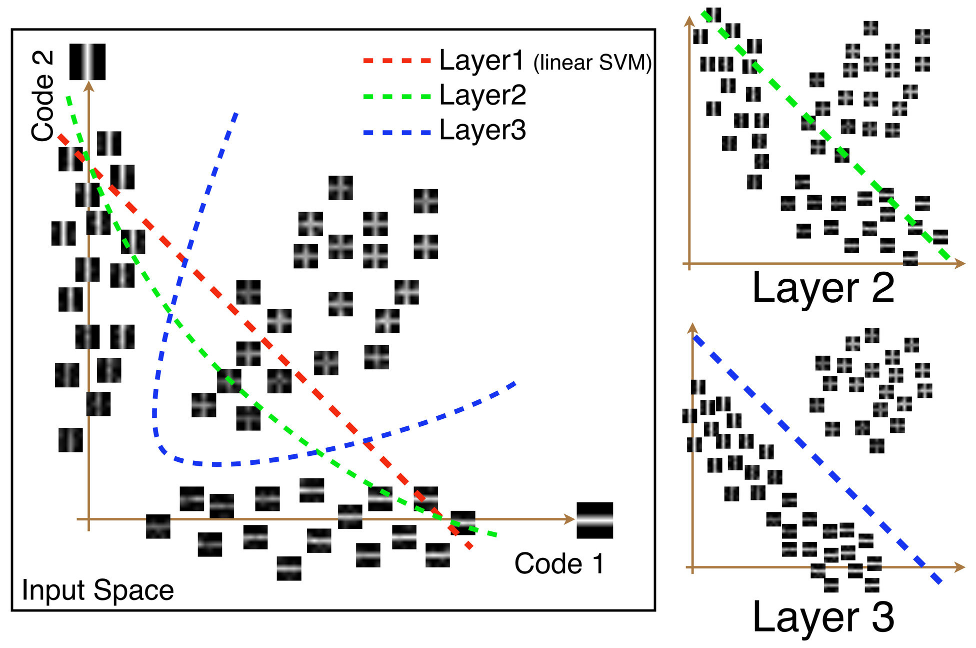
Learning with Recursive Perceptual Representations

Oriol Vinyals   Yangqing Jia   Li Deng   Trevor Darrell
UC Berkeley EECS & ICSI & Microsoft Research

1. Contribution

The key contributions of our work are:

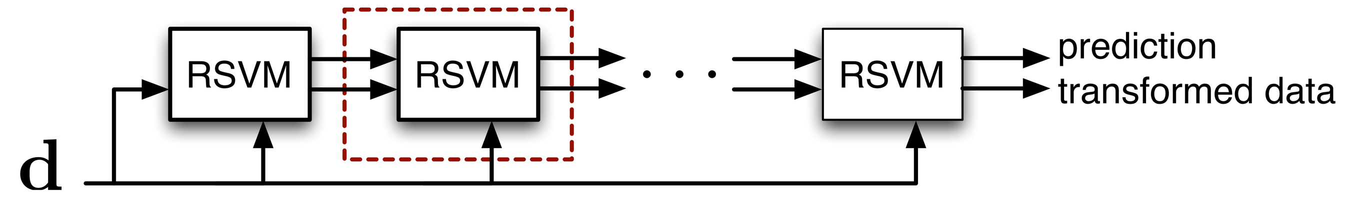
- We propose a new method based on linear SVMs, random projections, and deep architectures.
- The method enriches linear SVMs without forming explicit kernels.
- The learning only involves training linear SVMs, which is very efficient. No fine-tuning is needed in training the deep structure.
- The training could be easily parallelized.



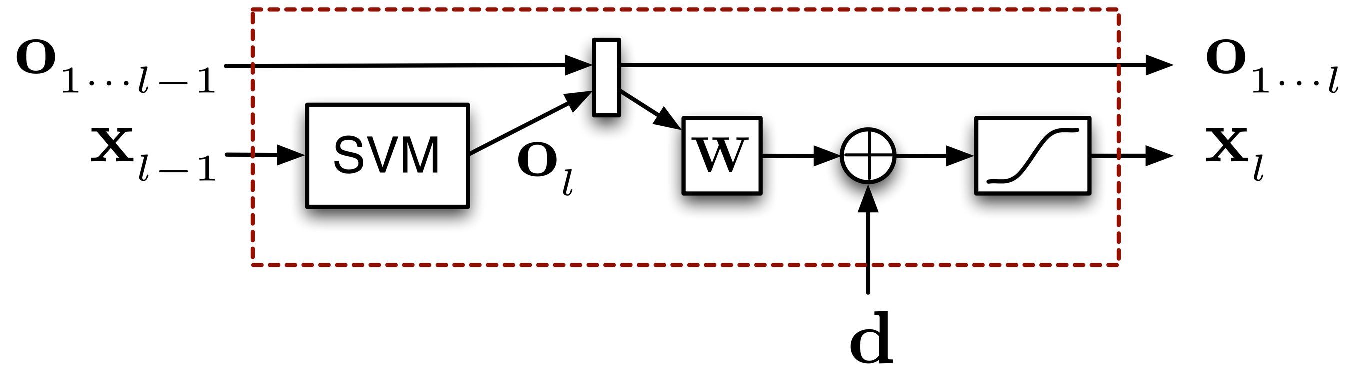
- **Based on the success of sparse coding + linear SVMs, we stacked linear SVMs introducing a non-linear discriminative bias to achieve nonlinear separation of the data.**

2. The Pipeline

- $\mathbf{o}_l = \boldsymbol{\theta}_l^T \mathbf{x}_l$
- $\mathbf{x}_{l+1} = \sigma(\mathbf{d} + \beta \mathbf{W}_{l+1}[\mathbf{o}_1^T, \mathbf{o}_2^T, \cdots, \mathbf{o}_l^T]^T)$
- $\boldsymbol{\theta}_l$ are the linear SVM parameters trained with $\mathbf{x}_l$
- $\mathbf{W}_{l+1}$ is the concatenation of l random projection matrices $[\mathbf{W}_{l+1,1}, \mathbf{W}_{l+1,2}, \cdots, \mathbf{W}_{l+1,l}]$
- Each $\mathbf{W}_l$ is a random matrix sampled from $N(0,1)$



The whole R^2SVM pipeline



Each RSVM component

3. Analysis

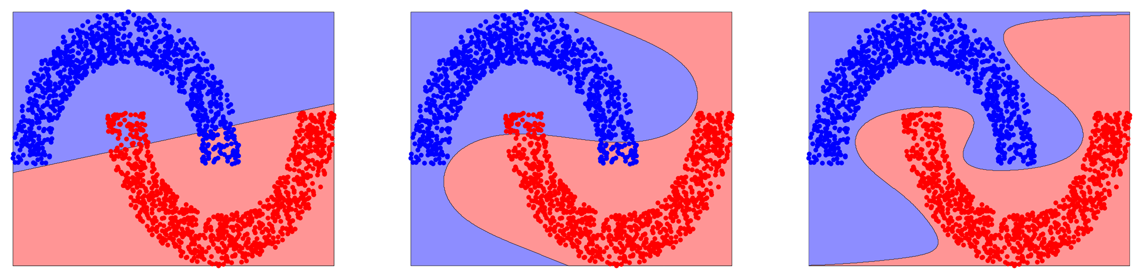
- We would like to "pull apart" data from different classes.
- Quasi-orthogonality: two random vectors in a high-dimensional space are much likely to be approximately orthogonal.
- In the perfect label case, we can prove that

Lemma 3.1. – *$\mathcal{T}$, set of N tuples $(\mathbf{d}^{(i)}, y^{(i)})$*

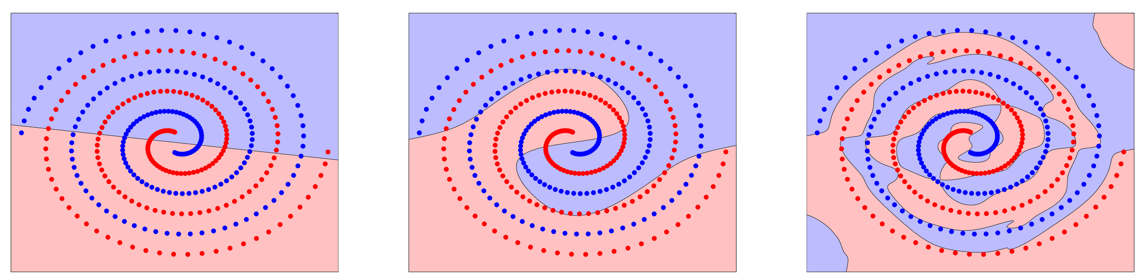
– *$\boldsymbol{\theta} \in \mathbb{R}^{D\times C}$ the corresponding linear SVM solution with objective function value $f_{\mathcal{T},\boldsymbol{\theta}}$*

– *There exist $\mathbf{w}_i$ s.t. $\mathcal{T}' = (\mathbf{d}^{(i)} + \mathbf{w}_{y^{(i)}}, y^{(i)})$ has a linear SVM solution $\boldsymbol{\theta}'$ with $f_{\mathcal{T}',\boldsymbol{\theta}'} < f_{\mathcal{T},\boldsymbol{\theta}}$.*

- With imperfect prediction, each layer incrementally "improves" the separability of the original data.
- Randomness helps avoid over-fitting (as will be shown in the experiments).



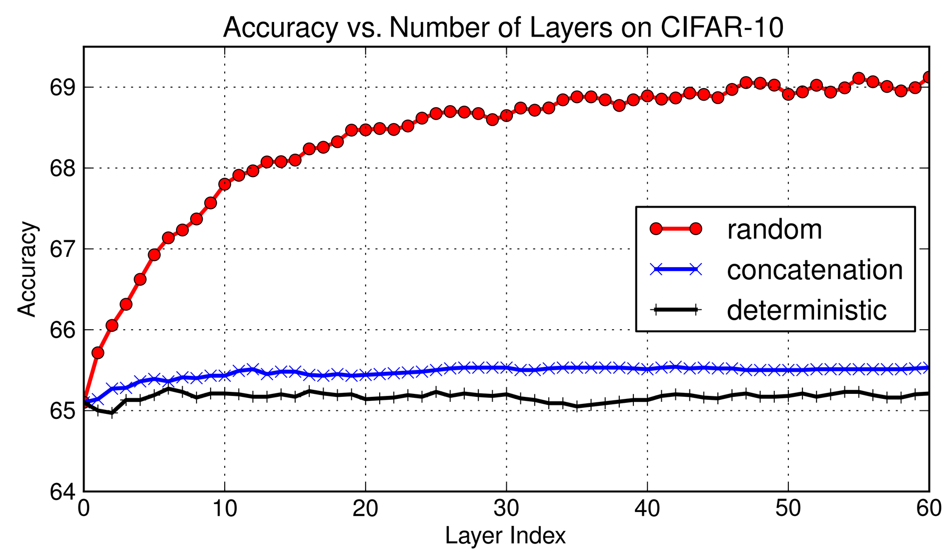
layer 1   layer 2   final layer



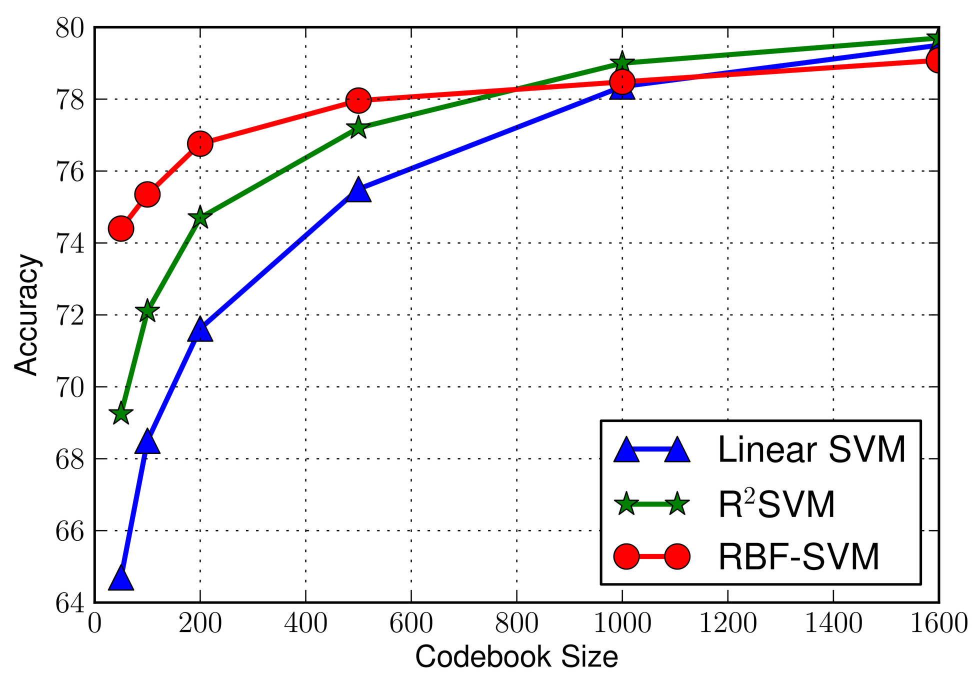
layer 1   layer 2   final layer

5. CIFAR-10

- Going deeper with randomness helps, while naive combination does not.



- Performance under different feature size
  - Small codebook size: R^2SVM improves performance without much additional complexity.
  - Large codebook size: R^2SVM avoids the over-fitting issue of nonlinear SVMs.



6. Results

Experimental results on both the vision (CIFAR-10) and the speech (TIMIT) data.

CIFAR10

Method	Tr. Size	Code. Size	Acc.
Linear SVM	25/class	50	41.3%
RBF SVM	25/class	50	42.2%
R^2SVM	25/class	50	42.8%
DCN	25/class	50	40.7%
Linear SVM	25/class	1600	44.1%
RBF SVM	25/class	1600	41.6%
R^2SVM	25/class	1600	45.1%
DCN	25/class	1600	42.7%

TIMIT

Method	Phone state accuracy
Linear SVM	50.1% (2000 codes) 53.5% (8000 codes)
R^2SVM	53.5% (2000 codes) 55.1% (8000 codes)
DCN, learned per-layer	48.5%
DCN, jointly fine-tuned	54.3%

MNIST

Method	Err.
Linear SVM	1.02%
RBF SVM	0.86%
R^2SVM	0.71%
DCN	0.83%
NCA w/ DAE	1.0%
Conv NN	0.53%

7. Summary and Discussions

Comparison over Different Models

Method	Tr	Te	Sca	Rep
Deep NN	×	✓	?	✓
Linear SVM	✓	✓	✓	×
Kernel SVM	?	?	×	✓
DCN	×	✓	?	✓
R^2SVM	✓	✓	✓	✓

- Tr: ease of training the model.
- Te: testing time complexity.
- Sca: scalability (does it handle large-scale well?).
- Rep: the representation power of the model.

Final Remarks

1. Non-sparse coded features: we applied the method on several UCI datasets and observed similar performance to kernel SVMs.
2. Number of layers: ~5 (TIMIT / MNIST), ~10-20 (CIFAR), depending on the nonlinear nature of data.

8. References

- J Yang, K Yu, and Y Gong. Linear spatial pyramid matching using sparse coding for image classification. In CVPR, 2009.
- A Coates and A Ng. The importance of encoding versus training with sparse coding and vector quantization. In ICML, 2011.
- L Deng and D Yu. Deep convex network: A scalable architecture for deep learning. In Interspeech, 2011.