

# Forward-Forward Algorithm for Hyperspectral Image Classification: A Preliminary Study

Sidike Paheding and Abel A. Reyes-Angulo

*Dept. of Applied Computing, Michigan Technological University,  
Houghton, MI 49931, USA*

## Abstract

The back-propagation algorithm has long been the de-facto standard in optimizing weights and biases in neural networks, particularly in cutting-edge deep learning models. Its widespread adoption in fields like natural language processing, computer vision, and remote sensing has revolutionized automation in various tasks. The popularity of back-propagation stems from its ability to achieve outstanding performance in tasks such as classification, detection, and segmentation. Nevertheless, back-propagation is not without its limitations, encompassing sensitivity to initial conditions, vanishing gradients, overfitting, and computational complexity. The recent introduction of a forward-forward algorithm (FFA), which computes local goodness functions to optimize network parameters, alleviates the dependence on substantial computational resources and the constant need for architectural scaling. This study investigates the application of FFA for hyperspectral image classification. Experimental results and comparative analysis are provided with the use of the traditional back-propagation algorithm. Preliminary results show the potential behind FFA and its promises.

## 1 Introduction

Deep Learning(DL) [1] has been revolutionizing many different fields due to its ability to achieve unprecedented performance when applied to real-world problems, including applications in agriculture [2, 3], medicine [4], cyber-security [5], and many others [6, 7, 8, 9, 10].

Hyperspectral image (HSI) contains an extensive array of continuous spectral information across numerous narrow bands. The inherent high-dimensionality of hyperspectral data poses substantial obstacles to accurate classification, owing to intricate spectral variations and a scarcity of labeled samples. Deep learning models, specifically convolutional neural networks (CNNs) [11], have exhibited remarkable accomplishments in numerous computer vision tasks, notably including the classification of hyperspectral images [12, 13]. Nevertheless,

the conventional backpropagation algorithm, which is commonly employed for training deep learning models, may confront certain limitations within this domain, such as computational or energy cost, and sensitivity to initial conditions [14, 15].

By propagating errors in a backward manner through the network, the backpropagation algorithm calculates gradients that guide the adjustment of the model’s parameters to minimize the objective function [16]. While backpropagation has proven effective in deep learning applications, it encounters difficulties when dealing with hyperspectral data such as limited availability of labeled samples. As a result, there is a need for alternative training approaches that can enhance the performance of deep learning models specifically in the context of hyperspectral image classification.

In this study, we investigate the performance of the forward-forward algorithm (FFA) [15] for the task of hyperspectral image classification. The FFA explores the relationships among input data samples by feeding forward both the original data (positive data) and an alternative version of this data (negative data), encouraging the model to learn robust features that capture the underlying characteristics of the HSI.

Our initial experiments show that solely using the FFA does not yield better results compared to the backpropagation algorithm. However, considering the advantages of both methods, we propose to combine the forward-forward pass algorithm with traditional backpropagation for hyperspectral image classification. The idea is to incorporate the forward-forward algorithm as an initial learning stage, allowing the model to learn more discriminative features. Subsequently, the model is fine-tuned using the backpropagation algorithm, which refines the learned representations and optimizes the classification performance. To the best of our knowledge, this work is the first attempt to utilize FFA for HSI classification task.

The remaining sections of this paper are structured as follows: Section 2 presents an overview of the FFA explored in this work. Section 3 details the proposed hybrid approach. Section 4 presents the benchmark dataset utilized for the experiments. Section 5 provides the experimental setup and analyzes the obtained experimental results. Finally, Section 6 provides a summary of the findings from our study.

## 2 Methods

### 2.1 The backpropagation

Introduced by Rumelhart et al. [16, 17], in the backpropagation algorithm for training artificial neural networks, the process involves computing the gradient of the cost function with respect to the network’s parameters and using this information to update the weights via gradient descent. Backpropagation has demonstrated strong generalization capabilities and effectiveness in handling non-linearities, making it applicable to a wide range of neural network types in-

cluding feedforward networks, recurrent networks, and convolutional networks.

## 2.2 The Forward-Forward algorithm

Forward-forward algorithm (FFA) [15] involves substituting the conventional forward and backward passes from the backpropagation algorithm, with two forward passes that function in a parallel manner but on distinct data with opposing objectives. The affirmative pass involves real data and modifies the weights to improve the goodness in each hidden layer, while the negative pass operates on “negative data” and modifies the weights to diminish the goodness within every hidden layer. In [15], two distinct criteria were investigated for measuring quality: the sum of the squared neural activities and the negative sum of the squared activities, although numerous other criteria can also be utilized. In the original FFA, the sum of the squares of the activities in the layer is expressed as

$$G = \sum_j z_j^2, \quad (1)$$

where  $z_i$  represents the activity of the  $j^{th}$  hidden unit.

Furthermore, the positive and negative passes adjust the weights locally, and the probability of the outputs are expressed as follows:

$$\text{prob}(\text{positive}) = \sigma(G - \theta) \quad (2)$$

where  $\sigma$  despite a logistic distribution function, and and  $\theta$  a given threshold.

To facilitate the contrast of positive and negative data during the supervised training process of FFA, we need to develop a method for merging the data with their corresponding labels. In [15], Hinton proposed to overlay the label information onto the data itself or embed the label within the input data. However, in this work, we take a different approach by appending the label at one end of the spectral signature of each sample. Moreover, we explore various methods for encoding the label information. For instance, we experiment with one-hot encoding representation, binary representation, and decimal representation. Through the analysis of experimental results, it is found that using the one-hot encoding representation to append the label information with the hyperspectral signature data yields better outcomes. As a result, all the experiments reported in this work utilizing FFA are trained using this approach. Figure 1 visually depicts the imputation method employed, where the label-encoded information is appended at one end of the pixel’s hyperspectral signatures.

## 3 FFA for hyperspectral image classification

In this preliminary study, we contemplate the implementation of FFA with the use of fully connected layers and 1D convolutional layers for HSI classification. The fully connected layer performs a linear transformation on the input data, usually in a vector shape, through weights matrix multiplication operations, in

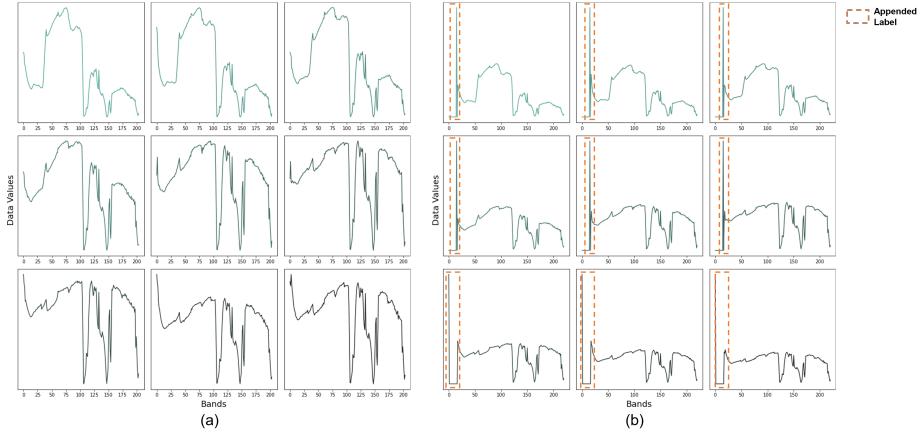


Figure 1: Illustration of label embedding method. (a) shows spectral signatures from the Salinas hyperspectral dataset. (b) shows the same samples with the appended label information, which is appended at the beginning of spectral signatures.

addition to a bias term. The fully connected layers comprise the contributions of all inputs for a final prediction. In contrast, convolutional layers in CNN can be used to capture local dependencies in sequential data, such as time series or text. Unlike fully connected layers that operate on the entire input, CNN considers the local receptive field of the input at a time. This allows them to extract features that are sensitive to local patterns and variations.

### 3.1 Fully Connected FFA network for pixel-wise HSI classification

The classification of hyperspectral images poses a substantial challenge attributed to the data's high dimensionality and spectral complexity. Deep learning architectures have shown promise in extracting discriminative features from hyperspectral images. Nonetheless, the efficacy of these architectures is heavily contingent upon the quality of the learned representations. In this study, we explore the use of the FFA with Fully Connected layers to enhance for HSI classification task. FFA comprises the use of a few hidden layers to extract features from the HSI data, scale it to a latent space, and produce the final pixel-wise classification. A total of 3 hidden layers were used in our FFA, with the following number of units: 784, 500, and 500, respectively.

### 3.2 Convolutional FFA network for pixel-wise HSI classification

In this work, we explore the implementation of the aforementioned types of neural network layers, limited to 1D data. Convolutional layers for 1D data are usually implemented as neural networks that convolve the input of the hyperspectral image with a set of learnable filters. These layers detect local relationships and patterns within the interaction of the hyperspectral bands per pixel. The capture of this spectral signature enables the architecture to learn discriminative representations of the data. By using the FFA technique, the network aims to update the weights layer by layer in a forward pass only, relieving the computational load of computing the gradient during the backpropagation of the error to update the learned parameters.

The implementation of the 1D CNN layer allows us to implement an FFA architecture similar to the type used for HSI classification [12]. The proposed FFA comprises the use of 1D CNN layers in the early stages to capture feature representations within a different latent dimensional space, while the fully connected layers are used at the end to learn how to properly discriminate among the classes. The network is configured as follows: an initial 1D convolutional layer with 64 kernels of size 64, followed by a set of two hidden layers with 128 and 256 feature maps, respectively, both with a kernel size of 36. Then, a max-pooling operation is applied to downscale the dimensions of the tensor by a factor of two. Another 1D CNN layer is applied with 256 kernels of size 36. This last 1D CNN layer is followed by another max-pooling operation with similar characteristics as the previous one, and a flattening operation. Finally, two fully connected layers are added with 100 and  $N$  units, where  $N$  represents the number of classes in the HSI dataset.

### 3.3 Combination of FFA with backpropagation

Given the similarities between the nature of FFA and the training procedure in contrasting learning, we propose the utilization of FFA during the initial stage of training. In this stage, each sample is contrasted with different output choices, enabling the model to learn how to effectively discriminate between the correct prediction and other alternatives. Subsequently, the model proceeds to refine its learning through traditional backpropagation using the same deep learning architecture. This process facilitates the model in adjusting the extraction of meaningful characteristics from the high-level spectral information and fine-tuning the latent representation for accurate final predictions. After this initial phase, the model transitions to the standard backpropagation technique, which leverages the deep learning model's architecture for further refinement. During this stage, the model fine-tunes its representation, optimizing its capacity to capture significant features from the high-dimensional spectral data.

## 4 Datasets

For simplicity of this proof-of-concept, we perform experiments over two publicly available dataset<sup>1</sup>: The Salinas valley and the Indian Pines.

### 4.1 The Salinas Valley

The Salinas dataset is a popular hyperspectral image dataset that is commonly used in the realms of remote sensing and image processing. It is named after the Salinas Valley in California, USA, where the data was collected. The dataset consists of a hyperspectral image of size  $512 \times 217$  pixels, with 224 spectral bands covering the range from 0.2 to 2.4 micrometers. Each pixel in the image represents a small area on the ground, and the spectral bands capture the reflectance of the surface at different wavelengths. The Salinas dataset was collected using an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, which was flown over an agricultural area in the Salinas Valley. The image contains 16 different crop types, including lettuce, broccoli, and bare soil, among others.

### 4.2 The Indian Pines

Collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over an agricultural area in Indiana, USA, the Indian Pines dataset consists of a hyperspectral image of size  $145 \times 145$  pixels, with 224 spectral bands covering the range from 0.4 to 2.5 micrometers. Each pixel in the image represents a small area on the ground, and the spectral bands capture the reflectance of the surface at different wavelengths. The dataset contains 16 different land cover classes, including crops, trees, roads, and buildings, among others.

## 5 Results

### 5.1 Experimental setup

To ensure a fair comparison, we evaluate the different techniques reported in this work using the same training and test datasets. The Salinas and Indian Pines HSI datasets were split into training, validation, and testing sets in an 8:1:1 ratio, respectively. To obtain robust performance estimates, we repeat the entire experimental process three times and average the results.

For each experiment, the respective models are trained for 250 epochs using the Adam optimizer with a fixed learning rate of  $1 \times 10^{-3}$ . When employing backpropagation, categorical cross-entropy was utilized as the loss function. However, when FFA is used, a custom loss function is implemented to measure the distance between the *goodness* of each positive and negative sample with respect to the provided threshold.

---

<sup>1</sup>[www.ehu.eus/ccwintco/index.php/Hyperspectral Remote Sensing Scenes](http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes)

Table 1: Summary of the qualitative results using the different techniques, mentioned in this work, over Salinas and the Indian Pines dataset. Best performance in marked in **bold** font.

Method	Salinas			Indian Pines		
	OA	AA	$\kappa$	OA	AA	$\kappa$
BP	0.9190	<b>0.9605</b>	0.9099	<b>0.8109</b>	<b>0.7759</b>	<b>0.7842</b>
FFA (Dense layers)	0.8392	0.8470	0.8206	0.5921	0.5841	0.5370
FFA (Dense & Conv layers)	0.9101	0.8518	0.8666	0.6660	0.5822	0.6194
FFA + BP	<b>0.9221</b>	0.9564	<b>0.9130</b>	0.7365	0.6978	0.6978

All the experiments are run on an NVIDIA RTX 3070 graphic card with 8GB of dedicated GPU.

## 5.2 Performance comparison

To assess and contrast the performance of the different techniques presented in this work, we employed the following evaluation metrics:

- **Overall Accuracy (OA):** This metric provides the percentage of correctly classified pixels from the respective HSI dataset.
- **Average Accuracy (AA):** This metric is computed by averaging each-class accuracy score, thus providing a class-specific evaluation of the technique’s performance.
- **Kappa Coefficient ( $\kappa$ ):** This metric measures the agreement between the predicted and true class labels, in which the accuracy that could be achieved by chance is taken into account.

The use of these evaluation metrics collectively enables a comprehensive assessment of the various techniques presented in this work for HSI classification.

Table 1 summarizes the experimental results obtained by evaluating the discussed techniques using the aforementioned performance metrics on the Salinas and Indian Pines HSI datasets. As shown in Table 1, when considering the Salinas HSI dataset, the combination of FFA and backpropagation (BP) achieved the best performance in terms of OA (0.9221) and  $\kappa$  (0.9130). However, BP alone achieved the highest AA (0.9605). On the other hand, for the Indian Pines HSI dataset, BP exhibits the best performance across all evaluation metrics, with OA (0.8109), AA (0.7759), and  $\kappa$  (0.7842). Nevertheless, the utilization of FFA in combination with BP demonstrated a significant improvement compared to using any of the FFN variants individually. Figure 2 and Figure 3 demonstrate classification maps of different models using the Salinas and Indian Pines datasets, respectively.

## 5.3 Discussion

In our experiments, we conduct a performance comparison of three different architectures: (1) a deep learning model trained solely using backpropagation,

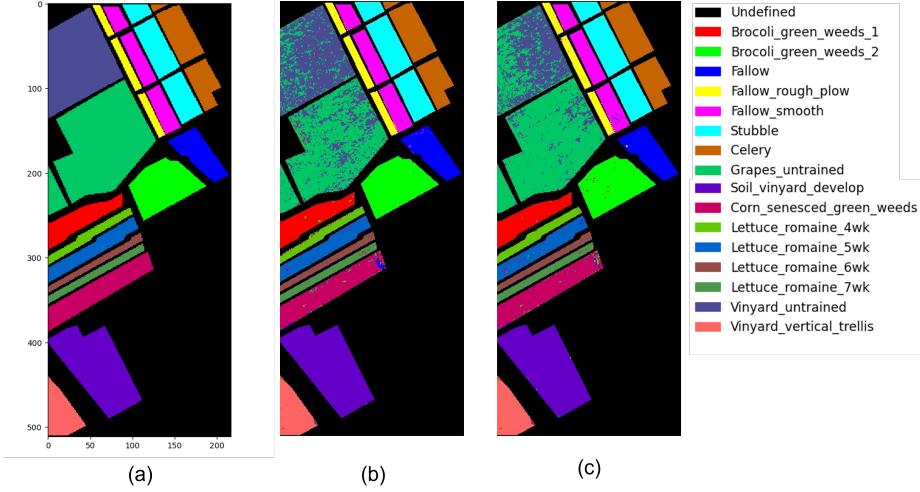


Figure 2: Illustration of the predicted classification maps over the Salinas HSI dataset. The images contain (a) the ground truth, (b) the prediction from the use of traditional BP, and (c) the predictions using FFA+BP (c)

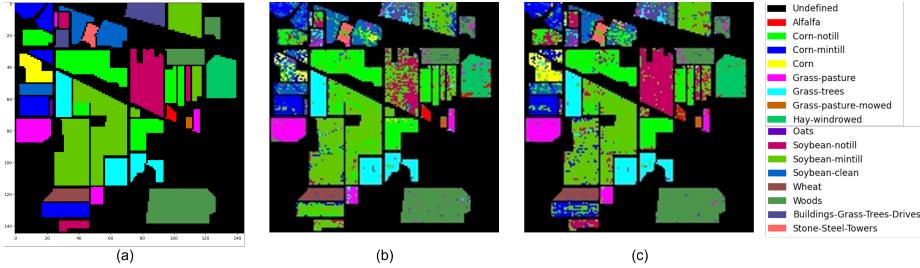


Figure 3: Illustration of the predicted classification maps over the Indian Pines HSI dataset. The images contain (a) the ground truth, (b) the prediction from the use of traditional BP, and (c) the predictions using FFA+BP

(2) a model trained exclusively with FFA, and (3) our proposed approach that combines FFA pre-training with subsequent fine-tuning using backpropagation. The results clearly illustrate the effectiveness of the combined approach in enhancing feature representation and improving classification accuracy, compared to FFA-only methods.

By combining FFA pre-training with subsequent fine-tuning using backpropagation, we leverage the respective strengths of both approaches. The FFA initializes the network with meaningful representations, which is then fine-tuned through backpropagation to adapt the network to the specific classification task at hand. This combination allows the network to capture both useful spectral characteristics obtained from the FFA and task-specific discriminative features

during the backpropagation process. The synergy between these two approaches yields promising results in our experiments.

## 6 Conclusion

In summary, the integration of the Forward-Forward algorithm and traditional backpropagation during the early stages of training proved to be highly effective in enhancing feature representation and improving classification performance in hyperspectral image analysis. By incorporating the FFA algorithm, the network was able to capture useful feature representation by adjusting network parameters in every hidden layer. Subsequent fine-tuning through backpropagation facilitated the extraction of discriminative task-specific features. This combined approach exemplifies the potential for leveraging the strengths of different learning algorithms to achieve superior results in hyperspectral image classification tasks.

## References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] A. Kiliaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and electronics in agriculture*, vol. 147, pp. 70–90, 2018.
- [3] M. Maimaitijiang, V. Sagan, P. Sidike, S. Hartling, F. Esposito, and F. B. Fritsch, “Soybean yield prediction from uav using multimodal data fusion and deep learning,” *Remote sensing of environment*, vol. 237, p. 111599, 2020.
- [4] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, *et al.*, “Opportunities and obstacles for deep learning in biology and medicine,” *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.
- [5] S. Mahdavifar and A. A. Ghorbani, “Application of deep learning to cybersecurity: A survey,” *Neurocomputing*, vol. 347, pp. 149–176, 2019.
- [6] D. Bourilkov, “Machine and deep learning applications in particle physics,” *International Journal of Modern Physics A*, vol. 34, no. 35, p. 1930019, 2019.
- [7] U. Khan, S. Paheding, C. P. Elkin, and V. K. Devabhaktuni, “Trends in deep learning for medical hyperspectral image analysis,” *IEEE Access*, vol. 9, pp. 79534–79548, 2021.

- [8] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, “U-net and its variants for medical image segmentation: A review of theory and applications,” *Ieee Access*, vol. 9, pp. 82031–82057, 2021.
- [9] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.
- [10] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari, “A state-of-the-art survey on deep learning theory and architectures,” *electronics*, vol. 8, no. 3, p. 292, 2019.
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [12] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, “Deep convolutional neural networks for hyperspectral image classification,” *Journal of Sensors*, vol. 2015, pp. 1–12, 2015.
- [13] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.
- [14] R. S. Sexton and J. N. Gupta, “Comparative evaluation of genetic algorithm and backpropagation for training neural networks,” *Information sciences*, vol. 129, no. 1-4, pp. 45–59, 2000.
- [15] G. Hinton, “The forward-forward algorithm: Some preliminary investigations,” *arXiv preprint arXiv:2212.13345*, 2022.
- [16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.