Department of Electrical and Computer
Engineering Technical Reports

Department of Electrical and Computer
Engineering

11-24-2018

Ladder Networks for Semi-Supervised Hyperspectral Image Classification

okan ersoy
ersoy@purdue.edu

Julian Buchel
ETH Zurich, jubueche@ethz.ch

Follow this and additional works at: https://docs.lib.purdue.edu/ecetr

Ladder Networks for Semi-Supervised Hyperspectral Image Classification

Julian Büchel[1,2] and Okan Ersoy[1]

[1]Purdue University, 610 Purdue Mall, West Lafayette, IN, USA
[2]ETH Zürich, Rämistrasse 101, Zürich, CH
jubueche@ethz.ch

Abstract

We used the Ladder Network [Rasmus et al. (2015)] to perform Hyperspectral Image Classification in a semi-supervised setting. The Ladder Network distinguishes itself from other semi-supervised methods by jointly optimizing a supervised and unsupervised cost. In many settings this has proven to be more successful than other semi-supervised techniques, such as pretraining using unlabeled data. We furthermore show that the convolutional Ladder Network outperforms most of the current techniques used in hyperspectral image classification and achieves new state-of-the-art performance on the Pavia University dataset given only 5 labeled data points per class.

Introduction

Over the last years, semi-supervised learning has increasingly gained interest in the Machine Learning community. Having only a few labeled data points has become the reality in many real-world applications. This is due to the fact that labeling data points is tedious and sometimes costly work.

This problem has also gotten attention in the Hyperspectral Imaging community and various techniques have been proposed in order to solve it [Wang et al. (2016), Wu et al. (2016), Zhang and Crawford (2016), Jamshidpour et al. (2016), Alhichri et al. (2015), Wu and Prasad (2018), Xu et al. (2018), Fang et al. (2018b), Liu et al. (2017b), Andekah et al. (2017), Zhan et al. (2018)]. A lot of those techniques make use of the unlabeled training data by pretraining in an unsupervised setting and afterwards performing supervised learning on the labeled examples. One example is the use of a denoising Auto-Encoder[Vincent et al. (2010), Ma et al. (2016a)] as the unsupervised learning technique and a simple convolutional neural network as the supervised learning mechanism. The Auto-Encoders goal is to reconstruct the original input after various transformations into a lower dimensional space. The learned representation can then be used for supervised training. The question arises whether the learned representation is valuable for the supervised learning task or not. Often

it is only partially useful. The Auto-Encoder retains information useful for reconstruction and not necessarily useful for classification. The Ladder Network (LN) resolves this problem by **jointly** optimizing a reconstruction and classification cost. It should be noted that Liu et al. (2017a) have used the Ladder Network in hyperspectral image classification, but we achieved even better accuracy on 5 times less labeled training examples per class. Our paper further investigates the effect of different components in the Ladder Network and presents a comparison to existing state-of-the-art algorithms. We additionally show that the Ladder Network achieves the best performance given only 5 labeled examples per class.

In the following section, we will provide an overview of how the Ladder Network works and what its key components are. In the experimental section, we compare two variants of the Ladder Network: One based on simple, fully-connected layers and the other one based on convolutional layers. We show that for any number of labeled training examples, the convolutional LN outperforms the fully connected LN. We then compare the two networks with other competitive techniques on two widely used datasets in the Hyperspectral Imaging community and show that Ladder Networks are superior to most of the current models.

At the end, we conclude our findings and describe further possibilities on how to gain higher accuracies using this approach.

The Ladder Network

The Ladder Network has a complex architecture, which can be derived by systematically adding more features to simpler architectures. For this reason, we will start with a simple Auto-Encoder and gradually build upon it to derive the Ladder Network.

Auto-Encoders (AE) consist of two parts: The *Encoder* and *Decoder*. The encoder maps input x, using a function

$f_\theta(x)$, to an output y, which has a lower dimension. The decoder on the other hand maps the input y to the output z using another transformation $g_{\theta'}(y)$. The output z is the reconstructed input x. The parameters θ and θ' of the transformations are learned via backpropagation by minimizing the reconstruction cost between x and z.

There are two fundamental issues with traditional AE's: First, the one-layer architecture may be too simple to learn complex features and second the network may just learn some kind of identity mapping.

The second problem can be resolved by simply reducing the dimensionality or by adding random noise. Adding noise has the effect that parts of the input are perturbed and the AE therefore needs to extract relevant information from the input by denoising it. Hence the name Denoising Auto-Encoder (**DAE**). In order to overcome the first problem, it is natural to think of simply adding more layers: Stacked Denoising Autoencoders (**SDAE**) are single **DAEs**, which try to extract features from the encoded input of a previous **DAE**. Figure 1a illustrates the unsupervised training process. We can then use the encoded input as a feature for any given classification task as shown in Figure 1b.
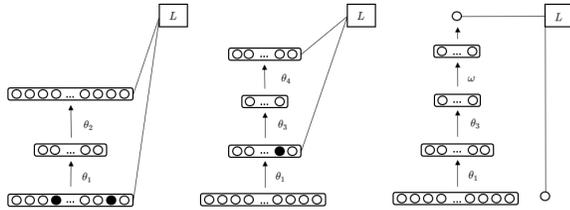

Figure 1: Figure 1a (left, middle) shows the unsupervised training process of a **SDAE** of size 2. Figure 1b (right) shows the network of stacked encoders with an additional layer performing a classification task. All the weights are subject to change during training of the complete network (right). Black dots symbolize data points that have been zeroed out.

We have provided a GoogleCollaborator notebook with implementations of the various Auto-Encoders, which can be found in the section 'Code'.

The **Ladder Network** is essentially a stacked denoising Auto-Encoder with lateral skip connections. It combines the cost of a supervised task with the reconstruction cost of intermediate representations of the input. Figure 2[Rasmus et al. (2015)] illustrates the architecture.

Initially, the network performs a noisy pass through the encoder. Input x is corrupted using Gaussian noise with zero mean and a user defined standard deviation. The intermediate layers perform linear or non-linear transformations of the input followed by batch normalization. Precisely, x initially
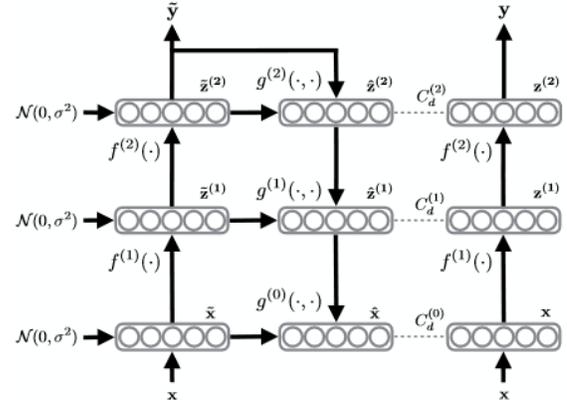

Figure 2: During training, the input x is corrupted and transformed in the encoder (left). The supervised training is done using the corrupted representation of x. The corrupted input is then reconstructed (middle) using the decoder. As targets for the reconstruction, the clean encoder representations are used (right).

gets transformed to

$$\tilde{h}^{(0)} = x + \mathbf{noise}$$

and for all layers l,

$$\tilde{z}^{(l)} = \mathbf{batchnorm}(W^{(l)}\tilde{h}^{(l-1)}) + \mathbf{noise}$$

Using any activation (for example ReLU) we obtain

$$\tilde{h}^{(l)} = \mathbf{activation}(\gamma^{(l)} * (\tilde{z}^{(l)} + \beta^{(l)}))$$

and repeat the process. Afterwards, the most abstract representation of the input, namely $\tilde{h}^{(L)}$, is passed through the decoder. Initially, we have

$$u^{(L)} = \mathbf{batchnorm}(h^{(L)})$$

And subsequently for all layers l

$$u^{(l)} = \mathbf{batchnorm}(V^{(l+1)}\hat{z}^{(l+1)})$$

where

$$\hat{z}^{(l+1)} = g(\tilde{z}^{(l+1)}, u^{(l+1)})$$

Here, $g(.)$ refers to the pointwise function, applied as the lateral skip connection. As shown in Figure 2 (right), we now pass the clean x through the encoder to obtain the intermediate representations denoted $z^{(l)}$. The reconstruction cost is therefore the sum over all differences between original representation and reconstruction

$$\mathbf{C}_{Recon} = \sum_l \lambda_l ||z^{(l)} - \hat{z}^{(l)}||^2$$

where λ_l is a user defined multiplier.

The supervised cost is defined as

$$\mathbf{C}_{Super} = -\frac{1}{N} \sum_i log(P(\tilde{y}_i = t_i | x_i))$$

where $\tilde{y}$ is the resulting prediction using the corrupted encoder pass. For prediction, we are using the clean encoder. The motivation behind using the corrupted encoder for the supervised training is the same as for the denoising Auto-Encoder: We are learning a denoising function, which essentially serves as a feature extractor.
The total cost is

$$\mathbf{C}_{Total} = \mathbf{C}_{Recon} + \mathbf{C}_{Super}$$

Several papers [Cao et al. (2016), Zhang and Li (2016), Zhong et al. (2017)] have exploited the spatial information carried in hyperspectral images by using Convolutional Neural Networks. Unfortunately, these networks lack the ability to generalize well given only few labeled data points, as we show in our experiments. By substituting ordinary, fully-connected layers in the Ladder Network by convolutional ones (Figure 3), we combine the benefits of CNN's and LN's.
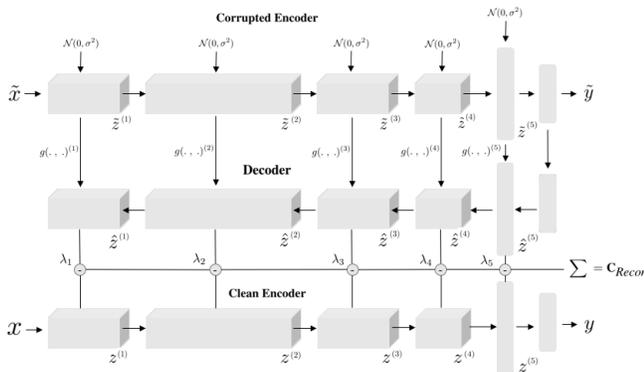

Figure 3: The convolutional version of the Ladder Network.

Experiments

In this section, we compare the convolutional and fully-connected Ladder Network to various other methods on a widely used dataset in the Hyperspectral Imaging community. We show that the convolutional Ladder Network is able to achieve highly competitive results, given all labeled data points, but also is superior to most other techniques in the semi-supervised setting. The convolutional LN was able to achieve new state-of-the-art performance given 5 labeled points per class. We further analyze the effect of the denoising cost given different amounts of labeled data, the influence of the injected noise and the effect of dimensionality reduction using Principle Component Analysis

(PCA) as a preprocessing step. All the experiments were conducted on GoogleCollaborator and we have made the code available (see section 'Code').

The dataset we used is a hyperspectral image of the Pavia University in Italy. This image was recorded using the RO-SIS sensor, and after the removal of 12 bands due to noise and water absorption comprises 103 spectral bands, with a range of 0.41 to 0.82 μm. The 610x340 image and its corresponding groundtruth can be seen in Figure 4. Due to imbalance in the dataset, we either down- or upsampled the minority labels, depending on the method. In all experiments, we used 25% of the points for testing and the remaining points for training. The code for the convolutional Ladder Network was written in TensorFlow[Abadi et al. (2015)] and we used the Adam optimizer[Kingma and Ba (2014)] to train the network.
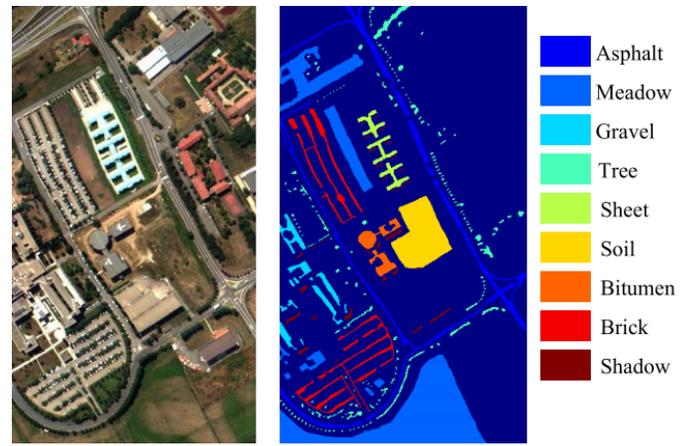

Figure 4: The false color image (left) and the groundtruth image (right) of the Pavia University with the corresponding 9 classes.

Figure 5 shows the accuracy of the convolutional and fully-connected Ladder Network given different numbers of labeled training points. As can be seen, the fully-connected Ladder Network is constantly being outperformed by the convolutional version. For the fully-connected LN, we used a learning rate of 0.005, a batch size of 100, 4 intermediate layers with sizes 300-200-100-100 and respective denoising costs of 100-10-1-0.1-0.1. For the convolutional LN, we used a learning rate of 0.01, a batch size of 100 and the architecture was conv-90 - conv30 - conv15 - fc30, where the number after conv denotes the filter size. The window size achieving the best accuracy was 7. After each convolutional layer we used the ReLU activation function. The denoising cost for all layers that achieved the best accuracy was 0-0-0-0-0-0.42.

Table 1 illustrates that the convolutional LN achieved new state-of-the-art performance given only 5 labeled datapoints.
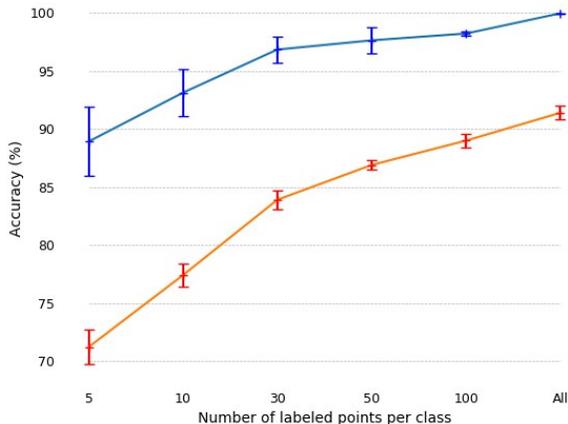
Figure 5: Accuracy versus different numbers of labeled examples per class using the convolutional LN (blue) and the fully-connected LN (red).

Algorithm	Number of labeled points	
	5	10
CNN	55.40 ± 3.89	69.02 ± 2.26
CDL-MD-L	72.85	82.61 ± 2.95
Co-DC-CNN	83.47 ± 3.01	94.99 ± 1.49
PNGrow	88.11 ± 2.87	93.85 ± 2.23
TT-AL-MSH-MKE	79.04 ± 3.95	86.00 ± 3.04
S^2CoTraC	50.76 ± 1.68	80.75 ± 0.35
Co-DC-Res	86.69 ± 2.94	$\mathbf{96.16 \pm 1.05}$
FC-Ladder	71.2 ± 1.5	77.4 ± 1.0
CNN-Ladder	$\mathbf{88.92 \pm 2.97}$	93.13 ± 2.03

Table 1: Overall accuracies of different algorithms compared to the FC-LN and the CNN-LN.

During our experiments, we observed that the convolutional Ladder Network either converges to an overall accuracy of $\approx 96\%$ or $\approx 90\%$ and that the chance of converging to 96% naturally increases with the number of labeled points. This shows that some points do carry more information than others and it would be interesting to try filter out the points not carrying any valuable information in order to guarantee convergence.

The algorithms we compared the Ladder Networks to are: **CNN**[Chen et al. (2016)], **CDL-MD-L**[Ma et al. (2016b)], **Co-DC-CNN**[Zhang et al. (2017)], **PNGrow**[Romaszewski et al. (2016)], **TT-AL-MSH-MKE**[Tan et al. (2016)], **S^2CoTraC**[Appice et al. (2017)] and **Co-DC-Res**[Fang et al. (2018a)]. The CNN is a basic deep CNN that uses spectral-spatial features to classify HSI images. CDL-MD-L is a contextual deep learning algorithm for the semi-supervised setting. It uses multi-decision labeling and is

based on self-training. PNGrow is a semi-supervised algorithm that has two experts, namely the P and N-expert, which learn features from the spatial and spectral structure of the images respectively. PNGrow uses a co-training approach, just like Co-DC-CNN and Co-DC-Res, which employ a dual-channel (DC) strategy and use CNN's and ResNets as their building blocks. TT-AL-MSH-MKE is an active-learning based approach using a tri-training method for spectral-spatial learning. It furthermore uses a combination of MLR,KNN and ELM (MKE). S^2CoTraC is a semi-supervised, co-training based algorithm that extracts information at pixel level using collective inference. It should be noted that the overall accuracies reported were recorded by Fang et al. (2018a). It is apparent that Co-DC-Res outperforms all other methods in the case of 10 labeled points per class, so it is worth explaining it in a little bit more detail: The algorithm proposed by Fang et al. consists of two main iterations. In the first iteration, the labeled points are used to train two ResNets, which are using the spectral and spatial information respectively. Both classifiers are then used to classify the unlabeled points and a sample selection strategy is used to select the most confident points for training the ResNets again. The employed sample selection strategy is based on the similarity of two given points. For the similarity metric, Fen et al. used a hierarchical representation of the input x, namely $[r_1, r_2, r_3]$, which are the output of the first convolutional layer and the two building blocks, respectively.

The striking difference of the Ladder Network compared to most of the other methods is the joint use of the labeled and unlabeled data. This has proven to be effective in the case of only a few labeled examples. In the following section, we investigate the effect of different noise levels, injected into the layers, as well as the effect of PCA as a pre-processing step and different denoising costs for the reconstruction of the layers.

Figure 6 illustrates the effect of different standard deviations of noise injected into the layers of the encoder. The noise encourages the learning of a denoising function that must learn to extract valuable information. However, denoising is only applied when a poor reconstruction induces some cost. That is why the Ladder Network has a denoising cost for each layer. The effect of having no reconstruction cost can be seen in Figure 7. For the experiment in Figure 6, we used a constant denoising cost for all layers, namely 0.1. This allows us to show the effect of different noise levels, without the side-effect of unevenly distributed denoising costs. As can be seen, no noise (std =0.0) causes the Ladder Network to have utterly poor performance. The injection of minor noise lifts up the accuracy level dramatically and the best standard deviation lies around 0.5. Note that too much noise causes a high level of corruption in the input, making it nearly impossible to extract features, as can be seen towards the end of the line.
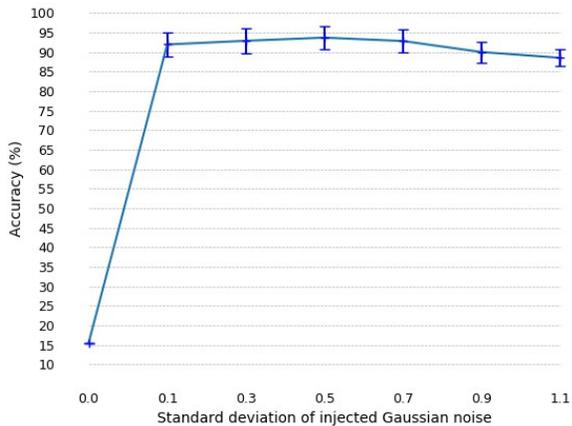
Figure 6: Accuracy of the convolutional Ladder Network with different standard deviations for the injected noise.
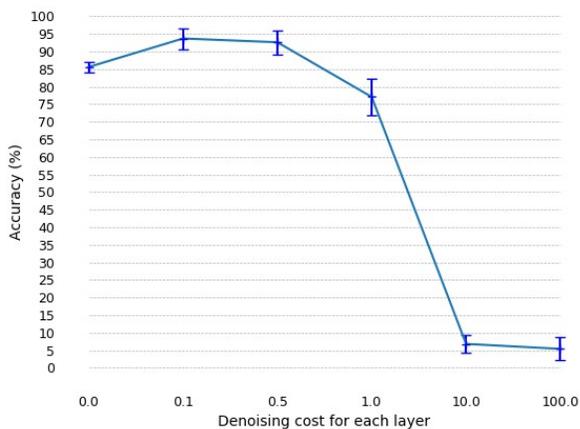

Figure 7: Accuracy of the convolutional Ladder Network with different reconstruction costs at constant noise injection level (0.5).

Figure 7 shows that in the case of only few labeled data points, the reconstruction cost is crucial for learning meaningful features. However, one has to tune this parameter with caution, since too much emphasis on the reconstruction causes extraction of features irrelevant to the supervised task, as can be seen in Figure 7. Because the chance of picking the wrong features for reconstruction is partially eliminated by dimensionality reduction, we applied PCA as a preprocessing step. Figure 8 confirms the hypothesis that training on too many principle components leads to overfitting on irrelevant features and training on too few principle components causes loss of valuable information. Just like

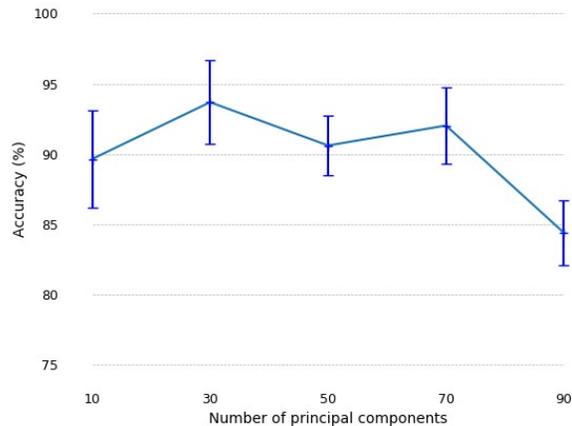the noise level, something in between is the optimal choice for above reasons.

Figure 8: Accuracy of the convolutional Ladder Network with different number of principle components for training.

Discussion

We have shown that the Ladder Network is a highly competitive architecture for semi-supervised hyperspectral image classification. In particular, we have shown that the performance of the Ladder Network given only very few labeled data points is superior to all state-of-the-art methods. However, there still remain some problems with this architecture: (1) The convergence, given only few labeled datapoints, is not guaranteed and therefore causes high fluctuations in the accuracies compared to the fully-labeled case. (2) The temporal-like structure of the spectral bands in each datapoint is not being exploited. Mou et al. (2017) have successfully applied deep Recurrent Networks (RNN) to hyperspectral image classification by making use of the sequence-like structure of the data. In "Recurrent Ladder Networks", Ilin et al. (2017) propose a recurrent extension of the Ladder Network. However, as for the convolutional Ladder Network, the implementation is far from trivial and we are currently working on applying it to HSI classification.

Code

The code for the Autoencoders, the fully connected Ladder Network and the convolutional Ladder Network can be found here: [**https://github.com/jubueche/Convolutional-LadderNet**].

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Alhichri, H., Bazi, Y., Alajlan, N., and Ammour, N. (2015). A hierarchical learning paradigm for semi-supervised classification of remote sensing images. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4388–4391.

Andekah, Z. A., Naderan, M., and Akbarizadeh, G. (2017). Semi-supervised hyperspectral image classification using spatial-spectral features and superpixel-based sparse codes. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 2229–2234.

Appice, A., Guccione, P., and Malerba, D. (2017). A novel spectral-spatial co-training algorithm for the transductive classification of hyperspectral imagery data. *Pattern Recognition*, 63:229 – 245.

Cao, J., Chen, Z., and Wang, B. (2016). Deep convolutional networks with superpixel segmentation for hyperspectral image classification. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3310–3313.

Chen, Y., Jiang, H., Li, C., Jia, X., and Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251.

Fang, B., Li, Y., Zhang, H., and Cheung-Wai Chan, J. (2018a). Semi-supervised deep learning classification for hyperspectral image based on dual-strategy sample selection. *Remote Sensing*, 10:574.

Fang, S., Quan, D., Wang, S., Zhang, L., and Zhou, L. (2018b). A two-branch network with semi-supervised learning for hyperspectral classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 3860–3863.

Ilin, A., Prémont-Schwarz, I., Hao, T. H., Rasmus, A., Boney, R., and Valpola, H. (2017). Recurrent ladder networks. *CoRR*, abs/1707.09219.

Jamshidpour, N., Homayouni, S., and Safari, A. (2016). Graph-based semi-supervised hyperspectral image classification using spatial information. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Liu, B., Yu, X., Zhang, P., Tan, X., Yu, A., and Xue, Z. (2017a). A semi-supervised convolutional neural network for hyperspectral image classification. *Remote Sensing Letters*, 8(9):839–848.

Liu, Q., Sun, Y., Hang, R., and Song, H. (2017b). Spatial–spectral locality-constrained low-rank representation with semi-supervised hypergraph learning for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(9):4171–4182.

Ma, X., Wang, H., and Geng, J. (2016a). Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9):4073–4085.

Ma, X., Wang, H., and Wang, J. (2016b). Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 120:99 – 107.

Mou, L., Ghamisi, P., and Zhu, X. X. (2017). Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655.

Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. (2015). Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672.

Romaszewski, M., Głomb, P., and Cholewa, M. (2016). Semi-supervised hyperspectral classification from a small number of training samples using a co-training approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 121:60 – 76.

Tan, K., Zhu, J., Du, Q., Wu, L., and Du, P. (2016). A novel tri-training technique for semi-supervised classification of hyperspectral images based on diversity measurement. *Remote Sensing*, 8(9).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408.

Wang, J., Zhang, G., Cao, M., and Jiang, N. (2016). Semi-supervised classification of hyperspectral image based on spectral and extended morphological profiles. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4.

Wu, H. and Prasad, S. (2018). Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(3):1259–1270.

Wu, J., Jiang, Z., Zhang, H., Cai, B., and Wei, Q. (2016). Semi-supervised conditional random field for hyperspectral remote sensing image classification. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2614–2617.

Xu, J., Liu, P., Sun, L., and Xiao, L. (2018). Discriminative pixel-pairwise constraint-guided extreme learning machine for semi-supervised hyperspectral image classification. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1518–1522.

Zhan, Y., Wu, K., Liu, W., Qin, J., Yang, Z., Medjadba, Y., Wang, G., and Yu, X. (2018). Semi-supervised classification of hyperspectral data based on generative adversarial networks and

neighborhood majority voting. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 5756–5759.

Zhang, H. and Li, Y. (2016). Spectral-spatial classification of hyperspectral imagery based on deep convolutional network. In *2016 International Conference on Orange Technologies (ICOT)*, pages 44–47.

Zhang, H., Li, Y., Zhang, Y., and Shen, Q. (2017). Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sensing Letters*, 8(5):438–447.

Zhang, Z. and Crawford, M. M. (2016). Semi-supervised multimetric active learning for classification of hyperspectral images. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1843–1846.

Zhong, Z., Li, J., Ma, L., Jiang, H., and Zhao, H. (2017). Deep residual networks for hyperspectral image classification. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1824–1827.