COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK
The New York Academy of Sciences

SPELA: Learning without Backpropagation

Aditya Somasundaram*, Pushkal Mishra*, Ayon Borthakur
as7458@columbia.edu, pumishra@ucsd.edu, ayon.borthakur@iitg.ac.in

The Motivation

The troubles with backpropagation

- Backpropagation is admittedly beautiful, but it does require a ton of matrix multiplications
- Could there be a way to learn faster: something which trades accuracy for speed?
- Is there a way to learn quicker: looking at a few samples only a handful of times?

Connection to Neuroscience

Observations from Biology

- SPELA uses neural priors to integrate external world information in the form of embedded vectors
- SPELA supports complete local Hebbian learning
- Inference can be performed at any layer, introducing a mechanism to study the speed-accuracy tradeoff (SAT)
- **None of the following is present:** update locking of weights, transport of weights, storage of activation, and backpropagation!
- SPELA has few shot and few epoch learning capabilities, similar to the human brain

The Algorithm

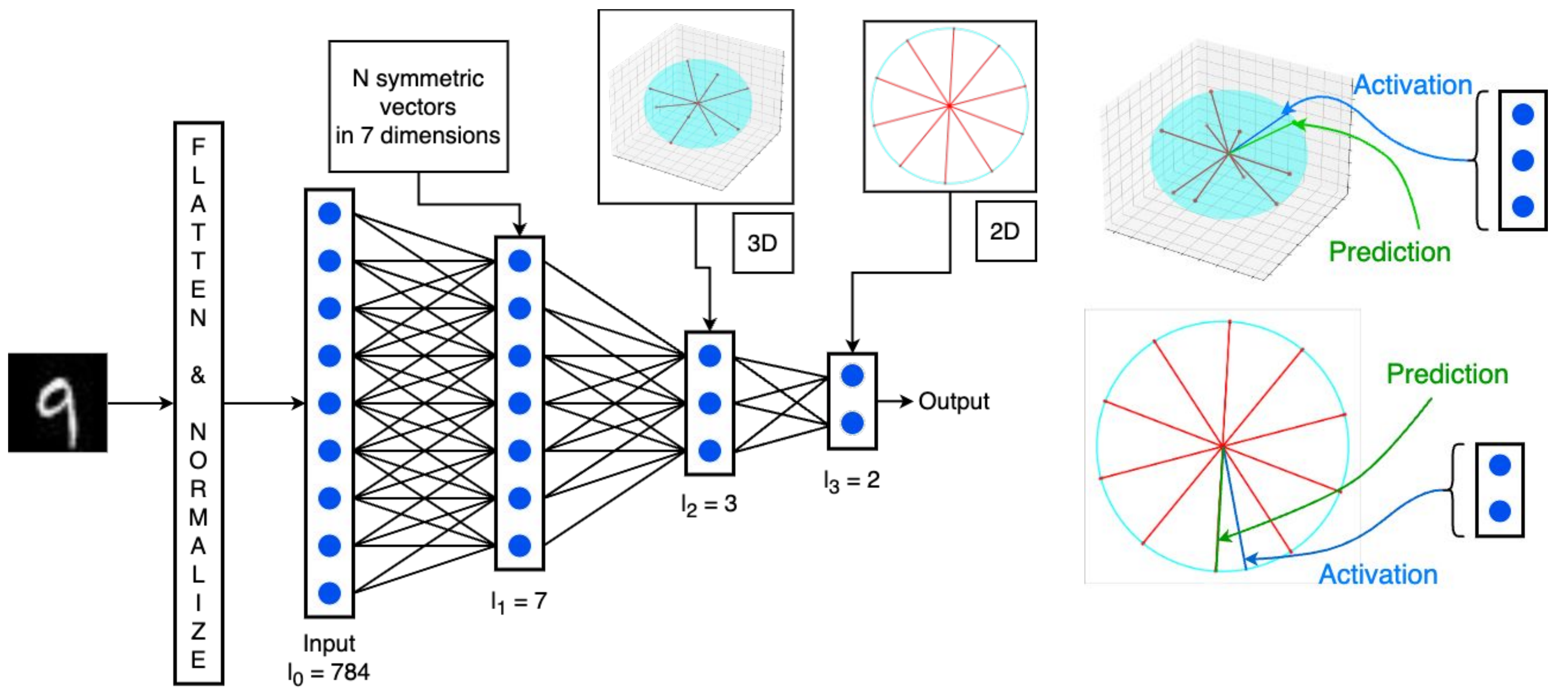
Algorithm 1 Training MLP with SPELA

Given: An input (X), label (l), number of layers (K), number of epochs (E)
Define: $\cos\_\mathrm{sim}(A,B) = \frac{A.B}{||A||.||B||}$ ▷ Dot product of normalized vectors
Set: $h_0 = x$
for $k \leftarrow 1$ to K **do** ▷ Iterate through layers
  for $e \leftarrow 1$ to E **do** ▷ Iterate through epochs
    $h_{k-1} = \mathrm{normalize}(h_{k-1})$
    $h_k = \sigma_k(W_k h_{k-1} + b_k)$
    $\mathrm{loss} = -\cos\_\mathrm{sim}(h_k, \mathrm{vecs}_k(l))$ ▷ $\mathrm{vecs}_k(.)$: set of embedded vectors
    $W_k \leftarrow W_k - \alpha * \Delta_{W_k}(\mathrm{loss})$ ▷ Weight update using local loss
    $b_k \leftarrow b_k - \alpha * \Delta_{b_k}(\mathrm{loss})$ ▷ Weight update using local loss
  end for
end for

Switching the inner and outer loops would result in further optimization of SPELA! This would result in updating weights of all layers in a serial fashion using a single batch of data.

tldr;

Network Architecture and Training Mechanism



A layerwise training mechanism for local updates:

1. Classification occurs at every layer
2. The loss from the classification at layer L updates the weights of layer L

Local loss: Every layer has C number of fixed embedded vectors. Each of these vectors is assigned a class to represent. We use a cosine loss to measure the closeness of the activation vector to the corresponding embedded vector.
Training: A simple layerwise weight update from the layerwise loss. No need to propagate it backwards!

An increase in number of layers would lead to more non linearities, hence better classification.

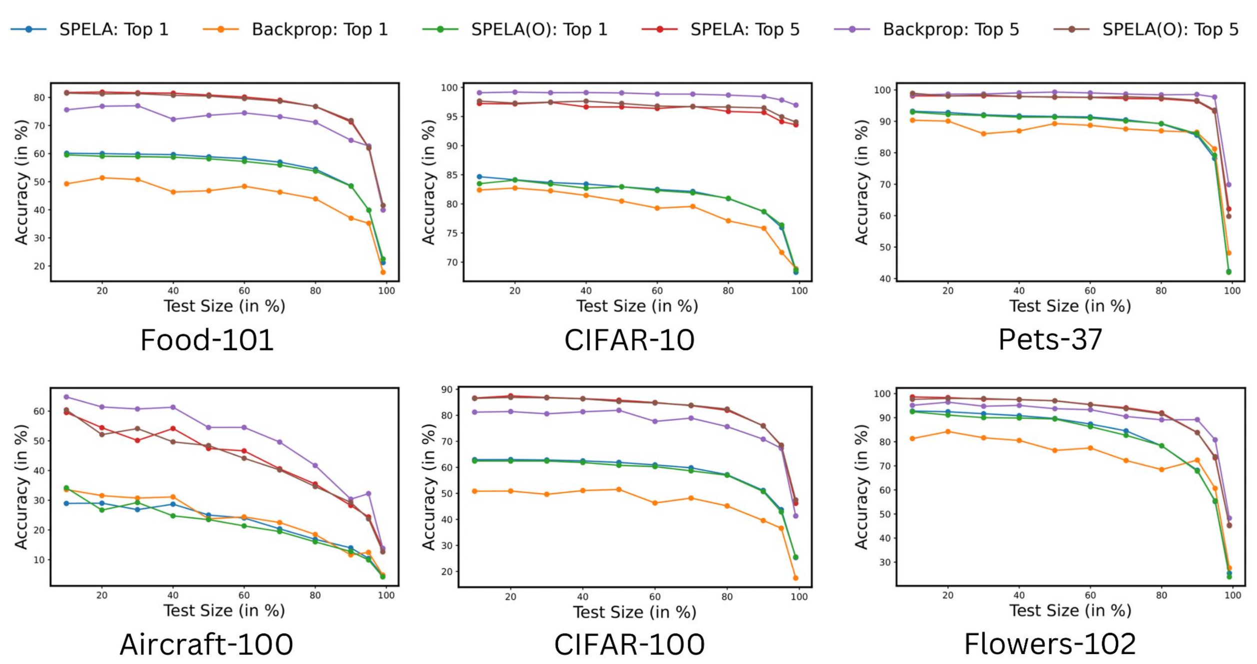
Results & Conclusion

Classification Accuracy

Task	Test Size	BP Test Accuracy	SPELA Test Accuracy
MNIST-10	0.2	**97.6 ± 0.1%**	94.8 ± 0.2%
	0.9	92.7 ± 0.3%	**94.7 ± 0.4%**
	0.99	82.8 ± 2.7%	**94.1 ± 0.5%**
KMNIST-10	0.2	**92.5 ± 0.9%**	87.5 ± 0.3%
	0.9	81.6 ± 0.4%	**82.1 ± 0.7%**
	0.99	**72.2 ± 0.8%**	63.5 ± 3.1%
Fashion MNIST-10	0.2	83.0 ± 2.2%	**85.4 ± 1.6%**
	0.9	**82.2 ± 1.6%**	81.4 ± 3.2%
	0.99	**74.2 ± 3.0%**	69.6 ± 3.6%
Network Intrusion-2	0.2	84.2 ± 1.7%	**96.4 ± 0.1%**
	0.9	87.7 ± 3.3%	**96.5 ± 0.1%**
	0.99	60.3 ± 8.6%	**96.6 ± 0.3%**
Fraud Detection-2	0.2	94.9 ± 0.9%	**95.5 ± 0.1%**
	0.9	94.4 ± 1.1%	**95.3 ± 0.1%**
	0.99	90.8 ± 2.3%	**95.5 ± 0.1%**

Comparison of classification accuracy is presented for some test sizes ratios. SPELA performs better than BP in most scenarios.

Transfer Learning by training MLP head



A ResNet50 model pretrained on ImageNet-100 is used to extract features and subsequently fed into the new MLP head. These features are then used to train both SPELA and BP.

Few Shot-1 Epoch Accuracy

Task	Test Size	BP Test Accuracy	SPELA Test Accuracy
MNIST-10	0.95	69.4 ± 4.9%	**94.1 ± 0.3%**
	0.99	32.6 ± 6.0%	**93.0 ± 0.2%**
KMNIST-10	0.95	61.2 ± 1.4%	**82.5 ± 0.7%**
	0.99	32.9 ± 5.4%	**79.0 ± 0.8%**
Fashion MNIST-10	0.95	62.9± 6.7%	**79.9 ± 3.6%**
	0.99	31.0± 8.9%	**75.5 ± 2.9%**

The few-shot 1 epoch learning capabilities of SPELA are put forth.

Ablations

Euclidean Loss

$$\mathrm{loss} \neq \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||\cdot||\mathbf{y}||}$$
$$\mathrm{loss} = ||\mathbf{x}-\mathbf{y}||^2$$

Performance is maintained when the loss is changed from cosine similarity to Euclidean distance.

Random Vector Embeddings

$$\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$$

Performance is maintained despite choosing vector embeddings randomly.

Binarizing weights

$$W = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Performance is maintained if a longer network is used with binarized weights.

Comparison to other algorithms

Learning Methods	BP	FF	PEP	MPE	SPELA
Forward Pass	1	2	2	3	1
Backward Pass	1	0	0	0	0
Weight Update	1	2	1	1	1
Loss function	global	local	global	global	local
Activations	all	current	all	current	current

Comparison of SPELA with other local learning algorithms. FF stands for Forward-Forward, PEP for PEPITA, and MPE for MEMPEPITA.

Conclusion

- We showcase a local learning algorithm which operates without backpropagation
- We have unique features which tie SPELA to biological plausibility and provide a framework to study biological learning mechanisms
- We showcase SPELA few shot and few epoch learning capabilities, all without bias!

References

Somasundaram, A., Mishra, P. and Borthakur A., 2024. Representation Learning Using a Single Forward Pass. *arXiv preparing arXiv:2402.09769.*