Investigation of different machine learning techniques for image reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

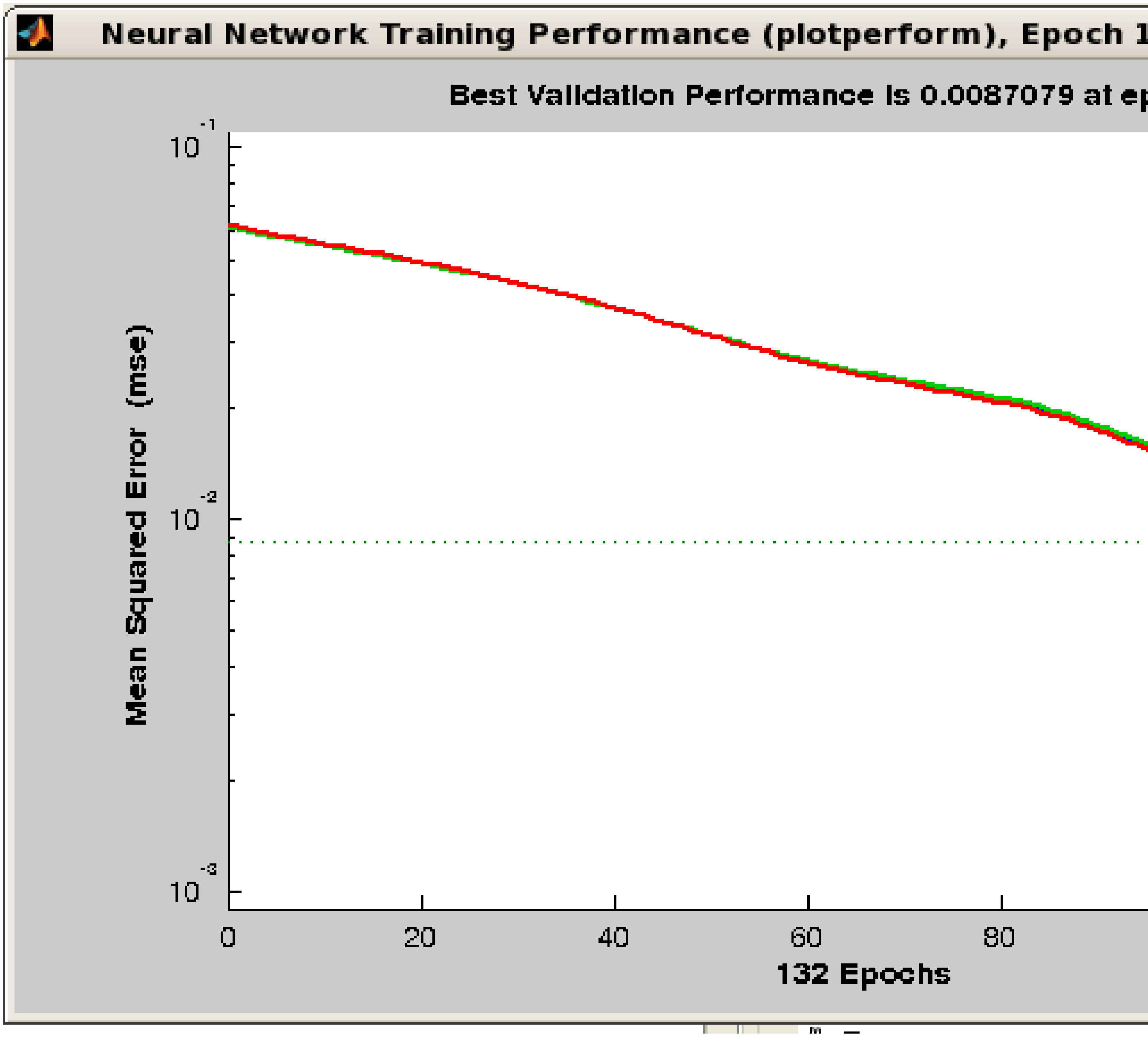
Imperfect sensors lead to problems of missing or lost data in a signal or image which must then be recovered or approximated. This filled in data must be coherent, noise free and as accurate as possible.Many signal processing techniques, such as compressed sensing exist for filling in missing data via an intelligent however, the majority of these techniques treat uniform noise in the image. Our work takes a different viewpoint: We examine the effectiveness of machine learning methods for filling in a missing region in an image. This missing region is fundamentally different from random or uniform noise. The missing piece of the image is assumed to be large enough that coherence and spatial locality is considered important. Unlike the problem of treating noise, we do not make any simplifying assumptions about the structure or format of the missing region. We treat this problem as a regression problem and explore three techniques for solving it. Artificial Neural Networks, Content Based Collaborative FIltering, and SVD Based Collaborative Filtering.

1 Artificial Neural Networks

Neural networks have been shown to be effective at a variety of image related tasks: Noise Reduction, Image recognition and clustering to name a few. Additionally, the Backpropagation algorithm provides a fairly straightforward way to train a network in a classification task. Given this line of thought, we question whether neural networks are suitable enough to predict pixel values in an image. The task needs to extend a trained classifier to a trained regressor given a choice of the neuron functions. However this approach is agnostic of the properties of the image. No information about spatial locality is taken into account nor are any spectral properties or rank of the image (if considered as a matrix). Therefore in our training data generation, we encode the cartesian coordinates of each pixel in an example along with the intensity values. Prior trials have suggested the need to preprocess the image so as to improve training times and convergence rates. We process by first scaling elements in a given column by their column-wise mean. The image is then converted to grayscale (8bit). We also learn input weights for a variety of network configurations, varying the following parameters: neuron type, count, and learning function to see if a particular attribute of the neural network allows for better prediction.

We primarily vary the type of neurons in each layer rather than the number of layers. The literature suggests that a network with a highly complex structure of hidden layers does not yield increased accuracy. Instead we vary the use of linear and logistic (sigmoid) units with our network structure. We consider four cases: 1) A network of 2 hidden layers each consisting of sigmoid units; 2) A network of 1 hidden layer each consisting of linear units; 3) A network as described before but with the first layer sigmoid and the second linear; 4) The reverse: first a hidden layer of linear units and then a layer of sigmoid units. We fix the evaluation function as the Mean Squared Error (MSE), and the training function as gradient descent with momentum and an adaptive learning rate. Finding a network of Type 1 (all sigmoid) to be the best performer, we train the network on camo.jpg. Training statistics are below for the network of all sigmoid units is below.

Varying the learning function on the camo image produced varying results: In some cases gradient descent alone proved to be non-convergent, while resilient backpropagation took the longest to converge. As each prediction from the neural network is able to capture some nonlinearity but is a local optima



Training statistics for a 2 layer hidden neural net (most optimal)

Predicted result of camo.jpg using resilient backpropagation (Rprop)

Predicted result of camo.jpg using gradient descent with adaptive LR and momentum

2 Content based CF

In content based CF, we assume that the each row of the image has n features which describe that row. And we assume that each column has n weights associated with it. The collaborative filtering method then uses an MLE to minimize the cost function .

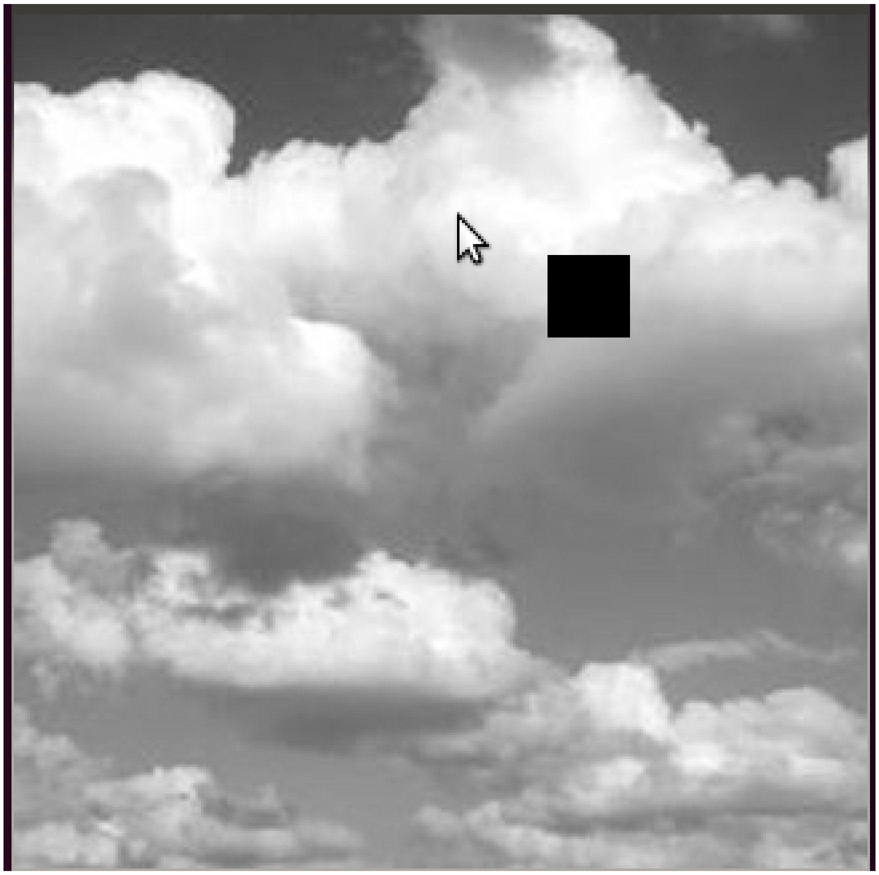
We minimize this cost function using various forms of gradient descent. While the results obtained using this method were satisfactory, the method converged far too slowly and hence we looked for an alternative method of performing collaborative filtering, namely SVD based collaborative filtering.

3 SVD based CF

The Singular Value Decomposition (SVD) of a matrix is a well known matrix factorization method which takes a matrix A and decomposes it in the following manner. $A = USV^T$ U and V are orthogonal matrices and S is diagonal matrix of singular values. The singular values of a matrix are nonnegative real numbers. Computing the SVD of a matrix is a popular method of making a low rank approximation of it. The low rank approximations of matrices are widely used in machine learning and image processing applications. However, in our problem, we are missing data in a part of the image. We cannot take the SVD of a matrix which is missing some values. To remedy this, we calculate the column averages of the columns which contain the missing value using available values in that column. And then we set every missing element in that particular column to the column average of that column. This can be seen as a rough approximation to the missing portion. This method of tentatively filling in the pixel values with column averages is quite popular in recommender systems. Once this is done, we take the SVD of this matrix. We then truncate this SVD upto k singular values. The predicted missing values are then given by;

The code for the SVD based collaborative filter was implemented using C++ and an image processing library called CoolImage. This is a very lightweight library with basic image processing functionality. The results of the experiments on an image of clouds is as follows;

Clouds.jpg



4 Conclusion

As we can observe from the images obtained from collaborative filtering, the edges are not preserved in the predicted image. As the human eye is very sensitive to the spatial locality of the entire image, not having edges preserved affects the visually observed effect of the image. Theoretically, the SVD gives the best low rank approximation to the matrix according to the 2-norm and the Frobenius norm. Even though technically, the predicted image is as close to the original as possible according to the 2-norm and the Frobenius norm, the missing edges (especially in Texture.jpg) make it plain to a human observed where the data is missing and affects the overall 'visual quality' of the image. This is because the norm of a matrix is a sum over all the entries in the matrix. This can be thought of as a 'cumulative' property or a global property of the matrix. However, an image has very strong local properties which do not exhibit themselves outside a very localised neighbourhood. So taking the cumulative sum destroys the local properties of the image which in turn affect the visual quality of the predicted image. We also found that because of this very reason, the SVD collaborative filter performs very poorly on complicated images with lots of features. It only performs relatively well on simple uniform images (like clouds and simple textures) or images which have repeating patterns in them. Researchers at University of Illinois in Chicago have reported similar results for image reconstruction from collaborative filtering. Thus we can conclude that collaborative filtering is not a very good method for performing image reconstruction.

The neural network as a pixel value predictor does not do overly well under the training functions and network configurations we've used. Primarily, it is not able to take spatial locality into account and has no knowledge of properties of the image. In the best case regressors, neural nets is able to match the average intensity of the region that it recovers but is unable to detect edges or sudden changes in intensity. It is at best a good local approximator to some extend but cannot recover the more important properties of the image without the data being present. The range of images that our networks can train and test on is very limited. For our work, we preprocessed the image in addition to training and testing on grayscale images.

References

[1] Vozalis M. & Markos, A Evaluation of standard SVD based techniques for Collaborative filtering. In G. Tesauro, D. S. Touretzky

[2] Shi, X. & Yu, P *Limitations of Matrix Completion via Trace Norm Minimization.*

[3] Hasoon, H.N., Yousif, J.H, Hasson, N.N., Ramli, A.R., (2011), Image Enhancement Using Nonlinear Filtering Based Neural Network

[4] Cands, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. Foundations of Computational mathematics , 9(6), 717-772.

[5] Keshavan, R. H., Montanari, A., & Oh, S. (2010). Matrix completion from a few entries. Information Theory, IEEE Transactions on, 56(6), 2980-2998.