truth in training the model as the method is totally unsupervised but only to calculate the Peak signal-to-noise ratio (PSNR) value for the better comparison among the different penalties.We calculated the PSNR value which is average of 100 PSNR values of sample predictions for each model. The high value of PSNR represents the good estimation and where as low value represents not so much deblurred and denoised or with other artifacts. The formula we used to compute the PSNR is the following :

$$PSNR = 20 \cdot log_{10}(\frac{Max_{GT}}{\sqrt{MSE(\mathbf{y}, \hat{\mathbf{y}})}})$$

Where Max_{Gt} is max pixel of the ground truth and $MSE(\mathbf{y}, \hat{\mathbf{y}})$ is mean sqare error of observed image $\mathbf{y}$ and predicted image $\hat{\mathbf{y}}$.

Choosing regularisation parameters : One of the challenging task we faced to reach a better suitable value of the regularisation parameter for each penalty. In each case we experimented multiple possible values after considering reconstruction loss $\mathcal{L}^R_{\phi,\theta}$, KL divergence loss $\mathcal{L}^{KL}_{\phi}$ and $R(\mathbf{x})$ ranges. We present some representative results using a selection of values for each regularisation parameter after experimenting with many of them.

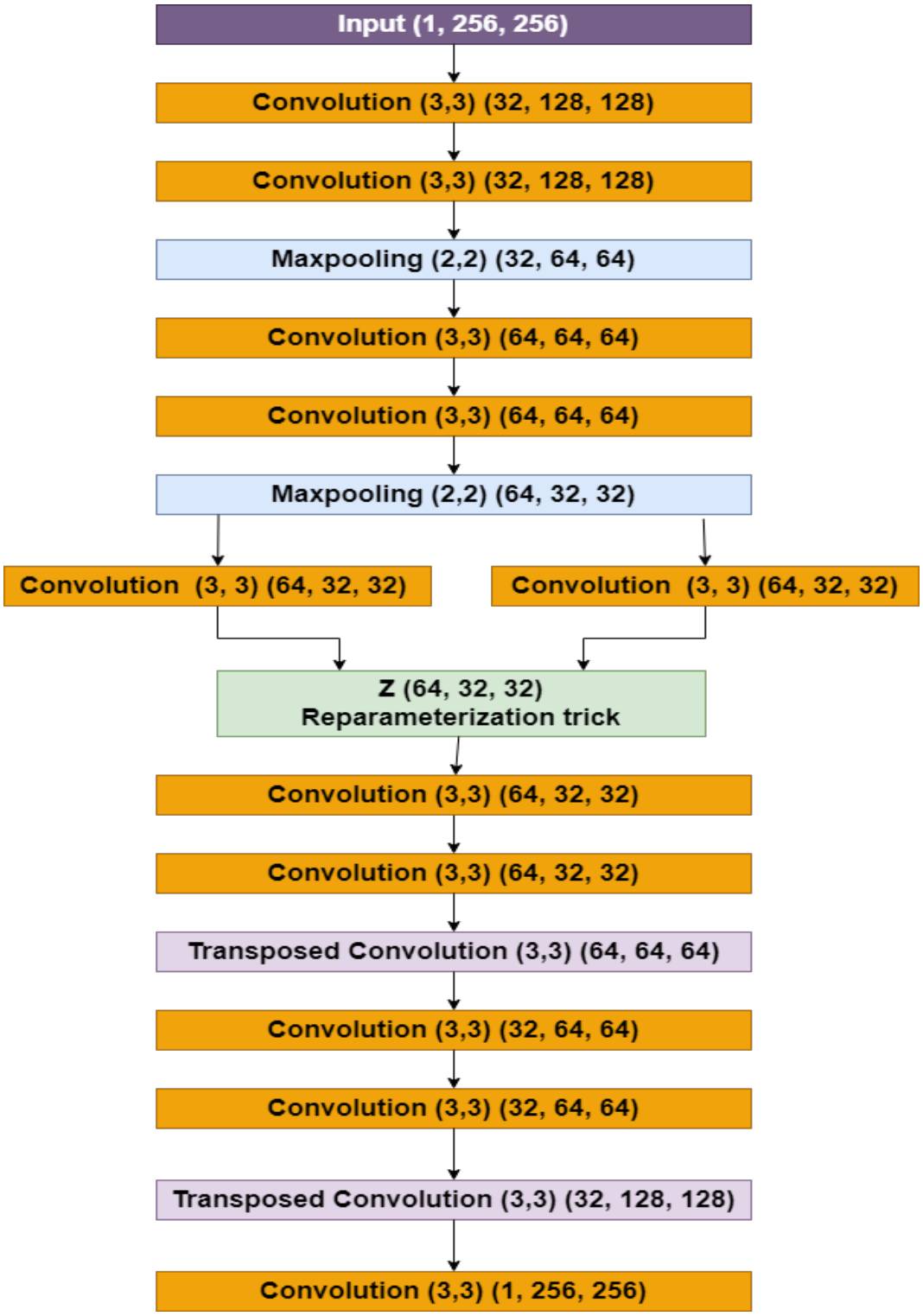

FIGURE 4.1 – The full architecture of the variational autoencoder for denoising and deblurring (following [14]) with 2 down sampling networks in encoder part with 2 convolution layers each and 2 up sampling networks in decoder part with 2 convolution layers each.

4.1.1 Existing work : DivNoising

The following results are estimated using the existing method DivNoising by passing the noisy and blurry data which is doing only denoising. In this method, there are two losses, reconstruction loss $\mathcal{L}^{R}_{\phi,\theta}$ and KL divergence loss $\mathcal{L}^{KL}_{\phi}$. Refer to section 2.2.

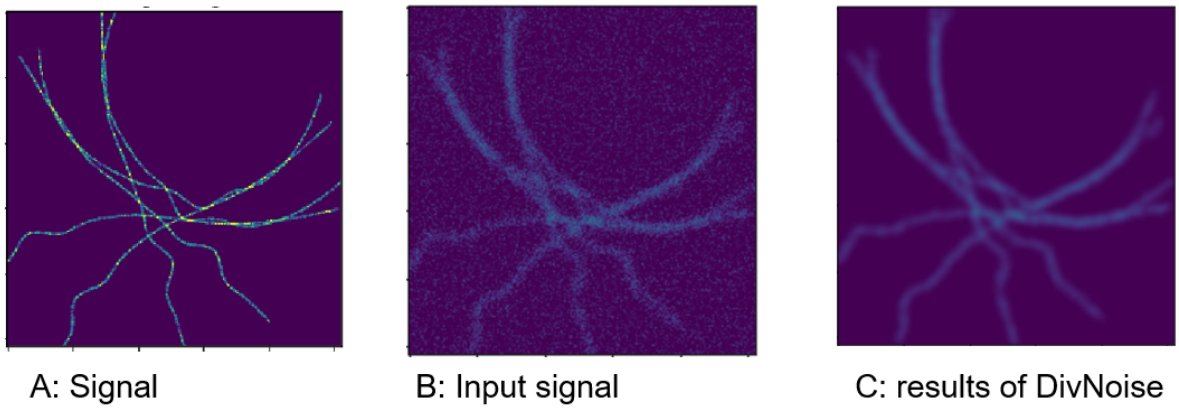

FIGURE 4.2 – True signal and noisy input along with predicted image by DivNoising.

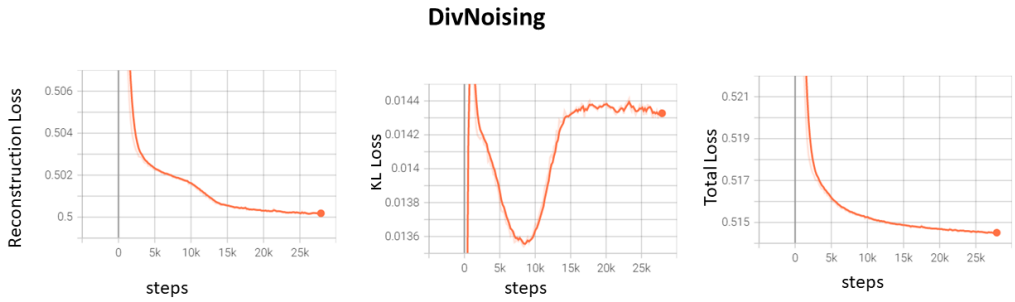

FIGURE 4.3 – The converging loss values in the case of DivNoising. For the steps calculation refer Eq 4.1.

This the method can effectively denoise the data but not good enough in deblurring. We can see that the reconstruction loss is decreasing, the KL loss is increasing and the total loss is in downwards direction.

4.2 Results of DataSet-1 (low noise and blur)

In the following sections, we present the results of existing method "DivNoising" and our method "DivBlurring" along with different regularisers.

In the model training process, analysing the convergence of loss values is essential. Observing the loss values evaluation over the epochs will not sufficient enough for the deep analysis because of the total number epochs are limited to 150 in our case (still we presented for DataSet-2 results over epochs). So, we are presenting the loss values evaluation over the steps for each category (Reconstruction loss, KL loss and regularisers loss) individually along with total loss.

The total number of steps can be calculated for the whole training process as follows :

$$steps = \frac{DataSetSize(7\kappa) \cdot epochs(150)}{batchsize(32)} \approx 32\kappa \tag{4.1}$$

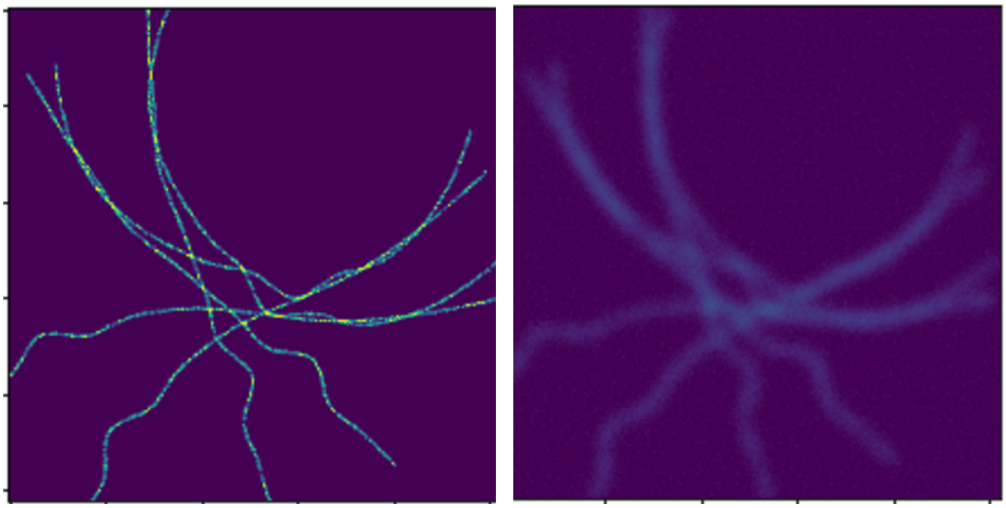

FIGURE 4.4 – DataSet-1 (high blur and low noise) with the level of blur and noise matches with real scenario. LEFT : true signal, RIGHT : noisy and blur version.

4.2.1 Proposed work : DivBlurring

This is the original form of DivBlurring method without any added penalties. In this method the total loss is the addition of only reconstruction loss $\mathcal{L}^R_{\phi,\theta}$ and KL divergence

loss $\mathcal{L}_{\phi}^{KL}$. Refer section 3.1 and Eq 3.3.

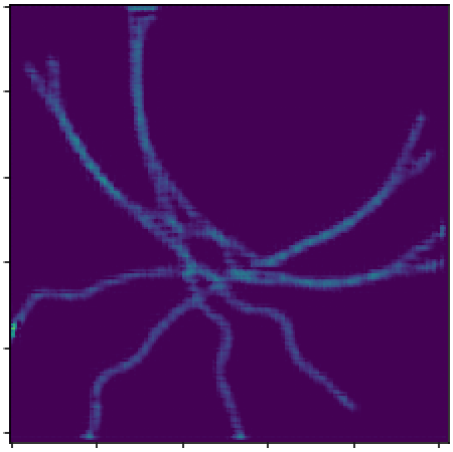

FIGURE 4.5 – The predicted image by DivBlurring. There is no any penalization.

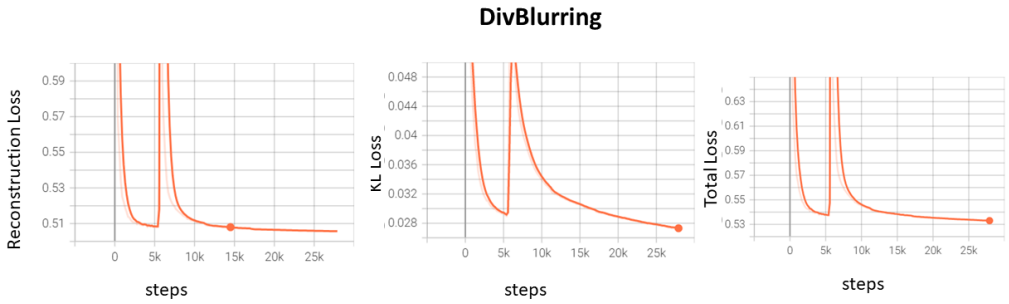

FIGURE 4.6 – The converging loss values in the case of DivBlurring. For the steps calculation refer Eq 4.1.

We can see that there is an improvement with respect to the blurry and noisy image given as an input to the network but still there are some filaments which are not distinguishable. Even though there is a increase in losses at starting but later they decreased.

4.2.2 DivBlurring with ℓ_1-norm regularization penalty

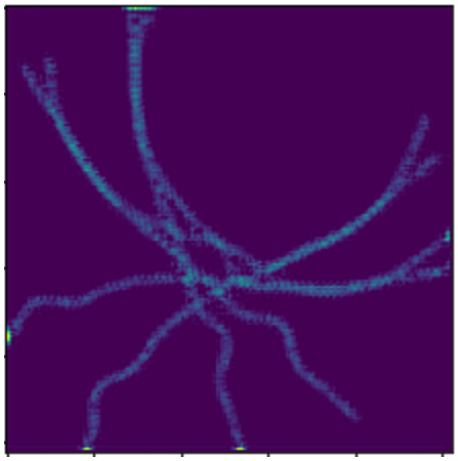

FIGURE 4.7 – The output of DivBlurring with ℓ_1 norm (λ =1e-10).

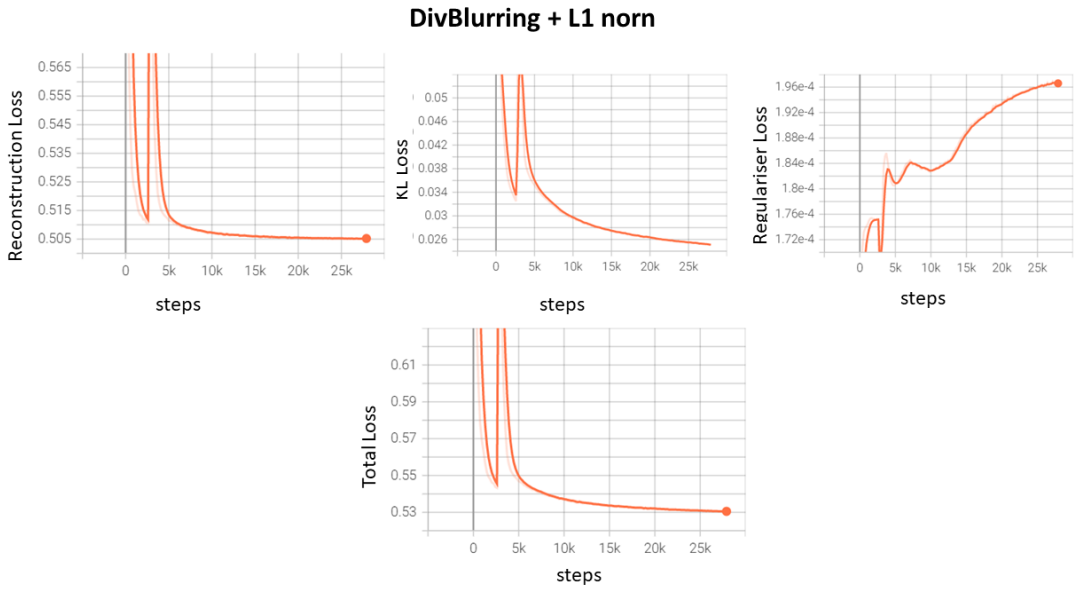

FIGURE 4.8 – Converging loss values of DivBlurring with ℓ_1 norm. For the steps calculation refer Eq 4.1.

We see that, this approach resulting the output which is visually similar to the DivNoising but we can observe the slight improvement in the PSNR values which is mention in the PSNR value comparison table in below sections. The regulariser loss is inconsistent till 10κ steps but afterwards that is in increasing direction. For the loss function, refer to section 3.2.1.

4.2.3 DivBlurring with ℓ_2-norm regularization penalty

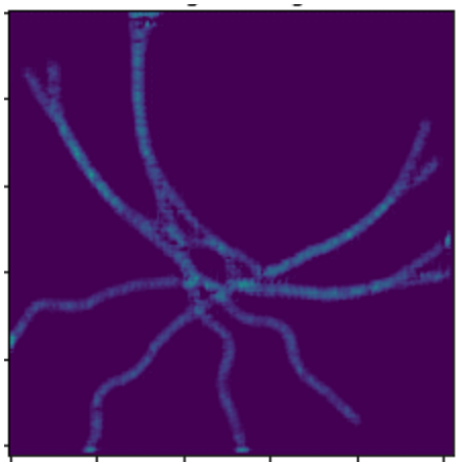

FIGURE 4.9 – The output of DivBlurring with ℓ_2 norm (λ =1e-10).

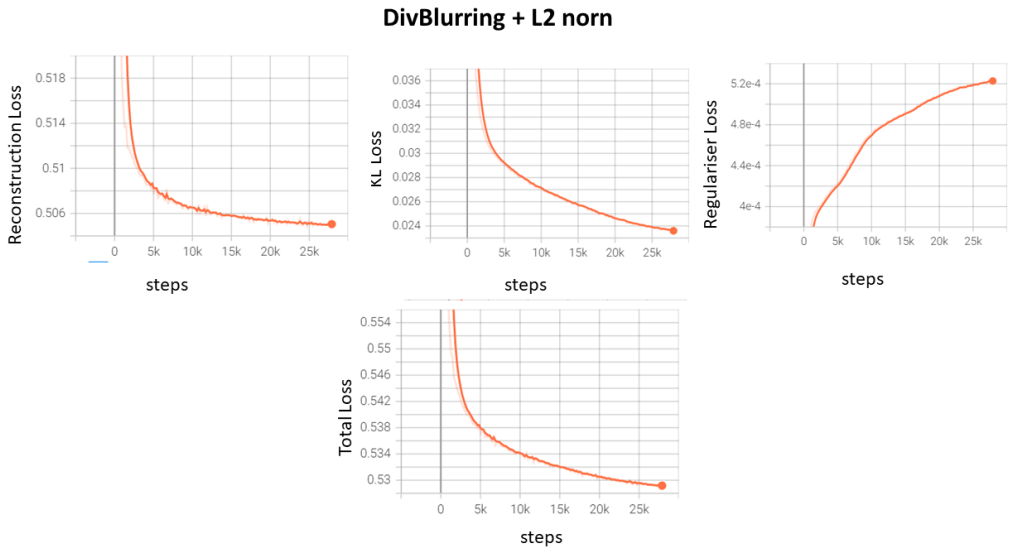

FIGURE 4.10 – Converging loss values of DivBlurring with ℓ_2 norm. For the steps calculation refer Eq 4.1.

Even Divblurring with ℓ_2-norm resulting similar result in visual prospect but notable increase in PSNR value with compared to all other method's PSNR values. In the case of losses, all losses are consist in direction though the regularisation loss is increasing. The main consideration is the total loss is decreasing overall. For the loss function, refer the section 3.2.2.

4.2.4 DivBlurring with Positivity constrain with high λ

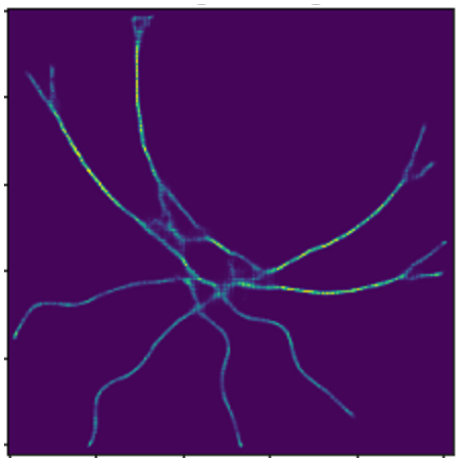

FIGURE 4.11 – The output of DivBlurring with positivity constrain (λ =1e-3).

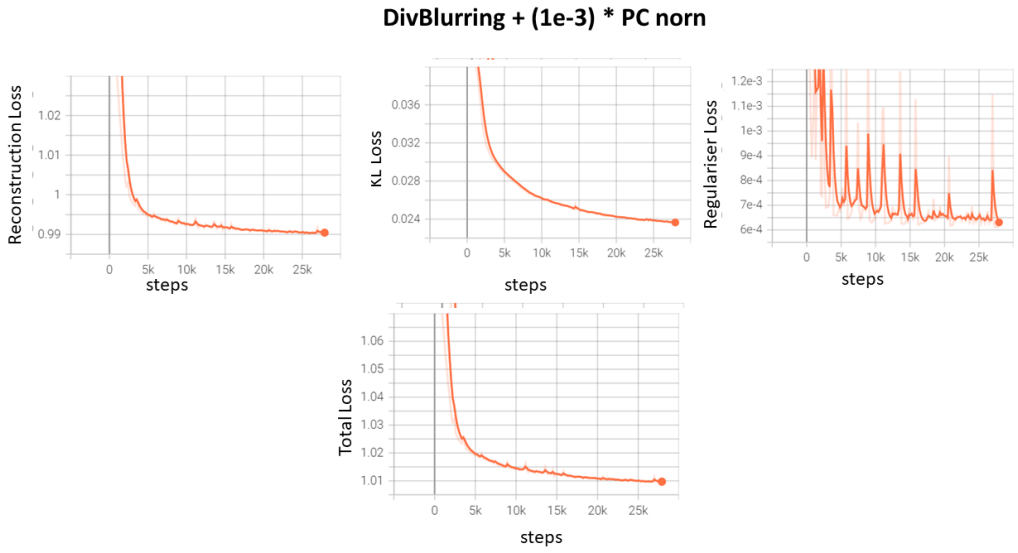

FIGURE 4.12 – Converging loss values of DivBlurring with positivity constrain with coefficient 1e-3). For the steps calculation refer Eq 4.1.

This model producing the better results over all other models in visual prospect. The produced results are visually good enough to segregate the tube-lines which are very thin and we can observe the cross sections of tube-lines. On observation of losses, we can see that all losses are decreasing. For the loss function, refer the section 3.2.3.

4.2.5 DivBlurring with Positivity constrain with low λ

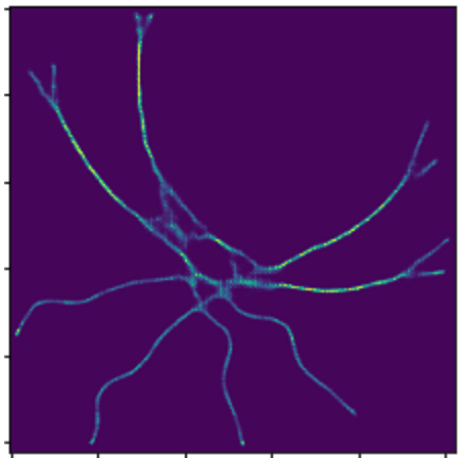

FIGURE 4.13 – The output of DivBlurring with positivity constrain (λ =1e-5).

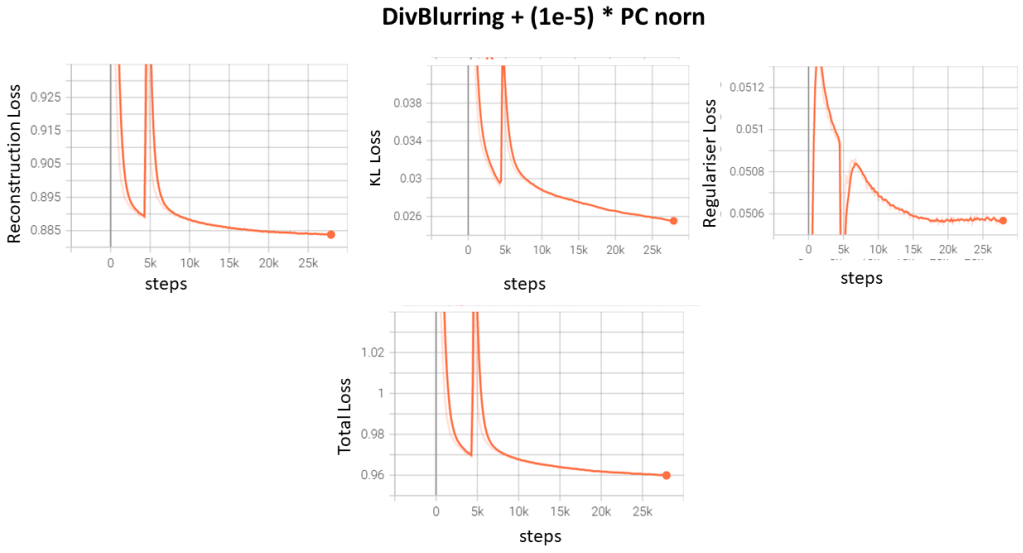

FIGURE 4.14 – Converging loss values of DivBlurring with positivity constrain with coefficient 1e-5. For the steps calculation refer Eq 4.1.

On a small decreasing in the regulariser parameter is resulting in the change in the PSNR value but visually difficult to identify the change. The main observation here is the loss values are not in consistent as compared to high regulariser parameter case. For the loss function, refer the section 3.2.3.

4.2.6 DivBlurring with Positivity constrain and ℓ_1-norm regularization penalty

This is the case where we combined two better working regularisers with better regularisers parameter. So, we combined the case of ℓ_1-norm with paramer value 1e-10 and Positivity constrain with paramer value 1e-3. For the loss function, refer the section 3.2.4.

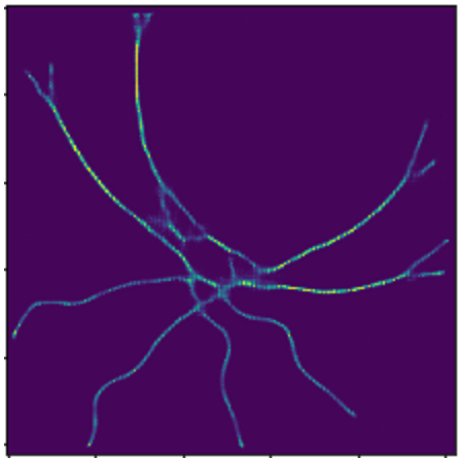

FIGURE 4.15 – The output of DivBlurring with positivity constrain and ℓ_1 norm with regularization parameter values 1e-3 and 1e-10 respectively.

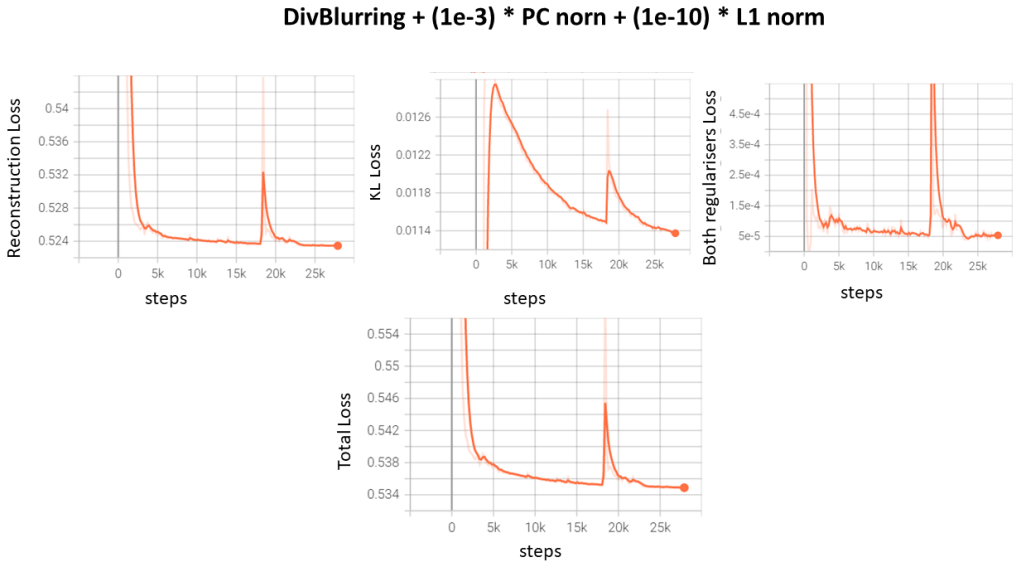

FIGURE 4.16 – Converging loss values of DivBlurring with positivity constrain and ℓ_1 norm with coefficients 1e-3 and 1e-10 respectively.

Though, we combined the two penalties, the resulting results are similar to Positivity constrain case. The loss values are not consistent in the direction but over all they are decreasing.

4.3 Results of DataSet-2 (high noise and blur)

Along with realistic scenario, we experimented with a high noisy scenario. Even in this case our approach are giving competitive results almost similar to low noise data set.

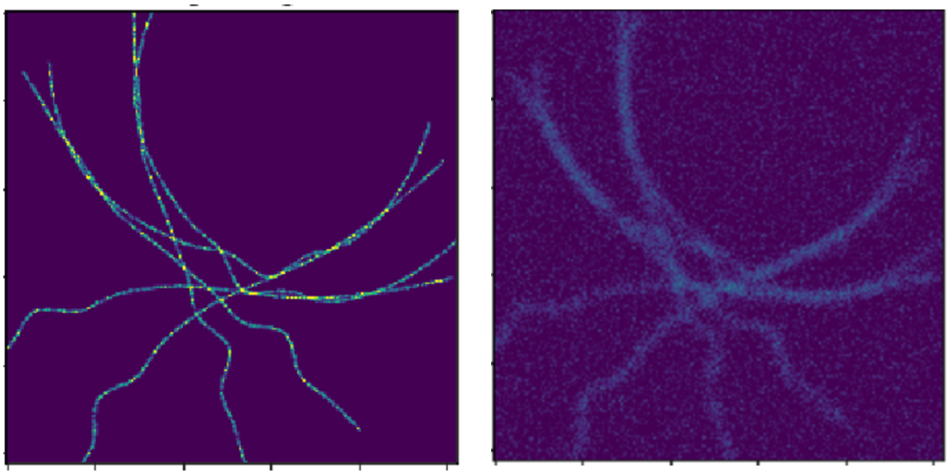

FIGURE 4.17 – Non realistic dataset with high blur and high noise. Left : true signal, Right : noisy and blur version.

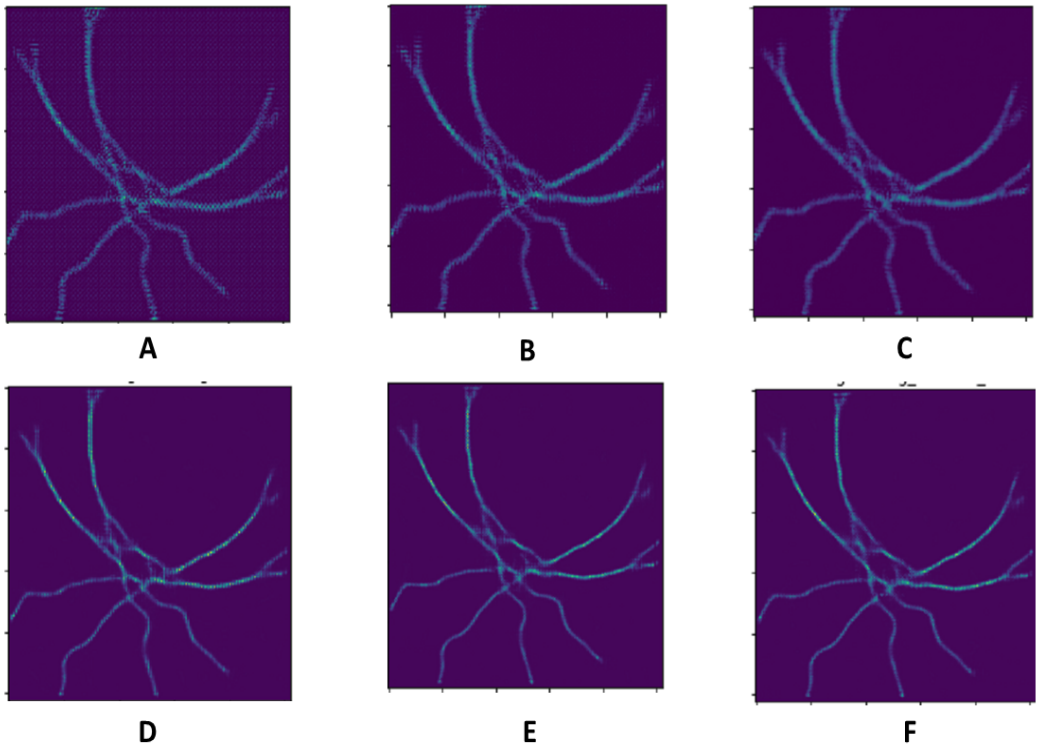

FIGURE 4.18 – Outcomes of DataSet-2 (high noise and high blur). A :DivBlurring B :DivBlurring with ℓ_1 norm C :DivBlurring with ℓ_2 norm D :DivBlurring with PC (1e-3) E :DivBlurring with PC (1e-5) F : DivBlurring with PC (1e-3) and ℓ_1 norm (1e-10).

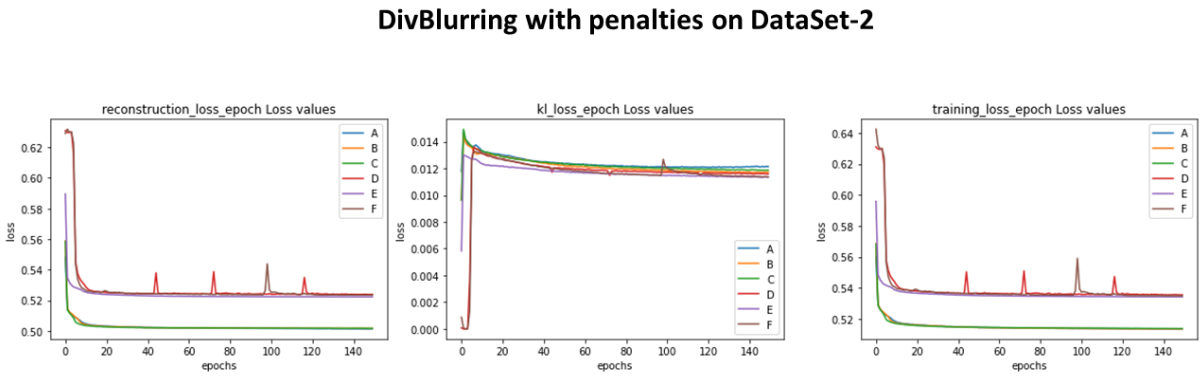

FIGURE 4.19 – The losses of DataSet-2 (high noise and high blur) over epochs. A :DivBlurring B :DivBlurring with ℓ_1 norm C :DivBlurring with ℓ_2 norm D :DivBlurring with PC (1e-3) E :DivBlurring with PC (1e-5) F : DivBlurring with PC (1e-3) and ℓ_1 norm (1e-10).

4.4 Results summary

We can compare the results with PSNR values by consolidating the all experiments along with penalties and respective regulariser parameter values. The following table describes the details :

S.No	Approach Name	λ	κ	PSNR (Low_n)	PSNR(High_n)
1	DivBlurring (DB)	-	-	27.6001	25.8511
2	DB + λ ℓ_1 norm	1e-10	-	27.2099	27.5071
3	DB + λ $\ell_2$norm	1e-10	-	27.6397	27.7475
4	DB + λ PC	1e-3	-	27.3126	27.2231
5	DB + λ PC	1e-5	-	27.4549	27.5092
6	DB +λ PC + κ ℓ_1 norm	1e-3	1e-10	26.9200	27.5534

On keen observation of results, In the prospect of PSNR values (average of 100 sample predictions), even though the results are visually good in "DivBlurring with the positivity constrain and a regularisation parameter equal to 1e-3" but the PSNR value (27.2231) is low with compared to "DivBluring with the ℓ_2-norm penalty and a regularisation parameter equal to 1e-10" PSNR value (27.7475). With deep analysis, the range of the background pixel values in the predicted results in the case of "DivBlurring with ℓ_2 norm" has mostly close to 0 (zero) which is indirectly lowering the denominator of PSNR calculation and resulting the higher PSNR value. But in case of "DivBlurring with positivity constrain" instead of being close to 0 they are further away in positive direction so the PSNR value is a little lower.

The outcomes for the given data from the models are mainly based on the data complexity and model parameters. So, we strongly recommend to chose the penalty term and regulariser parameters based on the data complexity, in our case we consider that "DivBlurring + λ positivity constrain" with λ = 1e-3 has producing the better result.

Conclusion

In every domain, estimating a full informative version of data is a potential requirement. In medical domain, analysing granular level of information is more essential along with accuracy. As we currently focus on fluorescence microscopic images, the inverse problems can help to estimate better version of images with more information. The inverse problems maps the ground truth image to observed image with forward operator, this forward operator will be a convolutional operator for the fluorescence microscopic images.

One of the generative deep learning models, variational autoencoders can estimate the training data distribution with its internal architecture. This distribution can helps to sample the required number of samples to estimate the true images. Adding a noise model in the VAE architecture is the key essential part in our research.

Though, the existing approaches are able to denoise the images but not good enough in the case of deblurring. To solve this problem, our approach "DivBlurring" can outperform in combined work of deblurring as well as denoising with added flavors of penalties. We strongly suggest the penalties can be added based on the data complexity. As in our case we process biological data, the positivity constrain penalty helped us to produce the better results.

Future work

Though we presented some of the possible cases, we could see the large space to dive with more diversified experiments. Some of the suggestions for future work we have are :

— Depending on framework level optimization may not be suitable in all cases. So, rather than depending on "AutoGrad" which calculates the gradients of function in PyTorch, calculating manually promises the better convergence.
— This approach is limited to fluorescence microscopic images. So, this can be formulated to, in such a way to generalise to be applicable on other image capturing methods.
— Along with mentioned regularizers, there is a possibility to experiment such as total variation, huber's regularizer, etc.
— Can be experimented with different architectures of variational autoencoder structures, namely VAE GAN, s3VAE and CVAE.
— Not only finding mean square error of the number of samples taken from the data encoder distribution, but also other efficient methods such as Mean absolute error, R-Squared and Root Mean Squared Error, or Coefficient of determination metrics.
— Super-resolution (reconstruction in a finer grid than the acquisitions) can be performed to promote the outcomes to next level.

Bibliographie

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[2] Vasiliki Stergiopoulou, José Henrique de Morais Goulart, Sébastien Schaub, Luca Calatroni, and Laure Blanc-Féraud. Col0rme : Covariance-based l0 super-resolution microscopy with intensity estimation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 349–352, 2021.

[3] Wikipedia contributors. Point spread function — Wikipedia, the free encyclopedia, 2022. [Online ; accessed 13-August-2022].

[4] Neelamani, Ramesh. Inverse problems in image processing. 2004. 2004.

[5] J. Kiefer and J. Wolfowitz. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3) :462 – 466, 1952.

[6] Lynn Miner. The Science of Fluorescence. 2013-2022. .

[7] Remko Dijksta. Design and realization of a CW-STEDsuper-resolution microscope setup. Oct. 2012. .

[8] Droste, I.E.A.C,. Deconvolution of 3D Fluorescence Microscopy Images using Scaled Gradient Methods. 2021. .

[9] D. Sage, H. Kirshner, T. Pengo, N. Stuurman, J. Min, S. Manley, and M. Unser. Quantitative evaluation of software packages for single-molecule localization microscopy. *Nature methods, 12*, 06 2015.

[10] S. W. Hell and J. Wichmann. Breaking the diffraction resolution limit by stimulated emission : stimulated-emission-depletion fluorescence microscopy. *Opt. Lett., 19 (11)*, pages 780–782, 1994.

[11] M. G. Gustafsson. Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of microscopy, 198 (2)*, 2000.

[12] O. Solomon, Y. C. Eldar, M. Mutzafi, and M. Segev. SPARCOM : Sparsity based super-resolution correlation microscopy. *SIAM Journal on Imaging Sciences, 12 (1)*, pages 392–419, 2019.

[13] Gili Dardikman-Yoffe and Yonina C. Eldar. Learned sparcom : unfolded deep super-resolution microscopy. *Opt. Express*, 28(19) :27736–27763, Sep 2020.

[14] Mangal Prakash, Alexander Krull, and Florian Jug. Fully unsupervised diversity denoising with convolutional variational autoencoders, 2020.

[15] Roger Yong. Variational Autoencoder(VAE). Jul 8, 2021. .

[16] Stephen Odaibo. Tutorial : Deriving the standard variational autoencoder (vae) loss function, 2019.

[17] David Dao (https ://stats.stackexchange.com/users/98120/david dao). How does the reparameterization trick for vaes work and why is it important ? Cross Validated. URL :https ://stats.stackexchange.com/q/199605 (version : 2016-03-02).

[18] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn : Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Schmidt-U. Boothe T. Weigert, M. Content-aware image restoration : pushing the limits of fluorescence microscopy., 2018.

[20] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It takes (only) two : Adversarial generator-encoder networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[21] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE : Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[22] Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-vaes, 2019.

A

Appendix

A more common notation to write (1) is :

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

where $\mathbf{A}$ is the linear transformation and $\mathbf{x}$, $\mathbf{y}$ and n are viewed as vectors. However, when working in image processing the matrix A will rearly be constructed, and we will not reshape the image $\mathbf{x}$ as a vector. It is important, as we have observed in the lab, that even though we use the notations $\mathbf{A}$, we do not actually construct the matrix.

In this lab, we considerer that the noise, n follows a multidimensional normal law of covarience $\sigma^2 I$ and of mean the zero vector. So, we can write : $n \sim \mathcal{N}(0, \sigma^2 I)$.

The probability density function $p_n(n)$ is written as :

$$p_n(n) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{\|n\|_2^2}{2\sigma^2}\right) \tag{A.1}$$

where N is the number of pixels and $\|n\|_p$ is the l^p-norm defined as :

$$\|n\|_p = \left(\sum_{i=1}^{N} |n|^p\right)^{\frac{1}{p}} \tag{A.2}$$

We want to find the unknown image x from the observation y. Therefore, we use the maximum likelihood estimation which maximizes the likelihood $L(y, x)$ with respect to the unknown image x. This likelihood is equal to the conditional probability of y knowing x, denoted $p_{y|x}(y|x)$. The probability is calculated from the image model, supposing that the n is white Gaussian noise.

More precisely, the likelihood $L(y, x)$ is given by :

$$L(\mathbf{y}, \mathbf{x}) = p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) = p_n(n = \mathbf{A}\mathbf{x} - \mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2}{2\sigma^2}\right) \tag{A.3}$$

We search for an estimation $\hat{x}$ of the real image x by maximizing the ligelihood $L(y, x)$ as follows :

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} L(\mathbf{y}, \mathbf{x}) \tag{A.4}$$

In order to avoid the difficulties related to the exponential, we often maximize the logarithm of the likelihood (which does not change the maximum argument since the logarithm is strictly increasing) :

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \ln(L(\mathbf{y}, \mathbf{x})) = \arg\max_{\mathbf{x}} \left(-\ln\left(2\pi\sigma^2\right)^{\frac{N}{2}} - \frac{1}{2\sigma^2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \right) \tag{A.5}$$

The term $-\ln\left(2\pi\sigma^2\right)^{\frac{N}{2}}$ is a constant with respect to x, and thus does not intervien in the estimation of arg max. Therefore, we can write :

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \left(-\frac{1}{2\sigma^2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \right) \tag{A.6}$$

The last step removes the proptionality coefficent $\frac{1}{2\sigma^2}$ and the negative sign by using the fact that $\arg\max_{\mathbf{x}} -f(\mathbf{x}) = \arg\min_x f(\mathbf{x})$. So, finally :

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \tag{A.7}$$

B

Appendix

By considering following representation :

x Evidence or Data,

z Latent variable,

$p(\mathbf{x})$ Evidence probability,

$p(\mathbf{z})$ Prior probability,

$p(\mathbf{z}|\mathbf{x})$ Posterior probability and

$p(\mathbf{x}|\mathbf{z})$ Likelihood probability

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} \tag{B.1}$$

KL divergence is always non-negative, so :

$$D_{KL}(q(\mathbf{x})||p(\mathbf{x})) = -\int q(x\mathbf{x}) log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x} \geq 0 \tag{B.2}$$

Closed form VAE Loss : Gaussian Latents

Say we choose :

$$p(\mathbf{z}) \rightarrow \frac{1}{\sqrt{2\pi\sigma_p^2}} exp\left(-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2})\right) \tag{B.3}$$

and

$$q_\theta(\mathbf{z}|x_i) \rightarrow \frac{1}{\sqrt{2\pi\sigma_q^2}} exp\left(-\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2})\right) \tag{B.4}$$

then the KL in the ELBO becomes :

$$- D_{KL}(q_\theta(\mathbf{z}|x_i)||p(\mathbf{z})) = \int \frac{1}{\sqrt{2\pi\sigma_q^2}} exp\left(-\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2})\right) log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_p^2}} exp\left(-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2})\right)}{\frac{1}{\sqrt{2\pi\sigma_q^2}} exp\left(-\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2})\right)}\right) dz \tag{B.5}$$

Evaluating the term in the log simplifies

$$\int \frac{1}{\sqrt{2\pi\sigma_q^2}} exp\left(-\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2})\right) \times$$

$$\left(-\frac{1}{2}log(2\pi) - log(\sigma_p) - \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2}log(2\pi) + log(\sigma_q) + \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2})\right) dz \tag{B.6}$$

further simplifies into,

$$= \frac{1}{\sqrt{2\pi\sigma_q^2}} \int exp\left(-\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2}\right) \left(log(\frac{\sigma_q}{\sigma_p}) - \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} + \frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2}\right) dz \tag{B.7}$$

representing as an Expectation :

$$= \mathbb{E}_q\left(log(\frac{\sigma_q}{\sigma_p}) - \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} + \frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2}\right) \tag{B.8}$$

$$= log(\frac{\sigma_q}{\sigma_p}) - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left((\mathbf{x}-\mu_p^2)\right) + \frac{1}{2\sigma_q^2}\mathbb{E}_q\left((\mathbf{x}-\mu_q^2)\right) \tag{B.9}$$

we know that

$$\sigma_q^2 = \mathbb{E}_q((\mathbf{x}-\mu_q)_2)$$

so, the B.9 becomes

$$= log(\frac{\sigma_q}{\sigma_p}) - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left((\mathbf{x}-\mu_p^2)\right) + \frac{\sigma_q^2}{2\sigma_q^2} \tag{B.10}$$

$$= log(\frac{\sigma_q}{\sigma_p}) - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left((\mathbf{x}-\mu_q+\mu_q-\mu_p)^2\right) + \frac{1}{2} \tag{B.11}$$

By applying the equation

$$(a+b)^2 = a^2 + 2ab + b^2$$

we can solve up to :

$$= log(\frac{\sigma_q}{\sigma_p}) - \frac{\sigma_q^2 + (\mu_q-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2} \tag{B.12}$$

And when we take $\sigma_p = 1$ and $\mu_p = 0$, we get,

$$- D_{KL}(q_\theta(\mathbf{z}|x_i)||p(\mathbf{z})) = \frac{1}{2}(1 + log(\sigma_q^2) - \sigma_q^2 - \mu_q^2) \tag{B.13}$$