Topics in Landmarking and Elementwise Mapping

Mehmet Kerem Turkcan

Submitted in partial fulfillment of the
requirements for the degree
of Master of Science
in the Fu Foundation School of Engineering & Applied Science

COLUMBIA UNIVERSITY

2016

ABSTRACT

Topics in Landmarking and Elementwise Mapping

Mehmet Kerem Turkcan

We consider a number of different landmarking and elementwise mapping problems and propose solutions that are thematically interconnected with each other. We consider diverse problems ranging from landmarking to deep dictionary learning, pan-sharpening, compressive sensing magnetic resonance imaging and microgrid control, introducing novelties that go beyond the state of the art for the problems we discuss.

We start by introducing a manifold landmarking approach trainable via stochastic gradient descent that allows for the consideration of structural regularization terms in the objective. We extend the approach for semi-supervised learning problems, showing that it is able to achieve comparable or better results than equivalent k-means based approaches. Inspired by these results, we consider an extension of this approach for general supervised and semi-supervised classification for structurally similar deep neural networks with self-modulating radial basis kernels.

Secondly, we consider convolutional networks that perform image-to-image mappings for the problems of pan-sharpening and compressive sensing magnetic resonance imaging. Using extensions of deep state of the art image-to-image mapping architectures specifically tailored for these problems, we show that they could be addressed naturally and effectively.

After this, we move on to describe a method for multilayer dictionary learning and feed-forward sparse coding by formulating the dictionary learning problem using a general deep learning layer architecture inspired by analysis dictionary learning. We find this method to be significantly faster to train than classical online dictionary learning approaches and capable of addressing supervised and semi-supervised classification problems more naturally.

Lastly, we look at the problem of per-user power supply delivery on a microgrid powered by solar energy. Using real-world data obtained via The Earth Institute, we consider

the problem of deciding the amount of power to supply to all each user for a certain period of time given their current power demand as well as past demand/supply data. We approach the problem as one of demand-to-supply mapping, providing results for a policy network trained via regular propagation for worst-case control and classical deep reinforcement learning.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Firstly, I would like to thank my parents for their colossal support and endless patience during my graduate studies.

Secondly, I would like to thank my advisor, Professor John Paisley, for introducing me to problems like landmarking, pan sharpening and compressive sensing MRI that I consider in the following pages and encouraging my interest in deep learning approaches.

I would also like to thank Albert Boulanger for funding me as a Graduate Research Assistant in my final Master's semester and helping me find interesting and fun projects. Without his support I would not have been able to contribute to a number of interesting experiments in my time at Columbia. I would also like to thank Professor Vijay Modi for allowing me to work on the deep learning aspect of microgrid supply control and Akhilesh Ramakrishnan for providing the data and baseline algorithms that were required for testing the efficacy of the algorithms.

Furthermore, I would like to thank the professors I have taken courses from or worked with during my Master's studies. Specifically, I would like to thank Professor Daniel Hsu, Professor Zoran Kostic, Professor Stephen Edwards, Professor Aurel Lazar and Professor Rocco Servedio.

Finally, I would like to thank my undergraduate advisor, Professor Tayfun Akgul, for the wisdom I have acquired while working with him, without which I would not have been able to become who I am right now.

Datta. Dayadhvam. Damyata.

Introduction

In this thesis, we consider a number of different landmarking and elementwise mapping problems and propose solutions that are thematically interconnected with each other. We consider diverse problems ranging from landmarking to deep dictionary learning, pan-sharpening, compressive sensing magnetic resonance imaging and microgrid control, introducing novelties that go beyond the state of the art for the problems we discuss.

We start by introducing a manifold landmarking approach trainable via stochastic gradient descent that allows for the consideration of structural regularization terms in the objective. We extend the approach for semi-supervised learning problems, showing that it is able to achieve comparable or better results than equivalent k-means based approaches. Inspired by these results, we consider an extension of this approach for general supervised and semi-supervised classification for structurally similar deep neural networks with self-modulating radial basis kernels.

Secondly, we consider convolutional networks that perform image-to-image mappings for the problems of pan-sharpening and compressive sensing magnetic resonance imaging. Using extensions of deep state of the art image-to-image mapping architectures specifically tailored for these problems, we show that they could be addressed naturally and effectively.

After this, we move on to describe a method for multilayer dictionary learning and feed-forward sparse coding by formulating the dictionary learning problem using a general deep learning layer architecture inspired by analysis dictionary learning. We find this method to be significantly faster to train than classical online dictionary learning approaches and capable of addressing supervised and semi-supervised classification problems more naturally.

Lastly, we look at the problem of per-user power supply delivery on a microgrid powered by solar energy. Using real-world data obtained via The Earth Institute, we consider the problem of deciding the amount of power to supply to all each user for a certain period of time given their current power demand as well as past demand/supply data. We approach the problem as one of demand-to-supply mapping, providing results for a policy network trained via regular propagation for worst-case control and classical deep reinforcement learning.

Part I

Landmarking Architectures

Chapter 1

Sequential Landmarking

1.1 Introduction

For this section, we follow from previous work on landmarking manifolds with Gaussian processes [Liang and Paisley, 2015]. Let us begin with a brief overview of the model.

Given a dataset $\boldsymbol{X} \in R^{M \times N}$ of M examples, we want to calculate a number of landmarks, $\boldsymbol{l}_i$, $i \in \{1, ..., K\}$ that exemplify the dataset in some way. Given n landmarks $\boldsymbol{l}_1, ..., \boldsymbol{l}_n$, let us consider the following objective:

$$\boldsymbol{l}_{n+1} = \arg\max_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}) - k(\boldsymbol{x}, \{\boldsymbol{l}_1, ..., \boldsymbol{l}_n\}) k(\{\boldsymbol{l}_1, ..., \boldsymbol{l}_n\}, \{\boldsymbol{l}_1, ..., \boldsymbol{l}_n\})^{-1} k(\{\boldsymbol{l}_1, ..., \boldsymbol{l}_n\}, \boldsymbol{x}) \quad (1.1)$$

where we can use the RBF kernel $k(\boldsymbol{x}, \boldsymbol{y}) = c \cdot \exp\left(-\|\boldsymbol{x} - \boldsymbol{y}\|^2 / \eta\right)$ (generalized for vector or matrix output) or use the approximation described in [Liang and Paisley, 2015] to consider $\boldsymbol{X}$ during the calculations.

1.2 Sparse Landmarking via Alternating Direction Method of Multipliers

In many machine learning datasets, the examples are sparse, that is, they have a small ℓ_0 norm. To give specific examples, bag-of-words representations of sentences or digits from the MNIST database have this property. Using the convex ℓ_1 norm instead of the difficult-to-optimize ℓ_0 norm, we can use the alternating method of multipliers (ADMM) to promote sparsity in the landmarks calculated.

The general ADMM formulation considers the problem

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c} \end{aligned} \tag{1.2}$$

which is solved via iterating with the following steps:

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{z}^i, \boldsymbol{y}^i) \\ \boldsymbol{z}^{i+1} &= \arg\min_{\boldsymbol{z}} L_\rho(\boldsymbol{x}^{i+1}, \boldsymbol{z}, \boldsymbol{y}^i) \\ \boldsymbol{y}^{i+1} &= \boldsymbol{y}^i + \rho(\boldsymbol{A}\boldsymbol{x}^{i+1} + \boldsymbol{B}\boldsymbol{z}^{i+1} - \boldsymbol{c}) \end{aligned} \tag{1.3}$$

where

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{y}^T \left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}\right) + \frac{\rho}{2} \left\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}\right\|_2^2. \tag{1.4}$$

ADMM has been used for a huge variety of methods from robust PCA [Candès *et al.*, 2011] to tensor inpainting [Liu *et al.*, 2013]. Given a black-box landmarking algorithm with objective function $L_{landmark}$ to learn the next landmark, then, the problem becomes

$$\begin{aligned} \text{minimize} \quad & L_{landmark}(\boldsymbol{x}) + \lambda||\boldsymbol{z}|| \\ \text{subject to} \quad & \boldsymbol{x} - \boldsymbol{z} = 0 \end{aligned} \tag{1.5}$$

with solution

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} \; L_{landmark}(\boldsymbol{x}) + \rho/2||\boldsymbol{x} - \boldsymbol{z}^i + \boldsymbol{y}^i||_2^2 \\ \boldsymbol{z}^{i+1} &= \arg\min_{\boldsymbol{z}} \; Shrinkage_{\lambda/\rho}(\boldsymbol{x}^{i+1} + \boldsymbol{y}^i/\rho) \\ \boldsymbol{y}^{i+1} &= \boldsymbol{y}^i + \rho(\boldsymbol{x}^{i+1} - \boldsymbol{z}^{i+1}). \end{aligned} \tag{1.6}$$

where $Shrinkage_{\lambda/\rho}$ is the soft thresholding function with threshold λ/ρ. In [Liang and Paisley, 2015], the landmarks are computed with the following gradient-normalized update:

$$\begin{aligned} &\boldsymbol{\gamma} = \boldsymbol{l}_{n+1}^i + \rho_s \nabla_l f_n(\boldsymbol{l}, \boldsymbol{X}_{batch})|_{\boldsymbol{l}_{n+1}} / (\left\|\nabla_l f_n(\boldsymbol{l}, X_{batch})|_{\boldsymbol{l}_{n+1}}\right\|_2) \\ &\boldsymbol{l}_{n+1}^{i+1} = Proj_S(\boldsymbol{\gamma}) \end{aligned} \tag{1.7}$$

which we extend by changing the second step as

$$\boldsymbol{l}_{n+1}^{i+1} = Proj_S(\boldsymbol{\gamma} + \kappa\left(\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{y}\right) / \|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{y}\|_2)$$

in order to solve the first step in Equation 1.6.

This method allows one to capture the hidden structure in a given dataset. In Figure 1.1, we show two sets of 9 landmarks calculated from the MNIST dataset with and without the ℓ_1 norm term. For these results, we have used $\rho = \kappa = 10^{-2}$ and run the algorithm for the same amount of steps ($I_{max} = 1000$) for both methods with other parameters set the same way as the source paper, running the outer ADMM loop 10 times. In Figure 1.2, we show the difference between the classical landmarking algorithm and the sparse model for the first 1000 landmarks by using RBF kernel matrix from examples to landmarks as features and training logistic regression models on the MNIST dataset, reporting the results on the testing set. To make the differences between the methods more noticeable, only 10000 examples were used for training and 1000 for validation. Training was stopped with early stopping after 10 iterations without improvement.

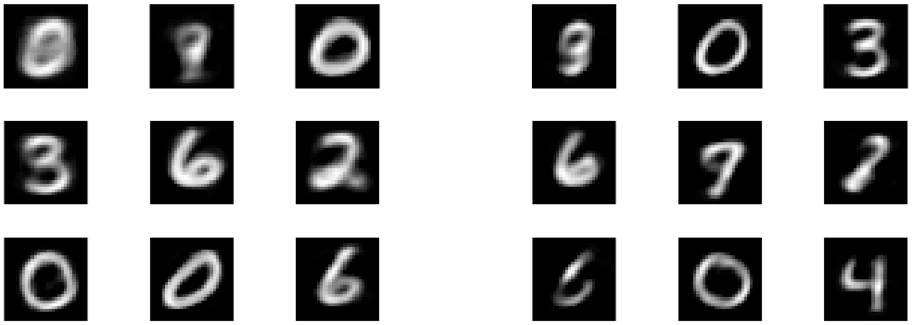


Figure 1.1: Visualization of the 9 landmarks from (a) the sequential landmarker we consider and (b) the ADMM variant with ℓ_1 sparsity for the full MNIST dataset.

1.3 Semi-Supervised Landmarking

Similar to support vector machines, construction of the full similarity matrix is the most demanding part for spectral clustering approaches. For this section, we will be following previous work on the subject that has utilized the centers of k-means clusters as landmarks or random landmarks and used the examples-to-landmarks similarity matrix for spectral clustering [Chen and Cai, 2011]. Following the paper, given the examples-to-landmarks

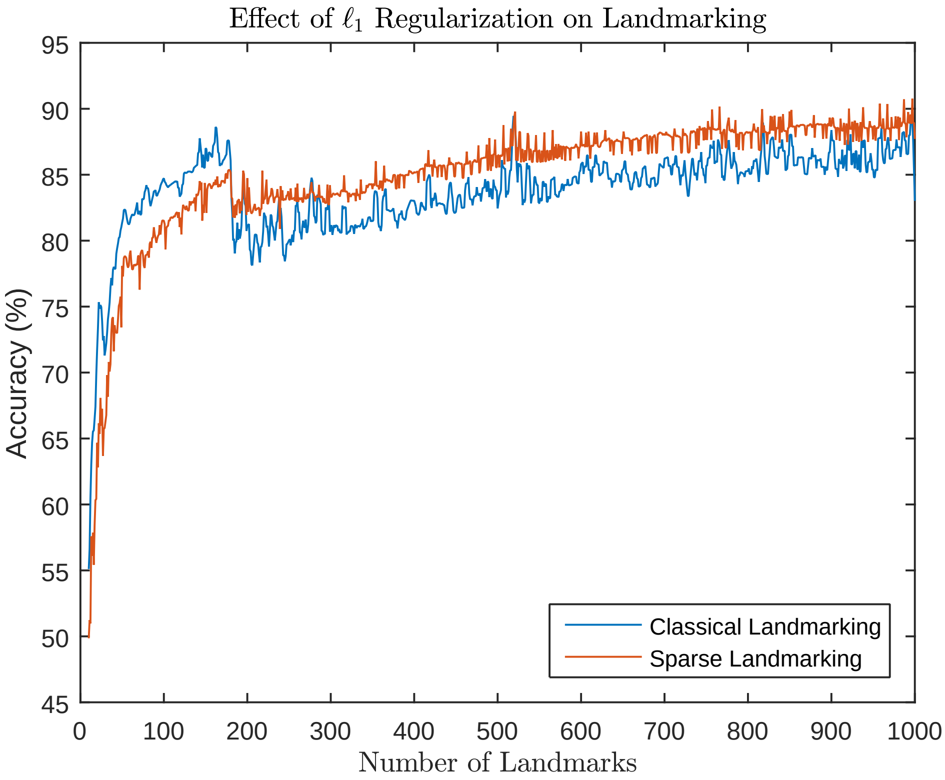


Figure 1.2: The difference between the classical landmarking algorithm and the sparse landmarking algorithm for the first 1000 landmarks by using RBF kernel matrix from examples to landmarks as features and training logistic regression models on the MNIST dataset. To make the differences between the methods more noticeable, only 10000 examples were used for training and 1000 for validation. Training was stopped with early stopping after 10 iterations without improvement.

kernel matrix $\Phi(\boldsymbol{X}, \boldsymbol{L}) \in \mathcal{R}^{M \times K}$ (where $\boldsymbol{X}$ is the matrix of examples and $\boldsymbol{L}$ is the landmarks matrix), one can calculate the eigenvectors of $\Phi(\boldsymbol{X}, \boldsymbol{L})^T \Phi(\boldsymbol{X}, \boldsymbol{L})$, then project $\Phi(\boldsymbol{X}, \boldsymbol{L})$ to that space and perform a final k-means to get the spectral clustering result. We will seek to use a similar idea to perform semi-supervised learning instead.

1.3.1 Semi-Supervised Learning on the Affinity

A direct way to extend this method is through the usage of $\Phi(\boldsymbol{X}, \boldsymbol{L})$ in addition with a one-hot label encoding $\boldsymbol{Y}_{train}$. For this subsection we choose to follow an old and famous methodology that makes use of the submatrices of the full examples-to-examples affinity matrix to get the desired result [Zhu *et al.*, 2003]. The equation used in the paper to calculate the result is

$$\hat{\boldsymbol{Y}}_{test} = \bar{\Phi}(\boldsymbol{X}_{test}, \boldsymbol{X}_{test})^{-1}\bar{\Phi}(\boldsymbol{X}_{test}, \boldsymbol{X}_{train})\boldsymbol{Y}_{train} \tag{1.8}$$

where $\bar{\Phi}$ refers to Φ after some normalization.

We seek the find a method for generating similar results using landmarks instead to significantly reduce the dimensionality required. We choose to use

$$\hat{\boldsymbol{Y}}_{test} = \bar{\Phi}(\boldsymbol{X}_{test}, \boldsymbol{L})^{\dagger}\bar{\Phi}(\boldsymbol{X}_{train}, \boldsymbol{L})^{T}\boldsymbol{Y}_{train} \tag{1.9}$$

to get similar results without actually computing the full affinity matrix.

We show a set of results as a function of landmarks using the classical landmarking algorithm from Chapter 1 against a simple k-means baseline for the examples on Figure 1.3. 1000 landmarks were used for each algorithm. As shown, the landmarking algorithm is able to beat k-means when the number of training examples is low; increasing the number of training examples quickly leads to both methods converging to the same accuracy.

1.4 Deep Landmarking

We will now look at landmarking as a step in a wide range of machine learning applications, from autoencoders to support vector machines and specifically deep RBF networks.

1.5 Autoencoders

Here, we consider the problem of learning an autoencoder with landmarking layers. An autoencoder is a model that is trained to mimic its input; the intermediate representation of the input, or *code*, generated by the autoencoder can then be used as features for an another method for classification, regression or unsupervised learning [Hinton and Salakhutdinov, 2006].

A once-popular extension of the autoencoder model is the stacked autoencoder, which is interesting due its ability to provide layerwise training; in a stacked autoencoder, a series of autoencoders are sequentially trained and every autoencoder uses the intermediate representation of the previous one as input/output [Bengio *et al.*, 2007]. More recently, variational autoencoders and their variants have been shown to give good results on a

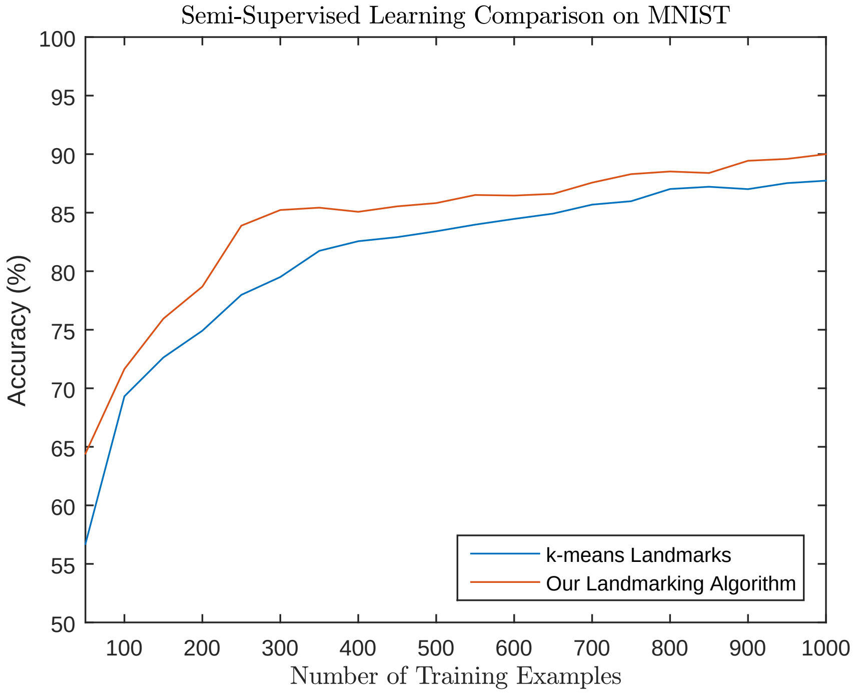


Figure 1.3: Semi-supervised classification results for k-means versus our approach (GML) with 1000 examples in the training set and 10000 in the test set. For these results, the classical landmarking algorithm from [Liang and Paisley, 2015] was utilized.

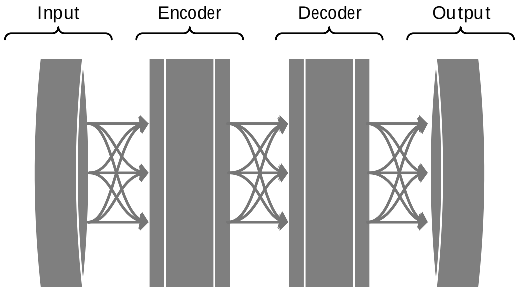


Figure 1.4: Model overview for a general autoencoder.

number of problems [Kingma and Welling, 2013].

Another interesting extension is the so-called denoising autoencoder, in which noise is added to the input of the autoencoder but not to the output so that the model will learn to denoise its input [Vincent *et al.*, 2008].

1.6 Landmarking Autoencoder

Let us consider an autoencoder in which the final layer of the encoder calculates the distance matrix between its inputs and a number of landmarks (wherein the landmarks are going to be the parameters of the layer). Then the decoder is forced to learn to reconstruct the data from this distance matrix. This very last layer of the encoder can then be seen as a landmarker with an Euclidean distance kernel (or log-RBF). One could also use the RBF kernel directly if desired.

In this model the landmarks can be learned sequentially, by freezing the landmarks that were previously obtained and restarting training to relearn the remaining parameters of the network as well as the new landmark. This type of model can be trained via stochastic gradient descent (SGD). Other, more modern stochastic optimization methods (like RMSProp [Tieleman and Hinton, 2012], Adadelta [Zeiler, 2012] and the recently-popular Adam [Kingma and Ba, 2014]) which allow for simultaneous learning of all landmarks could be utilized as well.

Whereas autoencoders are interesting approaches, by itself data reconstruction is uninteresting as end-to-end training is more desirable for supervised tasks. Therefore, we consider a number of landmarking support vector machine methods for classification in the upcoming sections.

1.7 Landmarking Support Vector Machine

Given an input vector $\boldsymbol{x} \in R^N$, a fully-connected (FC) hidden layer calculates

$$\boldsymbol{y} = g(\boldsymbol{x}\boldsymbol{W} + \boldsymbol{b}) \tag{1.10}$$

where $\boldsymbol{y} \in R^B$ are the output activations, g an activation function, $\mathbf{W} \in R^{N\times B}$ a weight matrix, and $\boldsymbol{b}$ a vector. Here, $\boldsymbol{W}$ and $\boldsymbol{b}$ are the parameters to be learned in a single layer, and in modern architectures $g(x) = max(0, x)$ (called the rectified linear unit, or ReLU) and its variants like parametric rectified linear units (PReLU) [He *et al.*, 2015b] are used as the activation. We can consider a kernel trick here to get a (potentially) more interesting set of features:

$$\boldsymbol{y} = g(\Phi(\boldsymbol{x}, \boldsymbol{W}) + \boldsymbol{b}) \tag{1.11}$$

which is a sort of representation that allows us to construct a support vector machine. For simplicity let's call this layer a Kernelized Fully-Connected (KFC) layer. We note that in literature this kind of layer is popularly referred to as a Radial Basis Function (RBF) network [Broomhead and Lowe, 1988], but the general framework in which we consider the architecture is going to be novel and we are ignoring the weighting of the output. Accordingly, for Φ, we choose to use

$$\Phi = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{W}\|_2^2}{2\sigma^2}} \tag{1.12}$$

in which we assume vectors broadcast to matrices.

For MNIST classification, we can consider a neural network with architecture:

$$Input \rightarrow Landmarking_{64} \rightarrow BatchNorm \rightarrow Softmax_{10} \tag{1.13}$$

where $Landmarking_{64}$ refers to a landmarking layer with 64 landmarks, $BatchNorm$ is batch normalization and $Softmax_{10}$ refers to a logistic regression layer (with 10 outputs corresponding to the 10 MNIST classes). We can train this architecture with multiclass hinge loss to get a proper landmarking support vector machine (LSVM).

We train a LSVM using the popular Keras library [Chollet, 2015]. First, we choose to train on the training set of MNIST by (1) setting $2\sigma^2$ to be equal to the variance of the input automatically or (2) making it a unique value per landmark. We initialize all $2\sigma^2$'s as 1000 and run the training for 100 epochs with the Adam optimizer and with only 16 neurons per landmarking layer.

We achieve 89.84% accuracy with the auto-sigma LSVM as compared to the 88.26% accuracy we received with a multi-sigma LSVM. Our result shows that some regularization is necessary to make sure that sigma values do not drift significantly from each other, hinting at the possibility of a Bayesian loss for enhanced results.

A limiting factor in RBF network training (as well as LSVM training) is the size of the input. To counteract this limit, we augment LSVM's with a FC layer after the input and before the KFC layer.

1.8 Multilayer Landmarking Support Vector Machine

In this section, we investigate whether it is possible to extend the method we consider into a multi-layer scheme. To perform this, we will need to use some recent regularization approaches like batch normalization [Ioffe and Szegedy, 2015] and dropout [Srivastava *et al.*, 2014]. These methods are used in virtually all modern deep learning approaches and allow for nearly-non-parameterized regularization, especially compared against the classical ℓ_1 and ℓ_2 regularization schemes that require extensive hyperparameter optimization.

1.9 Residual Networks

Residual networks are composed of blocks with so-called skip connections [He *et al.*, 2015a]. The general look of a residual block is shown on Figure 1.5.

We consider residual networks as a simple way to enhance the results and to make multilayer landmarking SVM's a viable training method. We compare residual blocks with several layers against multilayer LSVM's on the MNIST database, finding that multilayer LSVM's can outperform standard residual blocks. Specifically, we consider a variant of a classical residual architecture [Johnson *et al.*, 2016] for deep residual networks

$$Input \to FC_{64} \to \left[[FC_{64} \to BatchNorm \to FC_{64}]^5 \right] \to Softmax$$

where $[\dots]$ refers to a residual block in which the input and the final result are summed up (via elementwise sum) at the very end and powers refer to the number of repetitions of the structure within the parentheses Furthermore, FC_a refers to a fully connected (or dense) layer with a hidden units. The landmarking equivalent of this is given by

$$Input \to FC_{64} \to \left[[Landmarking_{64} \to BatchNorm \to Landmarking_{64}]^5 \right] \to Softmax$$

We find that the landmarking approach we consider is able to beat the equivalent residual version on the MNIST dataset, achieving 97.81% accuracy against the 96.56% accuracy baseline for the residual network when the dense or landmarking layers are succeeded by dropout layers with a dropout rate of 0.2. Note that we have purposefully considered only non-convolutional layers with a low number of neurons for the experiments so that there is a visible difference in the final results.

1.10 Future Work and Discussion

Immediate future work for the new residual approach we formulate would be to attempt to extend the approach for more general deep learning tasks (like convolutional architectures) and to perform extensive benchmarking; expensive hardware and large amounts of time are required to build the experiments for more advanced datasets like SVHN, CIFAR-10 and

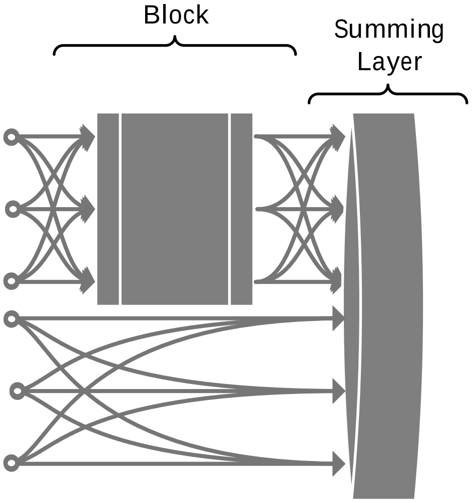


Figure 1.5: Overview of a general residual block. A standard residual block consists of a classical block of fully connected or convolutional layers with batch normalization and/or dropout as well as a summing layer that sums the input of that classical block with its output.

ImageNet. We are hoping to continue this line of research during our PhD, when we will have access to such hardware.

Kernelized layers are unique in that even a single layer is quite formidable, unlike classical fully connected layers that are only capable of straightforward subspace projection. Modern approaches for deep architectures that introduce skip connection or gates (like highway networks) to the layers are capable of alleviating the intractable overfitting issues encountered when attempting to generalize kernelized networks to multilayer schemes. Combination of dense residual network generalizations, like the one proposed in [Huang *et al.*, 2016], with kernelized layers is thus an additional promising line of research.

Part II

Learning Convolutional Mappings for Image Restoration

Chapter 2

Deep Pan-Sharpening

The pan-sharpening problem considers the estimation of a high-resolution multispectral image given a low-resolution multispectral (MS) image and a high-resolution panchromatic (PAN) image as inputs. The multi-input and super-resolution-like structure of the problem renders convolutional layers a natural approach to solving the problem. In this chapter, we propose a pan-sharpening neural network architecture that operates independently on all channels. This allows model training with large amounts of real world data. Through experiments on the QuickBird and IKONOS images, we demonstrate that our proposed method achieves state-of-the-art performance on a number of metrics.

2.1 Introduction

As physical limitations render the acquisition of high-resolution multispectral images difficult, for remote sensing applications a high-resolution panchromatic image and a low-resolution multispectral image is acquired via two separate sensors instead. A pan-sharpening method can then be used to fuse these two images to approximate a high-resolution multispectral image.

2.1.1 Relevant Work

A large number of different approaches have been offered for the pan-sharpening problem [Nikolakopoulos, 2008; Vivone *et al.*, 2015].

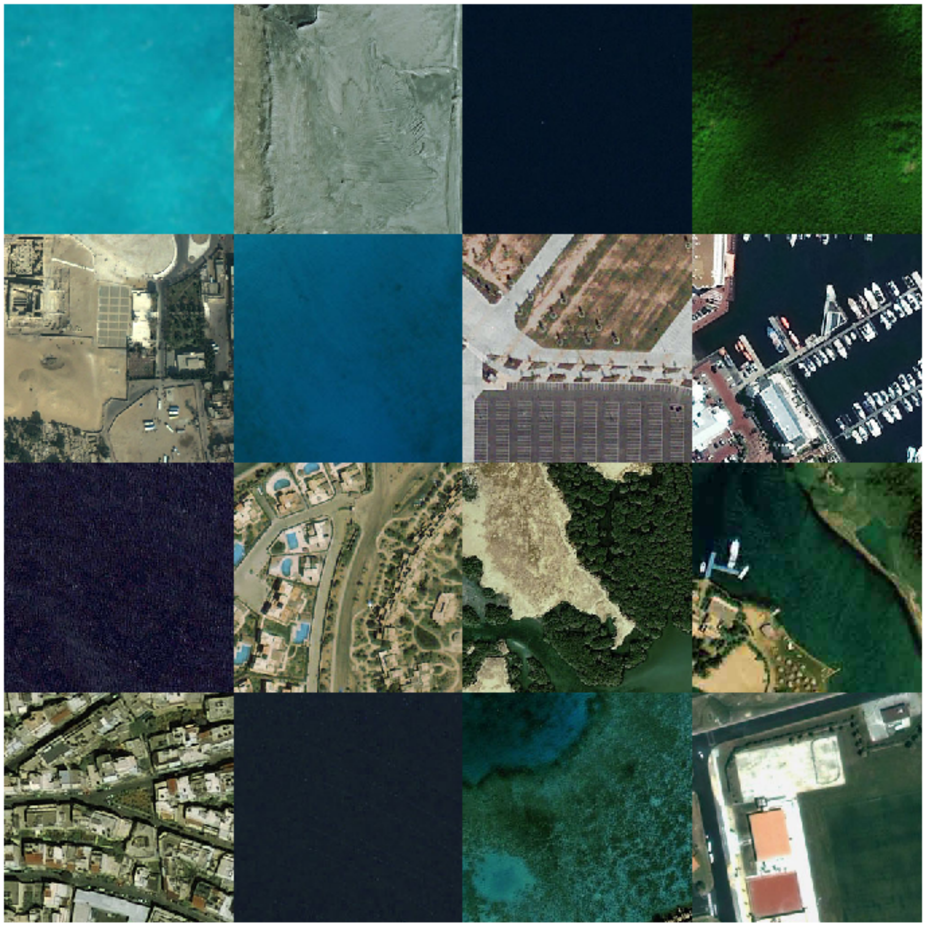


Figure 2.1: 16 high-resolution patches from the QuickBird dataset used in the experiments.

More recently, approaches based on sparse coding and Bayesian dictionary learning have been proposed [Li and Yang, 2011; Zhu and Bamler, 2013; Jiang *et al.*, 2015]. Following the sparse coding methodology and combining it with deep learning ideas, a recent work has even used a modified, deep sparse denoising autoencoder with greedy layerwise training and fine-tuning [Huang *et al.*, 2015b].

Deep learning methods with convolutional layers have shown promise on a large number of image transformation tasks. A variety of deep convolutional neural network architectures been proposed for super-resolution and achieve impressive results [Dong *et al.*, 2014b;

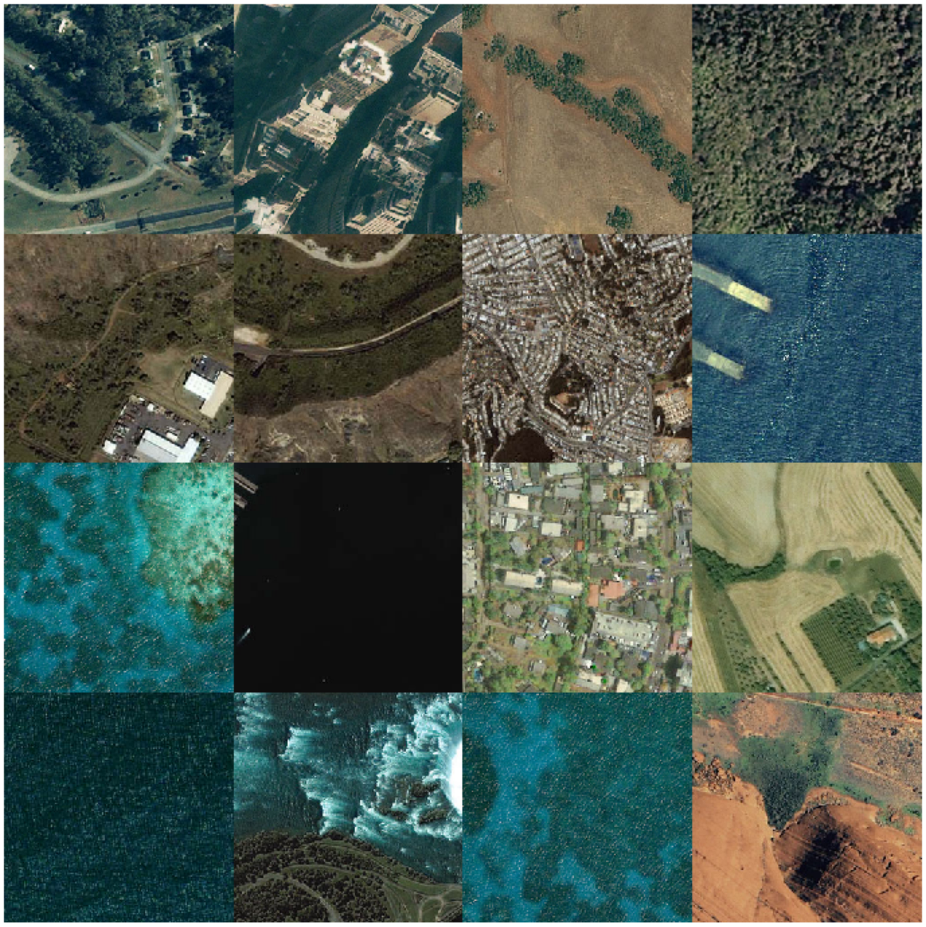


Figure 2.2: 16 high-resolution patches from the IKONOS dataset used in the experiments.

Wang *et al.*, 2015; Mao *et al.*, 2016; Huang *et al.*, 2015a; Ledig *et al.*, 2016]. Convolutional architectures have shown success on other difficult tasks like image colorization, style transfer and sketch inversion as well [Zhang *et al.*, 2016; Gatys *et al.*, 2015]. A convolutional neural network approach for the pan-sharpening problem has recently been proposed [Masi *et al.*, 2016], in which authors have improved the results through the addition of domain-specific features into the network.

The past convolutional network approach to pan-sharpening has utilized a simple network architecture (seemingly inspired by the simple yet effective super-resolution architecture in [Dong *et al.*, 2014a]) as satellite imagery data is relatively rare compared to the datasets for which deep learning is usually performed, which renders obvious deep learning approaches with many parameters difficult to train.

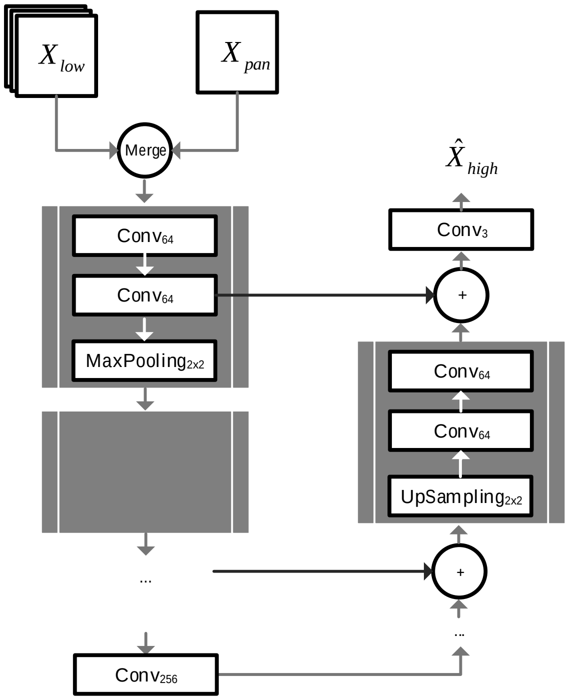


Figure 2.3: An overview of our multi-channel model, which could be used for single-channel outputs as well. We first get the low-resolution multispectral image to the same size as the high-resolution panchromatic image. We then add the panchromatic image as an another channel. In the left column, we gradually lower the resolution in the input, while upscaling the output in the second column. We use skip connections to make sure that the network can decide on the scale at which the filtering operations are to be performed. All convolutional layers are followed by a dropout layer with dropout fraction 0.25 and PReLU activations. Note that the blocks in the left column are different than the blocks in the right column. We use 5 blocks in each column in this chapter.

2.1.2 Our Contribution

In this chapter we introduce a pan-sharpening method that directly takes only a single channel of the low-resolution multispectral image and the panchromatic image as input. Our method first increases the resolution of the low-resolution multispectral input to match the resolution of the panchromatic image by upsampling with bicubic interpolation. The panchromatic image is then concatenated super-resolved tensor as a new channel, which is fed to the deep superresolution network and then eventually converted into an image of the expected output size. Finally, the pixel values are acquired by applying a sigmoid function at the output, assuming the target output values are in $[0, 1]$. We train this model with the Adam optimizer using a Keras-based implementation and show state-of-the-art performance in patches sampled from QuickBird and IKONOS images.

Table 2.1: **Baseline Network Configuration:** Input A is the low-resolution multispectral image and Input B is the high-resolution panchromatic image. $\text{Conv}^{Y}_{X\times X}$ refers to a convolutional layer with Y $X \times X$ dimensional filters, and we assume that the multichannel input resolution has been quadrupled with bicubic interpolation.

Input A: $4M \times 4N$ Naively Upscaled RGB Image	Input B: $4M \times 4N$ Grayscale Image
Merge Channels	
$\text{Conv}^{56}_{3\times3}$	
$\text{Conv}^{32}_{3\times3}$	
$\text{Conv}^{3}_{3\times3}$	
Sigmoid Activation	

2.2 Method

In this section, we propose a model trained with the mean-squared error (MSE) metric for solving the problem. A visual overview of the model we propose is given on Figure 2.3, and a baseline convolutional architecture, based on previous deep learning work on pan-sharpening, that we use for comparison is given on Table 2.1 [Masi *et al.*, 2016].

We use the Adam optimizer for training and a Dropout of 0.25 after the convolutional layers [Kingma and Ba, 2014; Srivastava *et al.*, 2014] We utilize the PReLU activation function for all convolutional layers [He *et al.*, 2015b]. We use 5 blocks for both the downsampling and the upsampling directions. For each model, during training, we use 20% of the data for validation and return the best model after running for 10 epochs or early stopping after no decrease in validation MSE after 5 epochs.

2.2.1 Channelwise Pan-Sharpening

In the previous section we have seen an approach to pan-sharpening that utilizes state-of-the-art deep learning methods. Whereas such models are powerful when given enough training data, they are not powerful when the size of the data is small due to the large number of parameters that require tuning.

A central problem about pan-sharpening, however, is the lack of training data. Every satellite has a number of non-RGB, near-infrared (NIR) channels that require superresolution as well. However, such satellite imagery is naturally extremely rare and thus is only available in small numbers. Due to the different number of channels in different satellites, the utilization of a standard convolutional neural network at the beginning of the deep neural network architecture is unfeasible as the weights of the first convolutional layer remain fixed. Here, we formulate a new approach that allows for general optimization schemes.

By building the same model as before, but only inputting one channel of the input at a time, we can train a network that performs channelwise pan-sharpening. In practice, this model could be the same as the normal multichannel pan-sharpening model we have introduced apart from the number of channels in the input and the output, which practically has the additional advantage of allowing the network to have less parameters at the more training-dependant earlier layers that naturally have a huge effect on the superresolution output. We will refer to this model as the Single Channel pan-sharpening model, and test it along with the multi-channel variant. For this specific model, to avoid comparison issues we choose to use the same architectural and training choices as above.

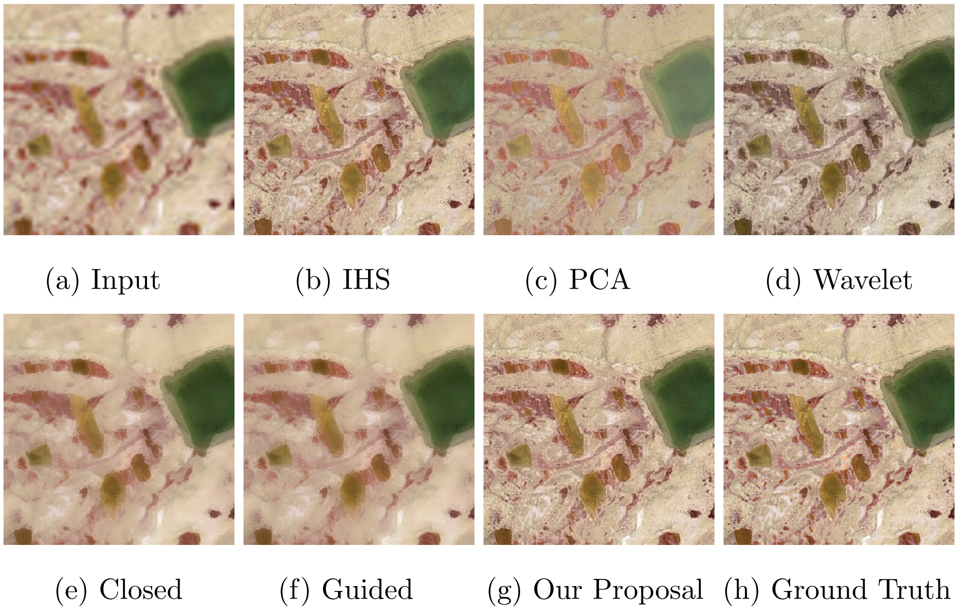
2.3 Experiments

2.3.1 Datasets

We run our experiments on publicly available QuickBird and IKONOS images. We extract 256×256 dimensional patches for each dataset. Following the established simulation methods, we perform 7×7 Gaussian blurring with $\sigma = 1$ followed by downsampling to get 64×64 dimensional low-resolution multispectral images and use grayscale versions of the high-resolution images as the panchromatic inputs.

2.3.1.1 QuickBird

We use 2800 patches for training and 340 for testing. We use disjoint image sets for training and testing. Examples from the dataset are shown on Figure 2.1.



(a) Input (b) IHS (c) PCA (d) Wavelet

(e) Closed (f) Guided (g) Our Proposal (h) Ground Truth

Figure 2.4: Examples of pan-sharpening results for the different methods and the method we propose. Due to the similarity between the approaches we consider, we only show results from the single-channel approach we describe.

2.3.1.2 IKONOS

We use 3600 patches for training and 390 for testing. We use disjoint image sets for training and testing. Examples from the dataset are shown on Figure 2.2.

2.3.2 Evaluation

We use RMSE, ERGAS, SAM and Q metrics for evaluation, and compare against the classical IHS, PCA, Wavelet, P+XS, Closed and Guided pan-sharpening methods. For our models we use the settings given above, specifically utilizing the MSE loss, and train each method by early stopping after no improvement for 10 epochs and loading the model with least validation error. Due to data restrictions, we only use the red, green and blue channels for the satellite data of both the QuickBird and KRONOS satellite imagery throughout our experiments.

We present the results on Table 2.2 for the QuickBird dataset and 2.3 for the IKONOS dataset. We also provide a visual example showing the results for the different methods on Figure 2.4. As shown, our approaches are good but not the best for every metric and dataset; this is to be expected, as our models have only taken RMSE into account during optimization. Practically, one must take into account the practical advantages of the single channel model as once could potentially train an all-purpose superresolution network to address every pan-sharpening problem regardless of the model specifications. This is more similar to the current and past pan-sharpening approaches that have been utilized to that purpose.

2.4 Discussion and Future Work

In this section, we have proposed and evaluated a number of deep convolutional network architecture with multiscale skip-connections to address the pan-sharpening problem. The architectures presented can be trained end-to-end via backpropagation. We have introduced a method that works on single channel imagery to be able to address the presence of near-infrared or other non-visible band channels for the same or different capture conditions under the constraint that the model was trained on similar channels beforehand.

Table 2.2: Results for the QuickBird dataset composed of 340 testing examples.

	RMSE	ERGAS	SAM	Q4
IHS	0.091	2.691	**2.049**	0.932
PCA	0.167	4.476	2.932	0.778
Wavelet	0.083	2.514	2.321	0.945
PxS	0.115	3.297	2.677	0.904
Closed	0.13	3.605	2.746	0.868
Guided	0.128	3.574	2.805	0.868
Convolutional Network Baseline	0.063	2.131	2.455	0.934
Our Proposal (Multichannel)	0.056	2.129	2.439	0.947
Our Proposal (Single Channel)	**0.052**	**2.126**	2.437	**0.949**

Table 2.3: Results for the IKONOS dataset composed of 390 testing examples.

	RMSE	ERGAS	SAM	Q4
IHS	0.098	2.698	2.456	0.94
PCA	0.175	4.484	2.94	0.786
Wavelet	0.091	2.522	2.328	0.953
PxS	0.123	3.305	2.685	0.912
Closed	0.137	3.613	2.754	0.876
Guided	0.136	3.582	2.812	0.875
Convolutional Network Baseline	0.07	**2.132**	2.442	0.957
Our Proposal (Multichannel)	0.064	2.136	2.447	0.955
Our Proposal (Single Channel)	**0.059**	2.134	**2.441**	**0.96**

In the future, we are hoping to improve the results further through the introduction of more regularization methods that promote realism such as the addition of texturing via adversarial generative networks, TV loss and VGG-16 loss as performed in some modern super-resolution approaches [Ledig *et al.*, 2016].

Chapter 3

Compressive Sensing Magnetic Resonance Imaging

A very similar problem is that of Compressive Sensing Magnetic Resonance Imaging (CS-MRI) [Lustig *et al.*, 2008]. In CS-MRI, we assume that we have a image that has been heavily subsampled in the Fourier domain and we seek to recover the original. This problem has been studied extensively over the past decade, with multiple approaches that utilize sparsity and/or dictionary learning constraints to achieve good reconstruction rates [Huang *et al.*, 2014].

For our approach, we make no assumptions regarding whether the sampling mask itself is known along with the image, though we utilize a fixed mask for simplicity and thus indirectly allow for the models we consider to learn it during the training phase. A further assumption is that the fully sampled Fourier domain signals we are seeking to recover have low sparsity, i.e. the number of nonzero entries are small. Of course, we make the usual assumption that we can minimize ℓ_1 norm instead.

3.1 Objective Function

The problem description itself tells us that we need to have an ℓ_1 error term related to the Fourier coefficients of the output. Usually, a TV-norm constraint is added in CS-MRI as well [Huang *et al.*, 2014]. As such, the objective function we seek to minimize becomes, for a single image,

$$J_{CS-MRI} = \left\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\right\|_2^2 + \alpha\left\|\hat{\boldsymbol{Y}}\right\|_{TV} + \beta\left(\left\|\left|\mathrm{Re}\left(\mathcal{F}\hat{\boldsymbol{Y}}\right)\right|\right\|_1 + \left\|\left|\mathrm{Im}\left(\mathcal{F}\hat{\boldsymbol{Y}}\right)\right|\right\|_1\right) \tag{3.1}$$

where α and β are constants, $\mathcal{F}$ denotes the 2D discrete Fourier transform, $\boldsymbol{Y}$ is the ground truth for an image and $\hat{\boldsymbol{Y}}$ is the estimate (in this case, the network output). We assume that our network takes in a corrupted image in the spatial domain, $\boldsymbol{X}$ as input and outputs $\hat{\boldsymbol{Y}}$. For multiple images, this objective effortlessly generalizes as a sum (or mean) over all images.

In the case when we are not assuming that a training set will not be available, the new objective function (that follows the approach from [Huang *et al.*, 2014]) becomes:

$$J_{CS-MRI-Single} = \alpha\left\|\hat{\boldsymbol{Y}}\right\|_{TV} + \beta\left\|\left(\left|\mathcal{F}\boldsymbol{Y}\right|\right) \circ \left(\mathcal{F}\hat{\boldsymbol{Y}} - \mathcal{F}\boldsymbol{Y}\right)\right\|_2^2 \tag{3.2}$$

which we will use for our single image CS-MRI experiments with a deep learning architecture that attempts to mimic a patch-based dictionary learning-based solution.

3.2 Model

A basic overview the model we choose to utilize is given in Figure 3.2, inspired by recent state of the art superresolution networks introduced [Ledig *et al.*, 2016]. Practically, we apply convolutional layers with ReLU activations (except for the final layer) and skip connections after every two convolutions. As is the norm with such models, filters in the convolutional layers are of size 3 × 3. As a balanced compromise between training speed and accuracy, we

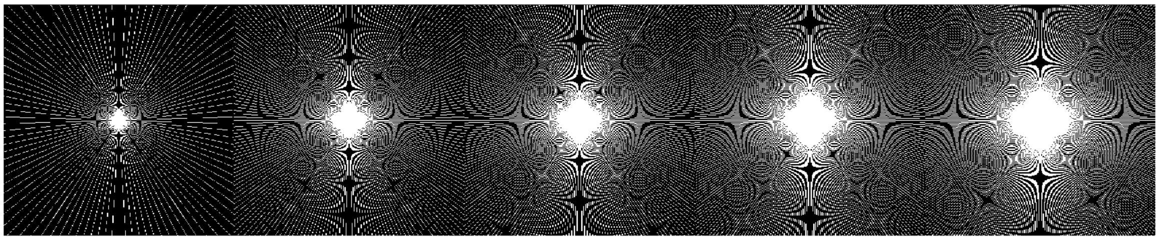


Figure 3.1: A number of radial CS-MRI sampling masks. In order, they correspond to sampling 10%, 20%, 25%, 30% or 35% of the original Fourier space signal. In this chapter, we use the middle mask, corresponding to sampling 25% of the Fourier domain representation.

consider 5 residual blocks for our models, and as shown in the figure utilize 64 filters per layer.

Due to numerical difficulties one would face with the last term, we find that we have to add extensive guards against NaN's. We use Adam for fast training [Kingma and Ba, 2014].

We also consider the single image CS-MRI problem, using Equation 3.2 as the objective function. For this model, we consider a very simple convolutional neural network with only two layers. The first layer contains 256 filters of size 6×6 (inspired by [Jiang *et al.*, 2015]) and the second layer contains a single filter of size 5×5; the first layer is succeeded by a ReLU activation and the second layer lacks a nonlinear activation function. We again use Adam for training of this network. We consider 512×512 dimensional single image inputs for the single image model.

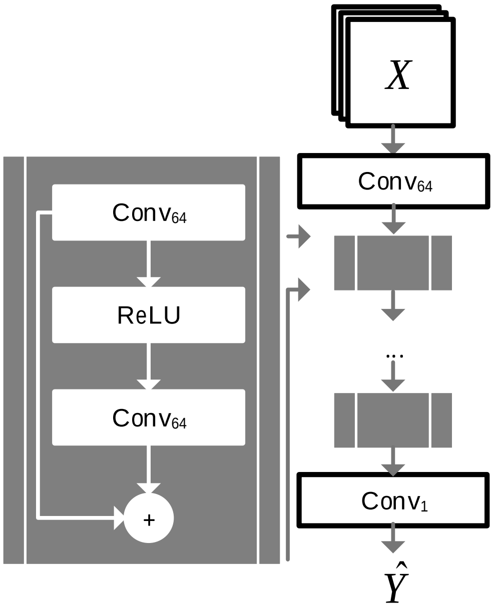


Figure 3.2: The image restoration model we utilize for CS-MRI. We consider convolutional layers to be without activations and have omitted the batch normalization after the convolutions. Whereas we have shown $\boldsymbol{X}$ as a multichannel image for the general case, for this section we assume that the input is a single-channel image.

For all approaches considered, we choose to use radial sampling masks for our approach as they are well known to be the ones that give the best results [Huang *et al.*, 2014]. We display images for 5 different masks with different sparsities on Figure 3.1. In this chapter, we use the mask in the middle, which corresponds to sampling 25% of the Fourier domain representation of the image.

3.3 Experiments

Due to the lack of a standard dataset to try CS-MRI algorithms on, we choose to use a subset of the publicly available OASIS dataset [Marcus *et al.*, 2007]. Specifically, we extract 6080 slices of cranial MRI from the first part of OASIS images. We show a montage of 16 images from the dataset we prepare on Figure 3.4.

We train our model using 75% of the data and validate on the 25%. We utilize five residual blocks for the network architecture and set $\alpha = \beta = 10^{-2}$ as higher values encourage sparsity further. We compare our results against an application of the sparse MRI algorithm with default parameters. We find that our algorithm is able to improve the 24.32dB input baseline (corrupted inputs) to 31.13dB over the testing set; in comparison, with the default parameters, the classical CS-MRI method is only able to achieve 25.19dB. With more training examples, deeper networks and heavy hyperparameter optimization, we believe we can improve the results significantly.

For the single image model, we provide results on two 512 × 512 images, shown on

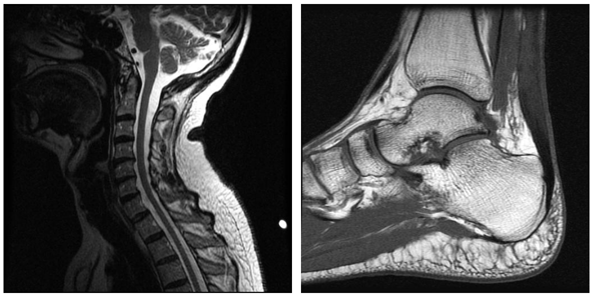


Figure 3.3: Two real-valued MRI images we choose to use for our single image model experiments.

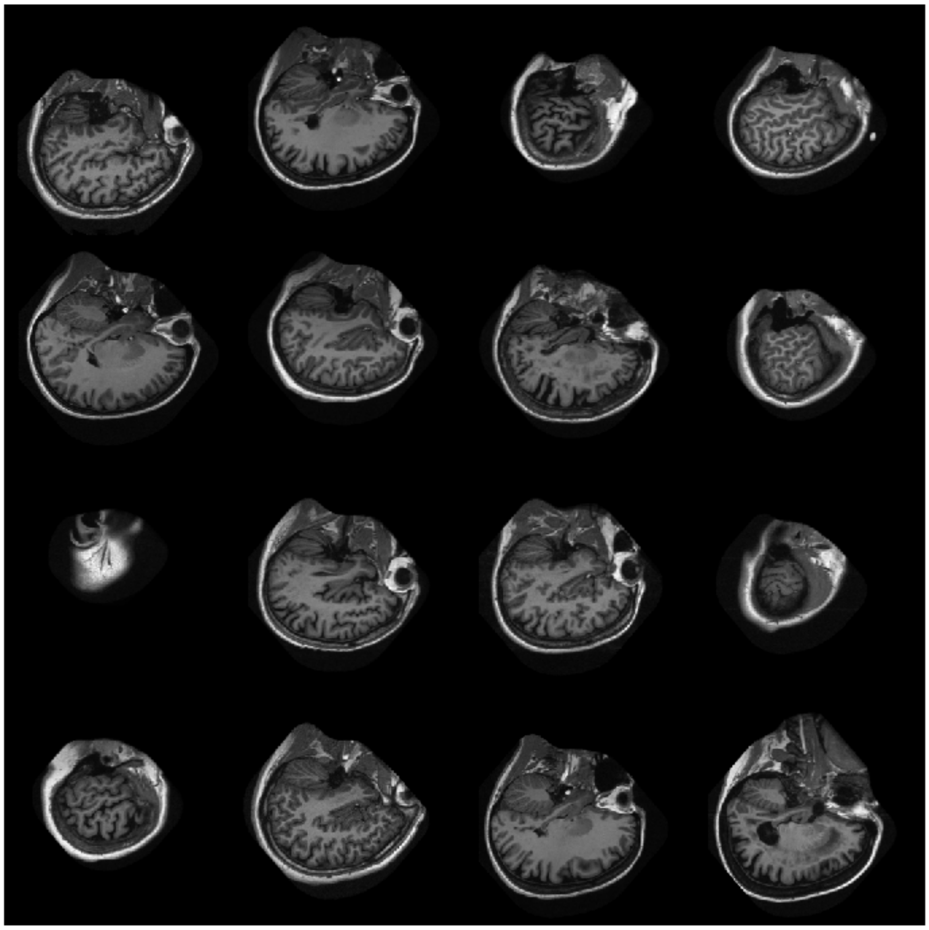


Figure 3.4: 16 images from the OASIS dataset used in the experiments.

Figure 3.3. For these experiments, we use the 10% sampling mask instead. For the first image, we find that our model improves the 18.45dB corrupted baseline to 21.12dB, while the default CS-MRI model achieves 19.51dB. For the second image, on the other hand, our model improves the 18.63dB corrupted baseline to 22.15dB, while the default CS-MRI model achieves 20.23dB.

3.4 Discussion and Future Work

Future work in this area should focus on finding ways to use the sampling mask within the network as a constraint along with the Fourier domain sparsity. In many ways, this problem is an interesting challenge for convolutional approaches as a convolution corresponds to an elementwise multiplication in the Fourier domain; as such, the sparse Fourier input leaves the activation function the main means of improvement for the network at hand unlike most other deep learning tasks in which the convolution filters are powerful and interpretable enough the describe the problem solution.

Furthermore, we believe that through the usage of more advanced, deeper architectures with more parameters and the inclusion of a more extensive hyperparameter search, the results shown can be improved significantly. Moreover, through the utilization of pre-trained neural networks like VGG-16 as feature extraction layers for the single-image approaches, we believe that significant strides in single-image CS-MRI could be attained. We are hoping to focus on such ideas in the future.

Part III

Lethargic Multilayer Dictionary Learning

Chapter 4

Lethargic Dictionary Learning

In this chapter we introduce a new non-greedy approach for coupled multilayer dictionary learning with an optionally supervised objective. We perform experiments on MNIST and CIFAR-10 databases, showing the efficacy of our method compared against classical online dictionary learning approaches.

4.1 Background

There has been a significant amount of interest in dictionary learning methods in recent years [Bengio *et al.*, 2013; Rubinstein *et al.*, 2013; Ravishankar *et al.*, 2015]. Similar methods, like autoencoders and adversarial generative networks, have similarly received significant attention [Vincent *et al.*, 2010; Bengio *et al.*, 2013; Goodfellow *et al.*, 2014; Radford *et al.*, 2015; Denton *et al.*, 2015].

In this part, we consider dictionary learning as a layer in general deep learning models, extending the idea of analysis dictionaries to perform end-to-end learning for arbitrary objectives to eliminate the mismatch between dictionary learning and supervised learning cost functions encountered when following greedy training schemes.

4.1.1 Our Contribution

We show that dictionary learning can be used as a deep neural network layer with a very specific regularizer. We assume that a dictionary learning layer can be fully parameterized by the analysis dictionary $\boldsymbol{D}^{-1}$ (assuming that $\boldsymbol{D}$ would in this case be the classical dictionary). Our coding step is entirely feed-forward and relies on the network being able to process its inputs (i.e. project them to a subspace easily seperable by the dictionary) with little error and without iterative optimization of an objective.

4.2 Method

Let us consider the dictionary learning objective. The task in general is to solve

$$\min_{\boldsymbol{D},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{D}\|_2^2 \tag{4.1}$$

For our solutions, we will assume that $\boldsymbol{D}^{-1}$ holds the parameters we want to keep. Observe that this parameterization directly turns our models into an *analysis dictionary learning* approach [Rubinstein *et al.*, 2013]. Then, taking the output of the layer $\boldsymbol{Z}$ as $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{D}^{-1}$, for a dictionary learning layer, the contribution to the overall objective function of the network can be given by

$$\min_{\boldsymbol{D}} \left\|\boldsymbol{X} - \boldsymbol{Z}\left(\boldsymbol{D}^{-1}\right)^{+}\right\|_2^2 + \lambda\|\boldsymbol{Z}\|_1 \tag{4.2}$$

In practice, this formulation is weaker than full dictionary learning as the sparse representation Z is not recovered via solving the basis pursuit problem on $\boldsymbol{Z}$ given $\boldsymbol{D}$. In order to solve this issue, we apply a nonlinear transform φ on $\boldsymbol{X}$ to get $\boldsymbol{Z}$ as follows:

$$\min_{\boldsymbol{D}} \left\|\boldsymbol{X} - \varphi\left(\boldsymbol{X}, \boldsymbol{D}^{-1}\right)\left(\boldsymbol{D}^{-1}\right)^{+}\right\|_2^2 + \lambda\left\|\varphi\left(\boldsymbol{X}, \boldsymbol{D}^{-1}\right)\right\|_1 \tag{4.3}$$

wherein for φ we can use the soft-shrinkage function with threshold θ (which could be learned as part of the network) as

$$\varphi\left(\boldsymbol{X}, \boldsymbol{D}^{-1}\right) = Shrinkage_{\theta}\left(\boldsymbol{X}\boldsymbol{D}^{-1}\right) \tag{4.4}$$

For the full LethargicNet model for classification and with categorical cross-entropy loss, the loss function is then given by

$$\begin{aligned} J_{Leth\arg icNet} = &- \sum_{u\in|\boldsymbol{X}|}\sum_{v\in C} \boldsymbol{y}_{u,v}\log(\hat{\boldsymbol{y}}_{u,v}) \\ &+ \sum_{i=1}^{I}\left[\alpha\left\|\boldsymbol{X}_i - \varphi\left(\boldsymbol{X}_i, \boldsymbol{D}_i^{-1}\right)\left(\boldsymbol{D}_i^{-1}\right)^{+}\right\|_2^2 + \lambda\left\|\varphi\left(\boldsymbol{X}_i, \boldsymbol{D}_i^{-1}\right)\right\|_1\right] \end{aligned} \tag{4.5}$$

where I denotes the number of hidden layers. As such the loss function has two regularization parameters, α and λ.

We can extend the approach we have proposed to unsupervised and semi-supervised problems without a significant alteration to the core of the method. For the unsupervised objective, we can simply choose to remove the cross-entropy loss. Then, the unsupervised objective function for the dictionary learning method we propose is

$$J_{Leth\arg icNet} = \sum_{j=1}^{J} \left[\alpha \left\| \boldsymbol{X}_i - \varphi\left(\boldsymbol{X}_i, \boldsymbol{D}_i^{-1}\right) \left(\boldsymbol{D}_i^{-1}\right)^{+} \right\|_2^2 + \lambda \left\| \varphi\left(\boldsymbol{X}_i, \boldsymbol{D}_i^{-1}\right) \right\|_1 \right] \tag{4.6}$$

whereas for the semi-supervised objective, we can utilize the well-known pseudolabel approach that only modifies the cross-entropy term for the unlabeled examples [Lee, 2013]. Simply put, if the per-class prediction vector for example i without labels is $\hat{\boldsymbol{y}}_i$, the label of that term in the cross-entropy objective is simply $argmax(\hat{\boldsymbol{y}}_i)$; so, the end result is practically the same as the one shown on Equation 4.5.

4.3 Experiments

We give results on two datasets, MNIST and CIFAR-10. We train our model first without an activation function, and with 3 layers followed by a softmax layer. We use the Adam optimizer for training and the Theano library for implementation [Kingma and Ba, 2014; Bastien *et al.*, 2012]. We compare our results against those achieved using greedy-trained stacked dictionaries via the well-known [Mairal *et al.*, 2009], which also allows for a mini-batch approach.

For all our experiments with the LethargicNet architecture, we have used 10000 examples in the training set for validation. We choose to report the testing set accuracy that corresponds to the highest accuracy in the validation set, using $\alpha = 5 \cdot 10^{-3}$ and $\lambda = 10^{-2}$. We report our results for the unsupervised learning coupled with k-nearest neighbors with $k = 1$, semi-supervised and supervised learning classification tasks. For the baseline results, classification is done with k-nearest neighbors with $k = 1$ as well. For the results to be more clear, we have used an architecture in which we train two consecutive dictionaries of 64 atoms each. Note that for the LethargicNet architectures 10000 examples from the training set are used for validation.

We provide results for both cases on all datasets on Table 4.1, using a batch size of 512 for both methods. In all of the experiments, we observe a speedup of at least a factor of 20 on a computer with an i7-6820k, 32GB RAM and a GTX 1080 graphics card.

4.4 Discussion and Future Work

In this chapter, we have introduced and evaluated a straightforward, end-to-end method for deep dictionary learning, testing the method under different training scenarios. We have found that the models we introduce are faster and perform better on a number of tasks, while being practical for implementing deep learning models with dictionary learning regularization as well as general dictionary learning problems.

Future work in this problem could proceed by introducing the same loss into convolutional architectures to enable patch-based dictionary learning, as used in modern approaches. Such a model would additionally benefit the single-image compressive sensing magnetic resonance problem we have considered in the preceding sections. In addition, by separating the code-generating network from the dictionary used in calculating the dictionary learning loss, better coders could be implemented in a straightforward manner without excessive loss of efficiency.

Table 4.1: Results for the LethargicNet method compared against classical online dictionary learning for MNIST and CIFAR-10 datasets. (64,64) refers to the model, wherein we train two consecutive dictionaries of 64 atoms each for ease of differentiation between the two methods. There is no trivial semi-supervised analogue for our method, so for the sake of fairness we leave the corresponding fields blank.

MNIST	Unsupervised	Semi-Supervised	Fully Supervised
LethargicNet (64,64)	**89.32%**	**92.12%**	**90.17%**
Baseline (64,64)	79.45%		80.01%
CIFAR-10	Unsupervised	Semi-Supervised	Fully Supervised
LethargicNet (64,64)	**26.20%**	**27.14%**	**40.21%**
Baseline (64,64)	23.17%		25.14%

Part IV

Microgrid Control

Chapter 5

The Microgrid Supply Control Problem

With increasing adoption of renewable energy, smart cities and related technologies, there is now an interest in efficient and fair power regulation methods for households when the customer demand in the grid exceeds the battery constraints at hand [Lopez, 2015; Soto *et al.*, 2012]. Of immense importance to the solution, or rather the formulation of the optimization problem, of this underspecified task is the tradeoff between efficiency and fairness .

Classical approaches adopt simple strategies that throttle down user power capacity depending on the amount of power available for allocation over a period, prioritizing more regular customers and attempting to not go below a certain minimum threshold specified to the customers beforehand (a schedule). Obviously, due to the jittery response of such strategies user happiness is affected negatively. Whereas electrical power is not a problem that many encounter in modern cities, Internet providers do utilize similar heuristics with weak guarantees that damage user happiness.

5.1 Introduction

In this chapter, we will consider a nonconvex optimization approach to this problem that's ≈ 20000 times as fast as the equivalent convex solution. We show how the model could be trained via classical stochastic gradient descent or for reinforcement learning of a policy network. Note that while the problem itself has some implementation-specific details regarding the physical implementation of the algorithms using available hardware, the machine learning viewpoint is quite simple as Arduino's and Raspberry Pi's provide enough computational power.

In the general *microgrid power allocation* problem, we are presented with a user demand at the beginning of every discrete time point (generally, updates are assumed to be hourly). Our task is to output the amount of power to supply per customer during the next hour. We assume that we have complete access to past supply and demand data. There are a number of soft and hard constraints regarding the supply output. Firstly, the ℓ_1 norm of the supply cannot exceed the battery capacity $P_{battery}$ or the available battery energy stored, $E_{battery}$ [Lopez, 2015]. To ensure those limits, all predictions are scaled down equally so that the ℓ_1 norm satisfies this upper limit.

Furthermore, every user has a schedule and an associated goodwill cost. The schedule acts as a soft lower bound for each user, discouraging a solution proposal from choosing to deliver an amount of power below that amount unless necessary. Finding the schedule itself, then, becomes an important question in itself. We want the method we propose to be capable of generating these schedules ahead of time as well.

5.2 Method and Architecture

Let us start by mathematically describing the problem at hand. The objective function we have, which tracks the revenue generated, can be written as

$$\begin{array}{ll} \underset{\theta}{\text{maximize}} & \sum_t \left(\min(\boldsymbol{d}_t, \boldsymbol{s}_t)p(t) + \boldsymbol{k} \circ \max\left(\boldsymbol{s}_t - \min(\boldsymbol{h}_t, \boldsymbol{d}_t)\right)\right)\mathbf{1}^T \\ \text{subject to} & \|\boldsymbol{s}_t\|_1 \leq P_{battery}, t = 1, \dots, t_{\max} \\ & \|\boldsymbol{s}_t\|_1 \leq E_{battery}(t), t = 1, \dots, t_{\max} \\ & 0 \leq \boldsymbol{s}_t \leq \boldsymbol{d}_t, t = 1, \dots, t_{\max} \end{array} \tag{5.1}$$

where $\circ$ is the element-wise product, $E_{battery}(t)$ is the energy left in the battery at time t, $\boldsymbol{d}_t$ and $\boldsymbol{s}_t$ are the demand at supply vectors at time t respectively, $\boldsymbol{k}$ is a vector of user priority coefficients (goodwill costs) and $\boldsymbol{h}_t$ is a vector of user schedules, $p(t)$ is the income which is a function of time (nights are more expensive for customers) and θ are the model parameters. $\boldsymbol{s}_t$ are calculated as the output of a deep neural network $\Theta_\theta(\boldsymbol{d}_t)$. Note that there is a temporal constraint in the constraints in terms of the battery capacity as $E_{battery}(t+1) = E_{battery}(t) + P_{solar}(t) - ||\boldsymbol{s}_t||_1$, wherein for the purposes of this thesis we assume that we do not have access to real $P_{solar}(t)$ values ahead of time.

For simplicity, we refer to the norm of the first term inside the sum in the objective as the *income*, whereas the negative of the norm of the second term becomes the amount *unmet*.

For the network itself, we utilize a variant of the networks we have considered in the previous sections, called Highway Networks [Srivastava *et al.*, 2015]. A feedforward variant of the highly popular Long Short Term Memory networks [Hochreiter and Schmidhuber, 1997], highway networks allow for two activation functions to exist simultaneously and thus grant the network the ability to automatically adjust the depth of the network during training. More specifically, we consider highway networks of form

$$\boldsymbol{Y} = f_{Highway}(\boldsymbol{X}) = g(\boldsymbol{X}\boldsymbol{W}_H + \boldsymbol{b}_T) \circ h(\boldsymbol{X}\boldsymbol{W}_T + \boldsymbol{b}_T) + \boldsymbol{X} \circ (1 - h(\boldsymbol{X}\boldsymbol{W}_T + \boldsymbol{b}_T)) \quad (5.2)$$

wherein $\boldsymbol{X}$ is the input matrix, $\boldsymbol{Y}$ is the output matrix, g and h are two activation functions, $\circ$ denotes the Hadamard (or elementwise) product and $\boldsymbol{W}_H$, $\boldsymbol{b}_H$, $\boldsymbol{W}_T$ and $\boldsymbol{b}_t$ are the parameters to be optimized within the layer.

A significant weakness of the highway architecture is that the very first layer needs have a sufficiently small dimensionality in order to avoid memory issues. As the input to the network could contain a window of arbitrary size, in practice we want the number of neurons in the initial hidden layer to be sufficient so that the architecture itself may remain unchanged when implemented on the field. As such, in practice, we implement a network

$$Input_{\#Users} \rightarrow FC_{512} \rightarrow (Highway_{512} \rightarrow)^{50} \rightarrow FC_{\#Users} \quad (5.3)$$

where $(\dots)^x$ refers to x repeats of the structure inside the parentheses and FC refers to a fully connected, or dense, layer so that the network does not overfit and remains fast to train even on simple hardware (such as a Raspberry Pi). We add dropout layers with a dropout rate of 0.5 after every layer as the low number of examples we have require heavy regularization for a network to be effective. Furthermore, we add Gaussian noise with $\sigma = 10^{-1}$ at the start of the training as an additional regularization method.

5.3 Evaluation

Obviously, the temporal component of the cost function given in Equation 5.1 renders offline training a difficult task. For offline training, we choose to use the worst-case scenario for the battery (assuming that the demand was always met) to generate the training examples while turning the constraints shown into soft constraints so that the network will be able to meet the constraints by gradient descent. Note that all of the constraints can be forced by simple calculations in real world implementations even if the network fails to meet them.

In addition, we choose to utilize a full reinforcement learning-based approach. For this, we utilize a policy network and take a similar approach to that taken in modern deep reinforcement learning approaches [Mnih *et al.*, 2013]. We train our network, simulating the real world results and considering the original objective function augmented with the size of the battery at time t as an additional additive term. We use a factor $\varepsilon = 0.05$ to determine the number of random choices our network is going to make, going against its predictions during training.

5.4 Experiments

We have been presented with hourly data for a three month period for 30 users by the Earth Institute at Columbia; due to the small size of the data, we use two months for training and the last month for testing. We have chosen to use this data for the rest of our experiments. This gives us 1440 examples for training and 720 examples for testing. We choose this split as we do not want to lose out on testing the capacity of the model introduced in adapting to long-range dependencies in the data.

As a balance between realism and idealism, we choose to use 97th percentile for each user as the schedule to employ on a per-user basis, and we calculate the priority coefficients k as the mean/std for the values of a particular user over the training set.

We show our results for two baselines and the models we consider in Figure 5.1. As shown, our neural network approach is able to outperform the baselines we consider. The first baseline we consider attempts to supply the demand at the previous time step, the second one simply uses the mean of the training, and the third supplies the 85th percentile of the demand for the user (85 was found through validation, and maximizes the revenue in a validation set composed of 20% of the training data). As shown, the reinforcement learning approach we consider is not as good as we would expect it to be, as expected since we lack enough data points to make the network capable of generalizing beyond the training.

We must note that our approach is very fast compared to a convex optimization-based solution for the problem that was considered previously, requiring 23ms (taken as the mean of 10000 runs) compared to the 60.45 second baseline reported in [Lopez, 2015] for a problem with a smaller microgrid.

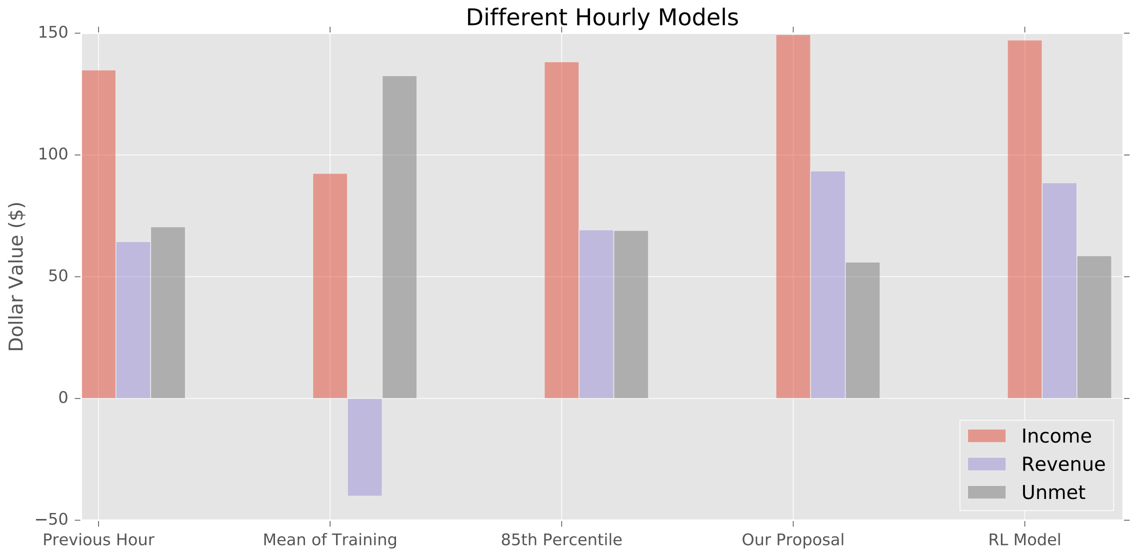


Figure 5.1: Results of the microgrid supply allocation problem we consider using 3-month data of 30 users, in which the last month is used for testing. As shown, our approach outperforms the simple baselines, while our reinforcement learning model shows promise but lags behind the offline approach due to, we believe, the lack of data and/or a good simulation method for better training.

5.5 Discussion and Future Work

In this chapter, we have looked at a method for applying non-convex optimization via deep learning methods to solve the constrained hourly supply allocation problem in an approximate manner. The approach we have outlined allows for online training, reinforcement learning and ease of adaptability to different constraints and objectives, especially regarding the goodwill cost.

Allowing for arbitrary differentiable and nonlinear goodwill regularizers, the approach we introduce could be used to solve allocation problems with arbitrary assumptions regarding customer benefits. This, in turn, makes it possible to have power allocation systems that are more merciful towards customers and thus capable of facilitating loyalty. One such approach would be the introduction of an ℓ_2-norm penalty to the objective. Many similar differentiable regularization techniques could be used to further granularize the notion of fairness and to maximize customer royalty by substituting user-specific punishments with negligible grid-wide cutoffs.

Part V

Conclusions

Chapter 6

Conclusion and Discussion

In this thesis, we have considered a number of exciting and potent approaches for problems ranging from landmarking to deep dictionary learning, pan-sharpening, compressive sensing magnetic resonance imaging and microgrid control, introducing and evaluating a number of exciting and novel approaches.

In the first part, we have described and evaluated a manifold landmarking approach trainable via stochastic gradient descent that allows for the consideration of structural regularization terms in the objective. Looking at the semi-supervised learning problems, we have shown that it is able to achieve comparable or better results than equivalent k-means based approaches on the MNIST database. We have also introduced an extension of this approach for general supervised and semi-supervised classification for structurally similar deep neural networks with self-modulating radial basis kernels. Whereas this lazy deep learning approach was not found to be as potent as the first approach we introduce in our experiments, we expect future ideas in deep learning training to render such approaches, which are more versatile due to their parameter-free nature, effective for a number of problems.

Switching to deep learning, we have looked at two interrelated image restoration problems, pan-sharpening and compressive sensing MRI. We have introduced a single channel convolutional autoencoding approach that directly sidesteps the data bottleneck problem in pan-sharpening that limits the effectiveness of the classical convolutional approaches to the problem.

Thirdly, we have introduced a method for multilayer dictionary learning and feedforward sparse coding by formulating the dictionary learning problem using a general deep learning layer architecture inspired by analysis dictionary learning. We have found the introduced method to be significantly faster to train than classical online dictionary learning approaches and capable of addressing supervised and semi-supervised classification problems more naturally.

Lastly, we have considered the problem of per-user power supply delivery on a microgrid powered by solar energy. Using real-world data obtained via The Earth Institute, we have looked at the problem of deciding the amount of power to supply to all each user for a certain period of time given their current power demand as well as past demand/supply data. We have approached the problem as one of demand-to-supply mapping, providing results for a policy network trained via regular propagation for worst-case control as well as classical deep reinforcement learning. Whereas we have not been able to make the reinforcement learning approach work more effectively than the worst-case model due to a large number of potential problems, we believe that it will be possible to render the approach viable through the addition of more real-world data, the introduction of more extensive regularization and hyperparameter optimization.

Part VI

Bibliography

Bibliography

[Bastien *et al.*, 2012] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[Bengio *et al.*, 2007] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[Broomhead and Lowe, 1988] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document, 1988.

[Candès *et al.*, 2011] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[Chen and Cai, 2011] Xinlei Chen and Deng Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, 2011.

[Chollet, 2015] François Chollet. Keras: Deep learning library for theano and tensorflow, 2015.

[Denton *et al.*, 2015] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[Dong *et al.*, 2014a] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *arXiv preprint arXiv:1501.00092*, 2014.

[Dong *et al.*, 2014b] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. *European Conference on Computer Vision*, pages 184–199, 2014.

[Gatys *et al.*, 2015] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[He *et al.*, 2015a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[He *et al.*, 2015b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[Hinton and Salakhutdinov, 2006] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2014] Yue Huang, John Paisley, Qin Lin, Xinghao Ding, Xueyang Fu, and Xiao-Ping Zhang. Bayesian nonparametric dictionary learning for compressed sensing mri. *IEEE Transactions on Image Processing*, 23(12):5007–5019, 2014.

[Huang *et al.*, 2015a] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015.

[Huang *et al.*, 2015b] Wei Huang, Liang Xiao, Zhihui Wei, Hongyi Liu, and Songze Tang. A new pan-sharpening method with deep neural networks. *IEEE Geoscience and Remote Sensing Letters*, 12(5):1037–1041, 2015.

[Huang *et al.*, 2016] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[Jiang *et al.*, 2015] Yiyong Jiang, Xinghao Ding, Delu Zeng, Yue Huang, and John Paisley. Pan-sharpening with a hyper-laplacian penalty. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 540–548, 2015.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Ledig *et al.*, 2016] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[Lee, 2013] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on Challenges in Representation Learning, ICML*, 3:2, 2013.

[Li and Yang, 2011] Shutao Li and Bin Yang. A new pan-sharpening method using a compressed sensing technique. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):738–746, 2011.

[Liang and Paisley, 2015] Dawen Liang and John Paisley. Landmarking manifolds with gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 466–474, 2015.

[Liu *et al.*, 2013] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[Lopez, 2015] Carlos Adrian Abad Lopez. Smart grid risk management. 2015.

[Lustig *et al.*, 2008] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, 25(2):72–82, 2008.

[Mairal *et al.*, 2009] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.

[Mao *et al.*, 2016] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016.

[Marcus *et al.*, 2007] Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.

[Masi *et al.*, 2016] Giuseppe Masi, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7):594, 2016.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Nikolakopoulos, 2008] Konstantinos G Nikolakopoulos. Comparison of nine fusion techniques for very high resolution data. *Photogrammetric Engineering & Remote Sensing*, 74(5):647–659, 2008.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Ravishankar *et al.*, 2015] Saiprasad Ravishankar, Bihan Wen, and Yoram Bresler. Online sparsifying transform learning—part i: Algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):625–636, 2015.

[Rubinstein *et al.*, 2013] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2013.

[Soto *et al.*, 2012] Daniel Soto, Edwin Adkins, Matt Basinger, Rajesh Menon, Sebastian Rodriguez-Sanchez, Natasha Owczarek, Ivan Willig, and Vijay Modi. A prepaid architecture for solar electricity delivery in rural areas. pages 130–138, 2012.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[Tieleman and Hinton, 2012] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[Vincent *et al.*, 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[Vivone *et al.*, 2015] Gemine Vivone, Luciano Alparone, Jocelyn Chanussot, Mauro Dalla Mura, Andrea Garzelli, Giorgio A Licciardi, Rocco Restaino, and Lucien Wald. A critical comparison among pansharpening algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2565–2586, 2015.

[Wang *et al.*, 2015] Zhangyang Wang, Yingzhen Yang, Zhaowen Wang, Shiyu Chang, Wei Han, Jianchao Yang, and Thomas Huang. Self-tuned deep super resolution. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2015.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zhang *et al.*, 2016] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. *arXiv preprint arXiv:1603.08511*, 2016.

[Zhu and Bamler, 2013] Xiao Xiang Zhu and Richard Bamler. A sparse image fusion algorithm with application to pan-sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2827–2836, 2013.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.