EE6151 : THEORY OF DEEP NEURAL NETWORKS

IMAGE DEBLURRING

- Course Project -

January 28, 2019

Student ID:

Namida M - EE15B123

Pradyumna V Chari - EE15B122

Shubhangi Ghosh - EE15B129

Contents

1 INTRODUCTION

Image deblurring is an important problem in computer vision owing to its several high-impact downstream applications such as medical imaging and criminal investigation. This report deals with one of the most common non-idealities occurring during the imaging process - blurring- and looks towards understanding and identifying optimal deep-learning based solutions for the same.

1.1 What is Image Blurring?

Image blurring arises because of relative motion between the scene and the imaging device. As a result, light from a scene 'point' interacts with several pixels. Physically, this manifests as a reduction in sharpness and scene information. Some different types of blurring are briefly explored below:

1.1.1 Motion of Subject

This kind of image blurring is caused by perceptible motion in the scene during the time the sensor is collecting scene information. A distinctive feature of such blurring is that the object(s) undergoing motion are blurred, while other scene features remain sharp. Such blurring may be understood better through the following image:

Figure 1: Depiction of object motion blurring (Courtesy: Darren Rowse, How to Capture Motion Blur in Photography)

1.1.2 Motion of Camera

This kind of blurring arises due to motion of the imaging sensor with respect to the entire scene. In such blurring, the entire image is "uniformly" blurred. The following graphics better describe this scenario:

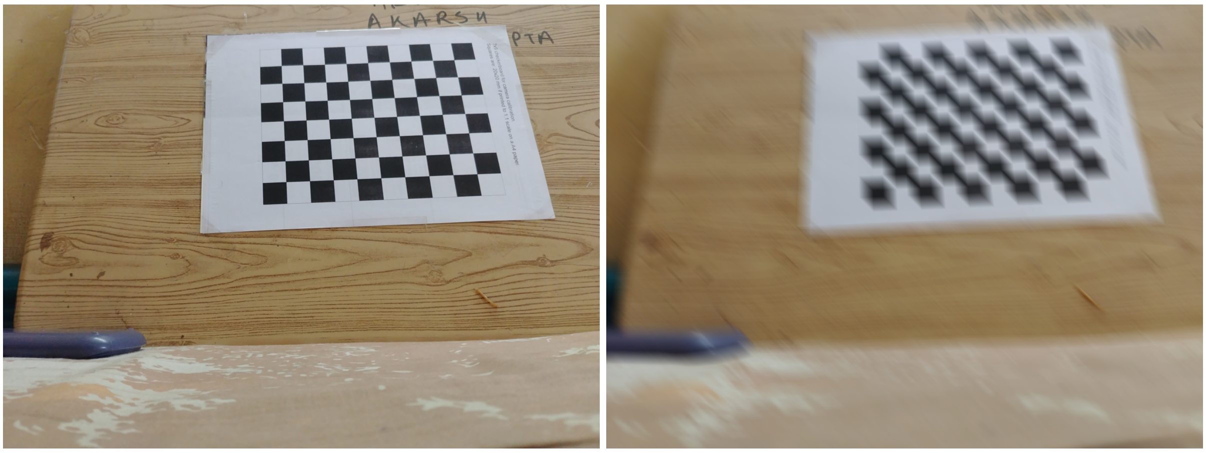

Figure 2: Depiction of camera motion blurring (Left: Unblurred Image; Right: Blurred Image)

1.1.3 Defocus Blur

This kind of blurring arises due to the subject not being in focus. The following graphic explains the process leading to defocus blur:

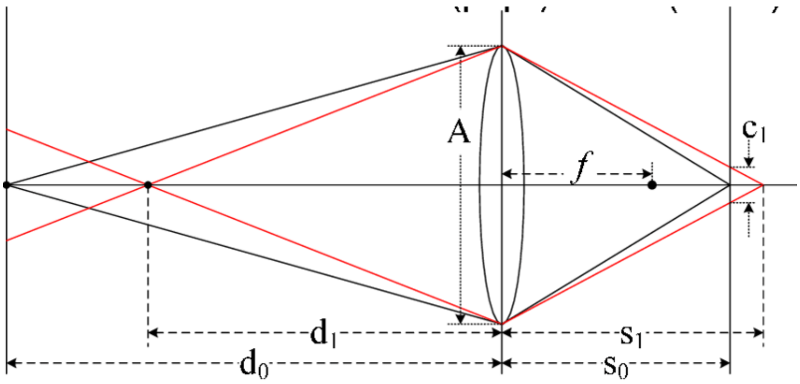

Figure 3: Generation of Defocus Blur (Courtesy: Wang et. al. Quantifying how the combination of blur and disparity affects the perceived depth)

The image of any object outside the focal plane undergoes a spread at the image plane, as can be seen. This spread can be modelled as a function of focal length, object position and camera aperture. The following image depicts such blur:

Figure 4: Depiction of Defocus Blur (Courtesy: Shutterstock)

We shall now look to understand Image Deblurring and what it entails.

2 THE IMAGE DEBLURRING PROBLEM

We try to capture the information in our surroundings using cameras with the intention of being able to recall/reconstruct/represent that information accurately sometime in the future. Images captured on a camera are made of pixels. An image contains a finite number of pixels.

The image deblurring problem involves trying to reverse the blurring process in order to extract the ground truth image. There are two classes of image deblurring. Blind and non-blind methods. In blind image deblurring we have no knowledge of the blurring kernel used. Hence, we need to solve for the optimal kernel as well as the ground truth image. In non-blind image deblurring, we have some information about the blurring kernel, although the problem is not directly solvable as the inverting the blur kernel is non-trivial. In our project we will be dealing with blind image deblurring.

Under the assumptions of spatially invariant blur kernels, additive noise, neglibible effect of camera response function(or preprocessing to compensate for it, as in [9] we get the following relation between the blurred image, the latent sharp image and the (approximate blur kernel)

$$y = x * h + n$$
$$= Ax + n$$

where y is the blurred image , x is the sharp image and n is additive noise(often considered to be Gaussian). A refers to the blur matrix(an alternate representation to the convolution form)

2.1 Setting up a Probabilistic Framework

Bayesian Inference is extremely powerful as it helps us arrive at the most likely hypothesis based on evidence. For blind image deblurring we have:

$$p(x,h|y) = \frac{p(y|x,h)p(h|x)p(x)}{p(y)} = \frac{p(y|x,h)p(h)p(x)}{p(y)}$$

The blur kernel is assumed to be independent of the image. This makes intuitive sense as the blur arises out of the conditions of the camera and its surroundings, and is not always specific to the content of the image.
To formally define our objective function we make use of the Maximum a Posteriori Estimator (MAP).

$$(x^*, A^*) = \text{argmax }_{x,A} p(x,A|y)$$
$$= \text{argmax }_{x,A} p(y|x,A)p(x)p(A)$$

In [6] and [2] we see that neural network implementations give very convincing deblurred images. We want to understand if the optimal solution to the blind deblurring problem can be represented by these architectures. If that is the case, we hope to be able to represent the optimal solution using layers of a neural network(using operations that are valid fir current architectures) . We can then proceed to validate if the solution the neural network finds has a possibility of being close to the optimal solution. If not, we should be able to suggest an optimal architecture.
In order to develop some intuition about what we are looking for we first explore some popular deep learning approaches to this problem.

2.2 The Era of Deep Learning

There are many instances in this world where a variable depends on many parameters in a very complex way, such that it is impossible for us to obtain any analytical expressions/equations that are able to exactly define the relationship. Technology has advanced to the extent where we have a huge amount of data and the ability to process such data. Deep neural networks form the bridge between data and its complex underlying functions.

2.3 Deep learning Primer

Neural networks were inspired by the structure of our brain. The backpropagation learning algorithm makes use of the chain rule in differentiation to update the parameters involved.

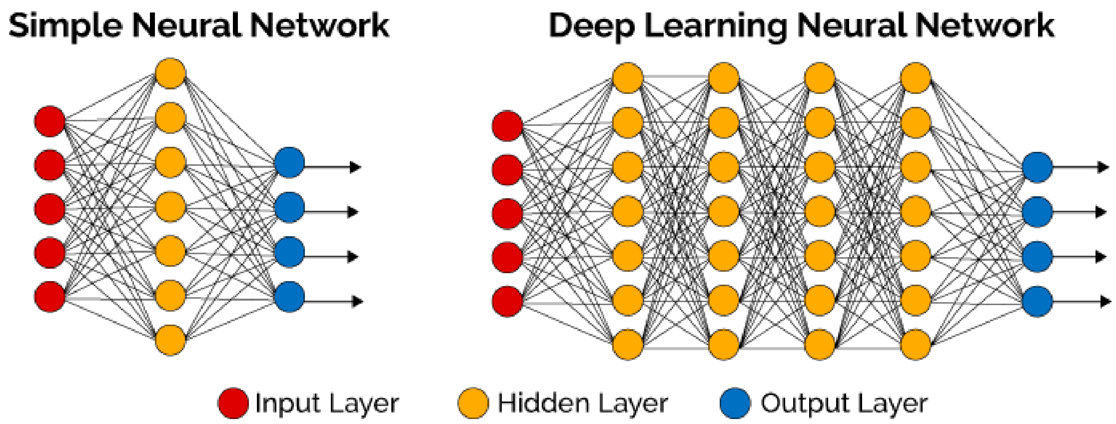

Figure 5: Possible Neural Network Architecture for the Gaussian Model

As the name suggests , deep neural networks are made 'deeper' by adding more hidden layers. According [8] deeper neural networks are able to achieve a similar approximation error than their shallower counterparts even when they use a smaller number of parameters.

2.4 The Deep Learning Approach to Deblurring

As mentioned in the previous section the aim of this project is to be able to come up with an optimal solution to the blind image deblurring problem that we can represent in terms of the layers of a deep neural network. This will hopefully help us get some more insight into what actually happens inside the black box. A better understanding of the current solution would allow us to design better systems in the future.

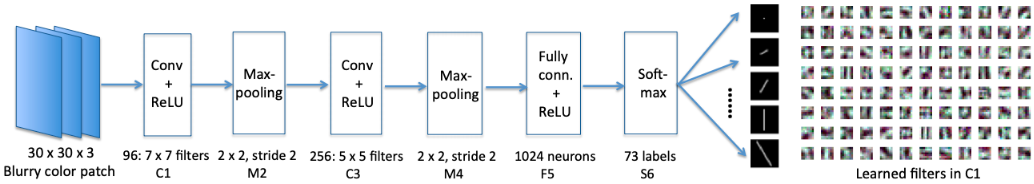

Figure 6: NN for motion direction classification as used in [6]

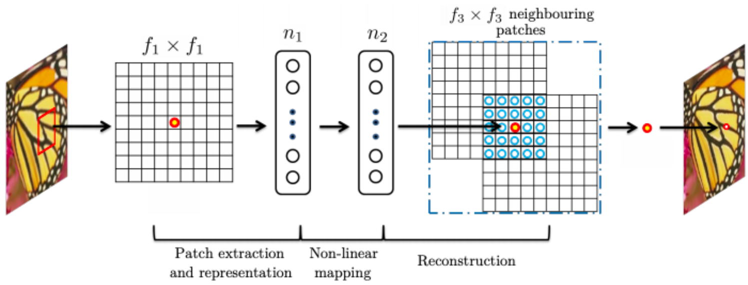

Figure 7: SRCNN(Super Resolution CNN) as used in [2]

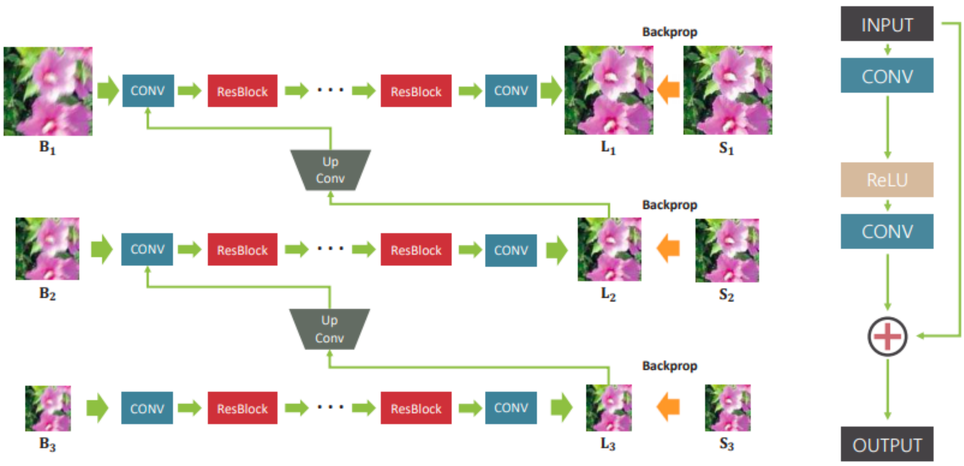

Figure 8: Multiscale network used in [10] and an expanded view of the residual block

One of the first neural network based models we considered was the model described in [6]. A convolutional neural network is trained as a classifier to identify the most likely blur direction within a patch. This information is aggregated to for the motion distribution. This information is then used to help 'deblur' the image using alternative optimisations over each patch while amking sure the overall image reconstruction error remains small. But the

solution used here is structurally very different from our approach.
The model in [2] makes use of a very simple CNN architecture to achieve remarkable results in image super-resolution(which is also an example of image restoration). [10] starts by trying to refine a coarse downsampled version of the image, and as the activations flow through the network we input downsampled versions of the blurred image as the network continues to refine the deblurred result.

3 MODELS

The solution and further analysis depends on the model for blurring being used, and the method being used to arrive at the optimal solution. In this section, we describe the models that have been analysed by us, and their respective developments.

3.1 Modelling With Poisson Noise

In this part, we explore the optimal solution for the deblurring problem in the case of Poisson noise. We first describe the imaging model being considered, followed by the optimal solution to the formulated deblurring problem. This is followed by a CNN implementation for the same.

3.1.1 Model Description

The model assumptions are described as follows (similar to the model used in [6]):

- We assume that the image is deblurred in patches, where each patch can be assumed to have been uniformly blurred.
- We assume prior knowledge of a finite set of blur kernels, any of which may act upon the image patches.
- We assume that all the blur kernels are equally likely (uniform prior)
- We assume that the log of the image prior is convolutionally realizable (reasonable assumption since image statistics are local).

3.1.2 Finding the Optimal Solution

As described earlier, we will look to use the Maximum Aposteriori Probability (MAP) solution as the optimal deblurred solution. Hence, Let m be the vectorized image patch and let A be the blurring matrix. Then, m^* is the optimal image. Let y represent the (vectorized) blurred patch.

$$f(y|m,A) = k\Pi_i e^{-a_i^T m}(a_i^T m)^{y_i}$$
$$f(m) = e^{-\lambda_1 f_1(m)}$$

Using these, the minimization expression can be written as:

$$(m^*) = argmin_m \Sigma_i((a_i^T m) - y_i log(a_i^T m)) + \lambda_1 f_1(m)$$

Taking gradient with respect to m and solving, we get:

$$(\mathbf{1} - (y./Am)^T A) + \lambda_1 \nabla_m f_1(m) = \mathbf{0}$$

Where $(./)$ is the element-wise division operator. We know that the m satisfying the above equation is optimal since the objective to be minimized is convex in m (assuming convex prior for m).
Hence, rearranging the terms gives us:

$$((y./Am)^T A)./(\mathbf{1} + \lambda_1 \nabla_m f_1(m)) = \mathbf{1}$$

Multiplying m element wise on both sides will give us,

$$m^T.*((y./Am)^T A)./(\mathbf{1} + \lambda_1 \nabla_m f_1(m)) = m^T$$

Hence, the optimal solution can be seen as a stationary point to the above transformation. Applying Banach Fixed Point Theorem, the following iterations can be set up to arrive at the optimal solution:

$$m^{i+1^T} = m^{i^T}.*((y./Am^i)^T A)./(\mathbf{1} + \lambda_1 \nabla_m f_1(m^i))$$

3.1.3 Understanding the Iterative Equation

The following points highlight some features of the above iterative equations:

- For uniform blurring, the matrix multiplication can be replaced by a convolution
- The iterations are also the Expectation Maximization iterations for the convex cost function (as shown in [3]); hence these iterations are guaranteed to reach a minima

- The iterations involve element-wise multiplications and divisions
- The deblurring problem has now been reduced to a classification problem; the network must identify optimal blur kernel for the patch

In order to provide an intuition for the operation of these equations, the same were applied to an image from the MNIST dataset. Note that an ML estimation was carried out (i.e image prior was assumed to be uniform).

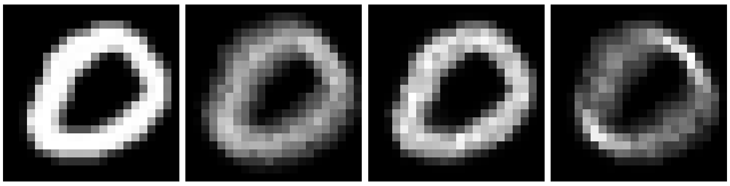

Figure 9: Left to Right: Original image, Blurred image, Unblurred image (using correct blur kernel assumption), Unblurred image (using incorrect blur kernel assumption)

The following can be noted from the above results:

- The blurring significantly reduces both the contrast as well as the sharpness of the image
- The deblurring process (with correct blur kernel assumption) improves both the contrast and the sharpness to a large extent, in a small number of iterations (eight in this case)
- The deblurring process (with incorrect blur kernel assumption) further degrades the image
- The correctly and incorrectly deblurred images are distinctively different, and may be distinguished by using a prior model
- Presence of an accurate prior model may further help with improving the performance, since the above results were obtained using a uniform prior model for the image

These observations provide further motivation to interpret this iterative equation as an optimal CNN implementation, given the problem formulation.

3.1.4 Realizing the CNN Architecture

- Element wise multiplications and divisions can be realized by using alternating log(.) and exp(.) gates as follows:

$$m^{i+1^T} = exp(log(m^{i^T}) + log(exp(log(y) - log(Am^i))^T A) - log(\mathbf{1} + \lambda_1 \nabla_m f_1(m^i)))$$

- The gradient term can be realized using a convolutional layer; intuitively, if the prior was Normal, this would mean pixels have non-zero correlations with neighbouring pixels
- y and m can be propagated using identity layers
- Iterations can be carried out for all the possible blur kernels; most probable kernel can be selected using a fully connected layer followed by a softmax layer, that estimates the most likely m based on the prior model

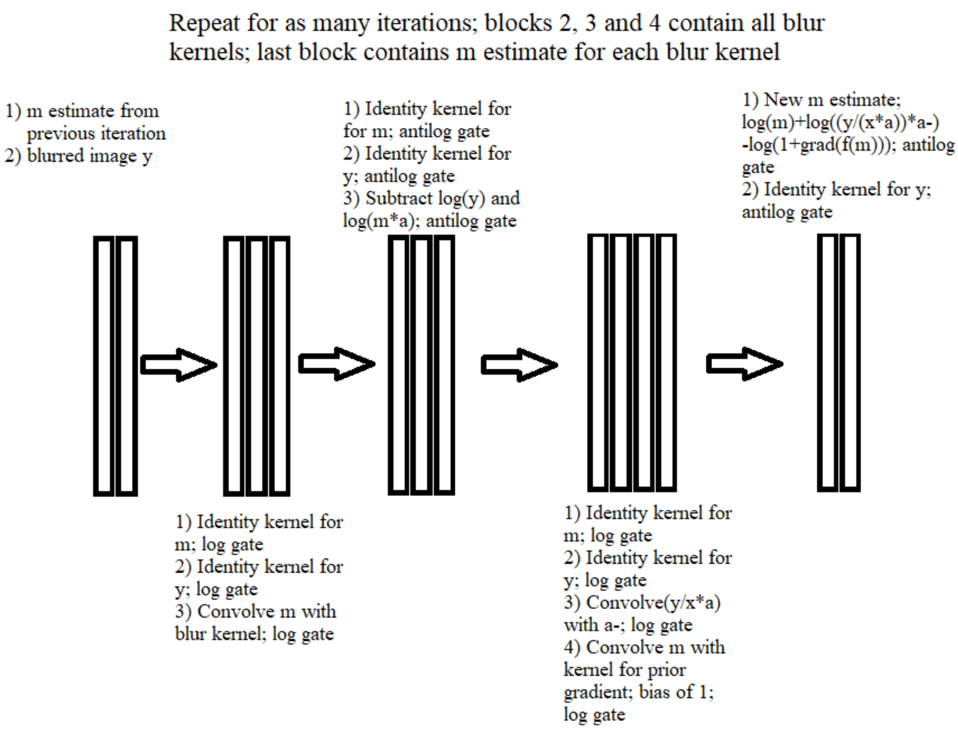

Figure 10: Representation of the CNN Implementation

3.1.5 Drawbacks

While this implementation arrives at one way of visualizing an optimal CNN architecture for image deblurring, it possesses a few drawbacks which need to be highlighted:

- Log and exp gates unconventional and may not be practically usable due to finite precision and behavior near zero
- Propagation of y (input blurred image) is not observed in practical CNNs
- We assume prior knowledge of blur kernels, which may not be the case always
- Each layer must contain convolutional kernels corresponding to every possible blur kernel, thereby increasing complexity of the implementation

Based on these drawbacks, we attempted to extend the Poisson noise model to a scenario where there is no prior knowledge of the various blur kernels. However, due to the nature of the Poisson probability distribution, the joint optimization problem quickly becomes complex. As a result, we shifted focus and looked towards analyzing the model described in the following subsection.

3.2 Modelling With Gaussian Noise

We now look at a different formulation for image blurring and subsequent deblurring, in the presence of Gaussian noise. The details for the model are covered in the following subsection.

3.2.1 Model Description

- We assume that the imaging process involves Gaussian noise
- No prior knowledge of blur kernel is assumed; the most likely blur kernel is estimated
- Gaussian priors are assumed for both the unblurred image and the blur kernel. Hence, the negative log of both these priors is convex in their respective variables

3.2.2 Finding the Optimal Solution

The Maximum Aposteriori Probability (MAP) solution is used as the optimal deblurred solution.

As in the previous model, let m be the vectorized image patch and let A be the blurring matrix. Then, m^* is the optimal image and A^* is the optimal blur kernel. Let y represent the blurred patch.

Our objective is to find m and A which maximise the posterior probability.

$$\max_{\mathbf{m},\mathbf{A}} p(\mathbf{m},\mathbf{A}|\mathbf{y})$$

$$\Leftrightarrow max_{\mathbf{m},\mathbf{A}} \frac{p(\mathbf{m},\mathbf{A},\mathbf{y})}{\mathbf{y}}$$

Since an uniform prior over blurred patches is assumed, the denominator **y** can be neglected.

$$\Leftrightarrow max_{\mathbf{m},\mathbf{A}} p(\mathbf{m},\mathbf{A},\mathbf{y})$$

$$\Leftrightarrow max_{\mathbf{m},\mathbf{A}} p(\mathbf{m})\, p(\mathbf{A}|\mathbf{m})\, p(\mathbf{y}|\mathbf{A},\mathbf{m})$$

Since the blur kernel is independent of the blur patch,

$$\Leftrightarrow max_{\mathbf{m},\mathbf{A}} p(\mathbf{m})\, p(\mathbf{A})\, p(\mathbf{y}|\mathbf{A},\mathbf{m})$$

$$\Leftrightarrow max_{\mathbf{m},\mathbf{A}} e^{-\lambda_1 f_1(\mathbf{m})} e^{\lambda_2 f_2(\mathbf{A})} \Pi_{i=1}^{n} e^{-\frac{(y_i - a_i^T m)^2}{2\sigma^2}}$$

Taking negative-log,

$$\Leftrightarrow min_{\mathbf{m},\mathbf{A}} \lambda_1 f_1(\mathbf{m}) + \lambda_2 f_2(\mathbf{A}) + \Sigma_{i=1}^{n} \frac{(y_i - a_i^T m)^2}{2\sigma^2}$$

The objective to be minimized is non-convex owing to the third term. Hence, we employ **Alternating Minimization** to approach the optimal values for **m** and **A**.

$$\mathbf{m}^+ = min_{\mathbf{m}} \lambda_1 f_1(\mathbf{m}) + \lambda_2 f_2(\mathbf{A}) + \Sigma_{i=1}^{n} \frac{(y_i - a_i^T m)^2}{2\sigma^2}$$

where **A** is fixed.

$$\mathbf{A}^+ = min_{\mathbf{A}} \lambda_1 f_1(\mathbf{m}) + \lambda_2 f_2(\mathbf{A}) + \Sigma_{i=1}^{n} \frac{(y_i - a_i^T m)^2}{2\sigma^2}$$

where **m** is fixed.

Finding optimal value of m by minimising w.r.t A

At the minima,

$$\lambda_1 \nabla f_1(\mathbf{m}) - \frac{A^T (y - Am)}{\sigma^2} = 0$$

$$\lambda_1 \nabla f_1(\mathbf{m}) - \Sigma_{i=1}^{n} \frac{y_i a_i}{\sigma^2} + \Sigma_{i=1}^{n} a_i^T m \frac{a_i}{\sigma^2} = 0$$

$$\lambda_1 \nabla f_1(\mathbf{m}) - \Sigma_{i=1}^{n} \frac{y_i a_i}{\sigma^2} + \frac{1}{\sigma^2} \Sigma_{i=1}^{n} a_i a_i^T m = 0$$

Let $\nabla f_1(\mathbf{m}) = g_1(\mathbf{m})$, $\Sigma_{i=1}^{n} \frac{y_i a_i}{\sigma^2} = \mathbf{u}$ and $\Sigma_{i=1}^{n} a_i a_i^T = \tilde{\mathbf{A}}$.
$\tilde{\mathbf{A}}$ is invertible because $\mathbf{a}_i$s are chosen independently.
(In our attempt to prove convergence of Alternating Minimization, we have taken the above assumptions into consideration.)

$$g_1(\mathbf{m}) - \mathbf{u} + \tilde{\mathbf{A}}\mathbf{m} = 0 - (1)$$

$$\mathbf{m} = \tilde{\mathbf{A}}^{-1}(\mathbf{u} - g_1(\mathbf{m}))$$

The value of $\mathbf{m}^+$ can also be found as the value of $\mathbf{m}$ which minimizes (1) by numerical methods.
If we choose $g_1(\mathbf{m}) = \mathbf{Bm} + b$, then the update equation is as follows:

$$\mathbf{m}^+ = (\lambda_1 \mathbf{B} + \frac{\mathbf{A^T A}}{\sigma^2})^{-1} \frac{\mathbf{A^T y}}{\sigma^2} - b$$

Finding optimal value of A by minimising w.r.t m
At the minima,

$$\lambda_2 \nabla f_2(\mathbf{A}) + \frac{1}{2\sigma^2} \nabla_A \|\mathbf{y} - \mathbf{Am}^+\| = 0$$

$$(\nabla_A \|\mathbf{y} - \mathbf{Am}^+\| = -2(\mathbf{y} - \mathbf{Am}^+)(\mathbf{m}^{+^T}))$$

$$\implies \nabla f_2(\mathbf{A}) - \frac{1}{\sigma^2}(\mathbf{y} - \mathbf{Am}^+)(\mathbf{m}^{+^T})) = 0$$

$$\nabla f_2(\mathbf{A}) + \mathbf{A}\frac{\mathbf{m}^+\mathbf{m}^{+^T}}{\sigma^2} = \frac{1}{\sigma^2}\mathbf{ym}^{+^T}$$

If $\nabla f_2(\mathbf{A}) = \mathbf{A}c$, s.t. $c + \frac{\mathbf{m}^+\mathbf{m}^{+^T}}{\sigma^2}$ is full-rank,
(In our attempt to prove convergence of Alternating Minimization (AM), we have taken the above assumptions into consideration.)

$$\mathbf{A}^+ = \frac{1}{\sigma^2}(\mathbf{ym}^{+^T})(c + \frac{\mathbf{m}^+\mathbf{m}^{+^T}}{\sigma^2})^{-1}$$

3.2.3 Neural Network Architecture for Gaussian Model

As mentioned in the previous section , due to the non-convex nature of the problem, we separated the objective function into two separate (marginally) convex objectives. We tried to hypothesise a plausible architecture by interpreting the struture of our solution.

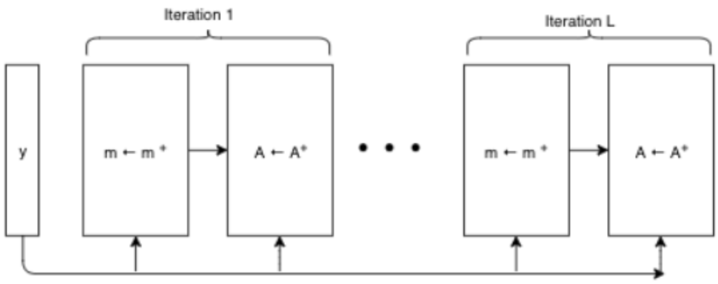

Figure 11: Possible Neural Network Architecture for the Gaussian Model

One possibility is a network in where each block is able to perform an iteration of the AM update(as shown in Fig 11). The number of such blocks would indicate the number of iterations performed. Each block consists of sub-blocks, one for the update of $\mathbf{A}$ and the other for $\mathbf{m}$. Since each iteration uses values from the previous iteration, skip connections will need to be employed.
Fixing the size of the blocks and sub-blocks places constraints that we can perform in any iterative process. The size of the sub-blocks can be significantly reduced if we are able to find a closed form solution for the fixed point updates. As discussed in previous sections this is achievable by choosing the priors appropriately. Division operations if any can be performed using the binomial approximation. But, for such an architecture to be feasible , we need to account for convergence and rate of convergence of the AM alogrithm . We look in to this in more detail in the following section.

3.2.4 Finding the Optimal Solution Using Alternating Minimization

As described earlier, we shall use an alternating minimzation based approach to find the optimal deblurred image and the corresponding blur kernel. Since our end goal is to find a Neural Network interpretation for this approach, we must keep a few things in mind:

- The alternating minimization approach must converge at the optimal solution
- The convergence must take place at a "reasonably quick" rate

We therefore refer to [4], to establish requirements that the objective function must satisfy so that it is suitable for our purpose. We begin by first defining a few properties of the objective function.

Marginal Strong Convexity and Strong Smoothness: A continuously differentiable function $f : R^p \times R^q \rightarrow R$ is considered (uniformly) α-marginally strongly convex (MSC) and (uniformly) β-marginally strongly smooth (MSS) in its first variable if for every value of $y \in R^q$, the function $f(., \mathbf{y}) : R^p \rightarrow R$ is α-strongly convex and β-strongly smooth. That is, for every x^1, $x^2 \in R^p$,

$$\frac{\alpha}{2}||x^2 - x^1||_2^2 \leq f(x^2, y) - f(x^1, y) - \langle g, x^2 - x^1 \rangle \leq \frac{\beta}{2}||x^2 - x^1||_2^2$$

Where $g = \nabla_x f(x^1, y)$. A similar definition can be extended for the other variable too.
Robust Bistability: A function $f : R^p \times R^q \rightarrow R$ satisfies the C-robust bistability property if for some $C > 0$, for every $(x, y) \in R^p \times R^q$, we have

$$f(x, y^*) + f(x^*, y) - 2f(x^*, y^*) \leq C(2f(x, y) - f(x, \bar{y}) - f(\bar{x}, y))$$

Where $\bar{x}$ minimizes $f(x, y)$ for a given y and $\bar{y}$ minimizes $f(x, y)$ for a given x.
With these properties understood, we now describing a **theorem** that establishes the nature of convergence for alternating minimization.

Let $f : R^p \times R^q \rightarrow R$ be a continuously differentiable (but possibly non-convex) function that satisfies the properties of Marginal Strong Convexity and Smoothness, and C-robust bistability. Then, if we use alternating minimization on this function, after at most $T = \mathcal{O}(log\frac{1}{\epsilon})$ steps, we have $f(x^T, y^T) \leq f(x^*, y^*) + \epsilon$.

With this foundation, we can now look to find prior models that help us obtain an objective function that satisfies the above properties.

3.2.5 Identifying Suitable Prior Models for Alternating Minimization

As explained earlier the objective function $f(m, A) = \lambda_1 f_1(\mathbf{m}) + \lambda_2 f_2(\mathbf{A}) + \Sigma_{i=1}^n \frac{(y_i - a_i^T m)^2}{2\sigma^2}$ has to minimized. We have made the following assumptions on priors:

- $\nabla_m f_1(m) = Bm + b$, where B is an appropriate matrix and b is a suitably sized vector. That is, we assume a Gaussian prior model for the image

- $\nabla_A f_2(A) = AC$, where C is a suitable matrix. That is, we assume a matrix Gaussian prior for the blur kernel

Then, as covered earlier, the alternating minimization updates are as follows:

$$\mathbf{m}^+ = (\lambda_1 \mathbf{B} + \frac{\mathbf{A^T A}}{\sigma^2})^{-1} \frac{\mathbf{A^T y}}{\sigma^2} - b$$

$$\mathbf{A}^+ = \frac{1}{\sigma^2}(\mathbf{y}\mathbf{m}^{+^T})(c + \frac{\mathbf{m}^+\mathbf{m}^{+^T}}{\sigma^2})^{-1}$$

Hence, the overall cost function at hand is given by:

$$f(m, A) = (m+b)^T B(m+b) + Tr(A^T CA) + \frac{1}{2\sigma^2}(y - Am)^T(y - Am)$$

Proving Marginal Strong Convexity and Smoothness

To prove marginal strong convexity and smoothness, the following must be shown to hold true:

Condition for strong convexity: $(\nabla f(x) - \nabla f(y))^T(x - y) \geq \mu||x - y||^2$
Condition for strong smoothness: $(\nabla f(x) - \nabla f(y))^T(x - y) \leq \eta||x - y||^2$

We need to prove that these hold for both m and A. (Note that while these conditions are different from the basic definition of strong convexity and strong smoothness, the two can be shown to be equivalent).
We start with proving the same for m. The **reasoning** is as follows:
Let $\bar{f}(m)$ be the part of the objective function that depends on m. Hence, if we can show that $\bar{f}(m)$ is strong convex and strong smooth, then it implies that $f(m, A)$ is marginally strong convex and strong smooth in m.

$$\bar{f}(m) = (m+b)^T B(m+b) + \frac{1}{2\sigma^2}(y - Am)^T(y - Am)$$

$$\nabla_m \bar{f}^T = (m+b)^T(B + B^T) - \frac{y^T A}{\sigma^2} + \frac{m^T A^T A}{\sigma^2}$$

Therefore,

$$(\nabla_m \bar{f}(m_1) - \nabla_m \bar{f}(m_2))^T = (m_1 - m_2)^T(B + B^T + \frac{A^T A}{\sigma^2})$$

This implies that,

$$(\nabla_m \bar{f}(m_1) - \nabla_m \bar{f}(m_2))^T (m_1 - m_2) = (m_1 - m_2)^T (B + B^T + \frac{A^T A}{\sigma^2})(m_1 - m_2)$$

Therefore, for strong convexity to be true,

$$(m_1 - m_2)^T (B + B^T + \frac{A^T A}{\sigma^2})(m_1 - m_2) \geq \mu (m_1 - m_2)^T I (m_1 - m_2)$$

Where I is the identity matrix. That is,

$$B + B^T + \frac{A^T A}{\sigma^2} \geq \mu I$$

must hold true. This means that the difference of the two sides of the equation must be a positive semi-definite matrix. Based on a similar argument, the following must hold true for strong smoothness to hold true:

$$B + B^T + \frac{A^T A}{\sigma^2} \leq \eta I$$

The appropriate values for μ and η can be found based on knowledge of the maximum and minimum Eigen values of $B + B^T + \frac{A^T A}{\sigma^2}$.
Therefore, through the above reasoning, we have shown that $\bar{f}(m)$ is strong convex and strong smooth. Hence, we have shown that our **objective function** $f(m, A)$ **is marginally strong convex and marginally strong smooth in** m**.**

Using a similar formulation as above, and by making use of the fact that the prior model for the blur kernel A is Gaussian, it can also be shown that our **objective function** $f(m, A)$ **is marginally strong convex and marginally strong smooth in** A**.**

Proving C-Robust Bistability

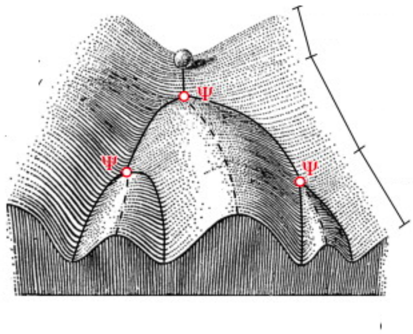

Figure 12: Intuitive understanding of Bistability (Courtesy: Waddington's epigenetic landscape)

Recall that the following condition must hold true for our objective function to be C-robust bistable:

$$f(m, A^*) + f(m^*, A) - 2f(m^*, A^*) \le C(2f(m, A) - f(m, \bar{A}) - f(\bar{m}, A))$$

for any (m, A). We shall now describe the methods we used to tackle this problem:

1. - **Intuition**: Try to analyse terms in the inequality case by case. We attempt to show that $f(m, A^*) - f(m^*, A^*) \le C(f(m, A) - f(\bar{m}, A))$ and $f(m^*, A) - f(m^*, A^*) \le C(f(m, A) - f(m, \bar{A}))$.
   - **Drawback**: Such an approach is unlikely to work. Consider the terms $f(m, A^*) - f(m^*, A^*)$ and $f(m, A) - f(\bar{m}, A)$. Consider if m is close to m^*, then LHS can be arbitrarily small while RHS is finite. However, if m is close to $\bar{m}$ then RHS is arbitrarily small while LHS is finite. Hence, the same constant C might not work. This gives us the intuition that we must look into the behaviour of both terms together. However, we are unable to account for the joint characteristics of the function. We also need to have a more intuitive understanding of the bistability condition, since a bistable point need not be a local minima. Consider the (Figure:12). We have interpreted bistable points as those points that are local minima when you cut the figure along atleast 2 planes.

2. - **Intuition**: Direct substitution of update equations in the bistability expression, in order to establish the required relation.
   - **Drawback**: The expression does not simplify in a tractable way. Attempts were made to simplify the resulting expressions using algebraic identities such as the Matrix Inversion Lemma. However, the expression could not be reconciled into the required form.

4 RECURRENT CONNECTIONS

4.1 Interpretation as a Progressive ConvLSTM Neural Network

The practical algorithms for deblurring which seemed most directly linked to our current alternating minimisation theory are Multi-scale convolutional/convLSTM networks [7], [5]. Multi-scale networks basically go from lower resolutions to higher resolutions in progressive steps. They have a neural model which they first apply on the blurred image to obtain an image of an intermediate resolution, on which they further apply the model to obtain higher resolutions.

Multi-scale neural networks have been widely used for the deblurring problems and have shown great improvements in results. The alternating minimization approach that emerged as the ideal form to solve the Maximum Aposteriori Probability formulation provides a strong theoretical justification for the success of Multi-scale convolutional networks.

4.2 Recurrence in Deblurring

The deblurring process can be formulated by an infinite impulse response (IIR) model [10]. Given a 1D signal x and a blur kernel k, the blur process can be formulated as:

$$y[n] = \sum_{m=0}^{M} k[m]x[n-m]$$

where y is the blurred signal, m represents the position of the signal, and M is the size of the blur kernel. The clear signal x can be obtained by,

$$x[n] = \frac{y[n]}{k[0]} - \frac{k[m]x[n-m]}{k[0]} - (2)$$

. which is an M-th order infinite impulse response (IIR) model. By expanding the second term, we find that the deconvolution process requires an infinite signal information as follows:

$$x[n] = \frac{y[n]}{k[0]} - \frac{k[m]}{k[0]}\left(\frac{y[n-m]}{k[0]} - \frac{\sum_{l=1}^{M} k[l]x[n-m-l]}{k[0]}\right)$$

$x[n-m-l]$ can be further expanded to give an infinite formulation.

The recurrent formulation in (2) is exploited to reduce the number of parameters of the infinite context.

5 CONCLUSIONS

1. The processes of image blurring and deblurring were explicitly discussed, understood and modelled
2. A robust probabilistic framework for finding the optimal deblurring solution was established
3. Rigorous analysis was carried out and an architecture was proposed for deblurring in the presence of Poisson noise. The proposal for the architecture was further bolstered through simulated runs of the iterative deblurring formulation
4. Extensive analysis was carried out for the case of deblurring in the presence of Gaussian noise. Various algorithms for optimization were explored and analyzed, while keeping in mind the eventual objective of linking these algorithms with Deep Network realizations
5. Work was carried out to better understand the application of Recurrent Neural Networks towards optimal image deblurring

6 FUTURE STEPS

- Interpretation of iterative solutions through recurrent neural networks
- Alternative methods for non-convex optimisation
- Convergence criteria for Alternating Minimisation for marginally convex (or non-convex) objectives
- Exploration of discrete models for deblurring. There is some existing work [1] that has proved convergence for such models with the help of some regularisers. It would be interesting to interpret these regularisers as priors
- Latent Variable models for image deblurring

Bibliography

[1] Tony F Chan and Chiu-Kwong Wong. "Total variation blind deconvolution". In: *IEEE transactions on Image Processing* 7.3 (1998), pp. 370–375.

[2] Chao Dong et al. "Learning a deep convolutional network for image super-resolution". In: *European conference on computer vision*. Springer. 2014, pp. 184–199.

[3] P J. Green. "On Use of the EM Algorithm for Penalized Likelihood Estimation". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 52 (Jan. 1990). DOI: 10.2307/2345668.

[4] Prateek Jain and Purushottam Kar. "Non-convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10 (Dec. 2017), pp. 142–336. DOI: 10.1561/2200000058.

[5] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. "Deep multi-scale convolutional neural network for dynamic scene deblurring". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3883–3891.

[6] Jian Sun et al. "Learning a convolutional neural network for non-uniform motion blur removal". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 769–777.

[7] Xin Tao et al. "Scale-recurrent network for deep image deblurring". In: *arXiv preprint arXiv:1802.01770* (2018).

[8] Matus Telgarsky. "Benefits of depth in neural networks". In: *arXiv preprint arXiv:1602.04485* (2016).

[9] Ruxin Wang and Dacheng Tao. "Recent progress in image deblurring". In: *arXiv preprint arXiv:1409.6838* (2014).

[10] Jiawei Zhang et al. "Dynamic Scene Deblurring Using Spatially Variant Recurrent Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2521–2529.