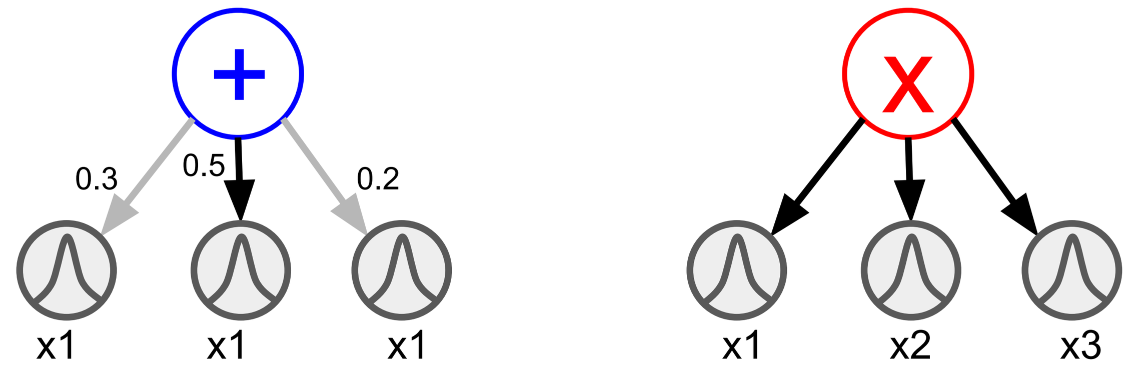
Sum Product Network with VAE Leaves

P. L. Tan*, R. Peharz*, T. Adel*
* Department of Engineering, University of Cambridge

UNIVERSITY OF CAMBRIDGE

What are Sum Product Networks?

SPN uses **sum nodes** and **product nodes** to compose many simple univariate distributions into a large complicated multivariate distribution



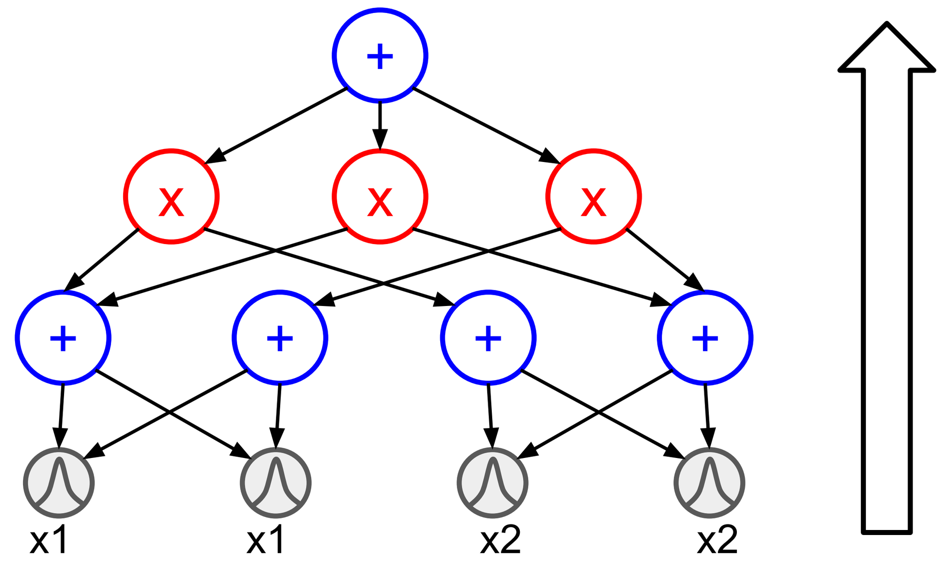
Mixture of gaussians
(over same random variable)

Diagonal multivariate gaussian
(over different random variables)

How to compose:

- **Sum nodes** over **disjoint** sets of random variables can be composed into a **product node**
- **Product nodes** over **identical** sets of random variables can be composed into a **sum node**

By recursively applying these two simple rules, we can build deep, complicated yet compact representations of multivariate distributions.



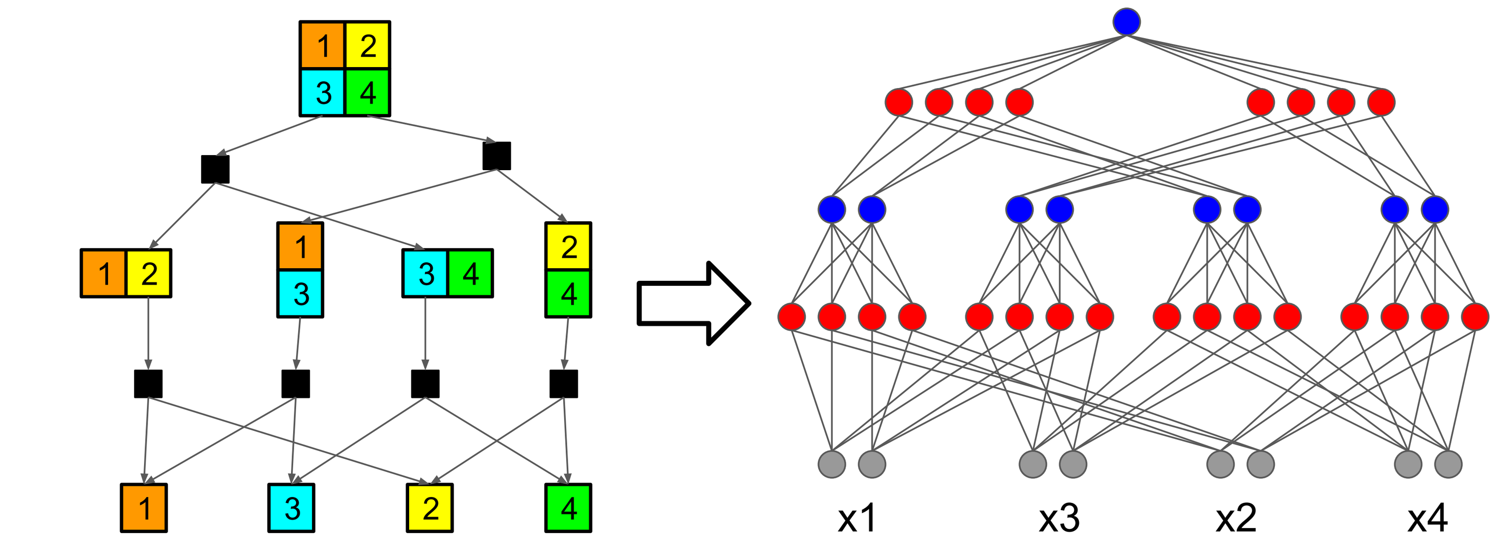
Composing into more complicated distribution

Properties of SPN

Easy	Hard
Sample Learn parameters Marginalise missing R.V. Infer missing R.V. Compute likelihood	Learn structure

SPNs for Images

Poon and Domingos[1] proposed a general dense SPN structure for images by recursively partitioning the image.



Region Graph
(constructed by recursively partitioning image)
[**black** - partition node; color - scope node]

Dense SPN
(constructed from Region Graph)
[grey - univariate distribution;
blue - sum node; red - product node]

Problem: dense SPN too large!
For a *m* x *n* image, and *K* **sum nodes** per **scope** node

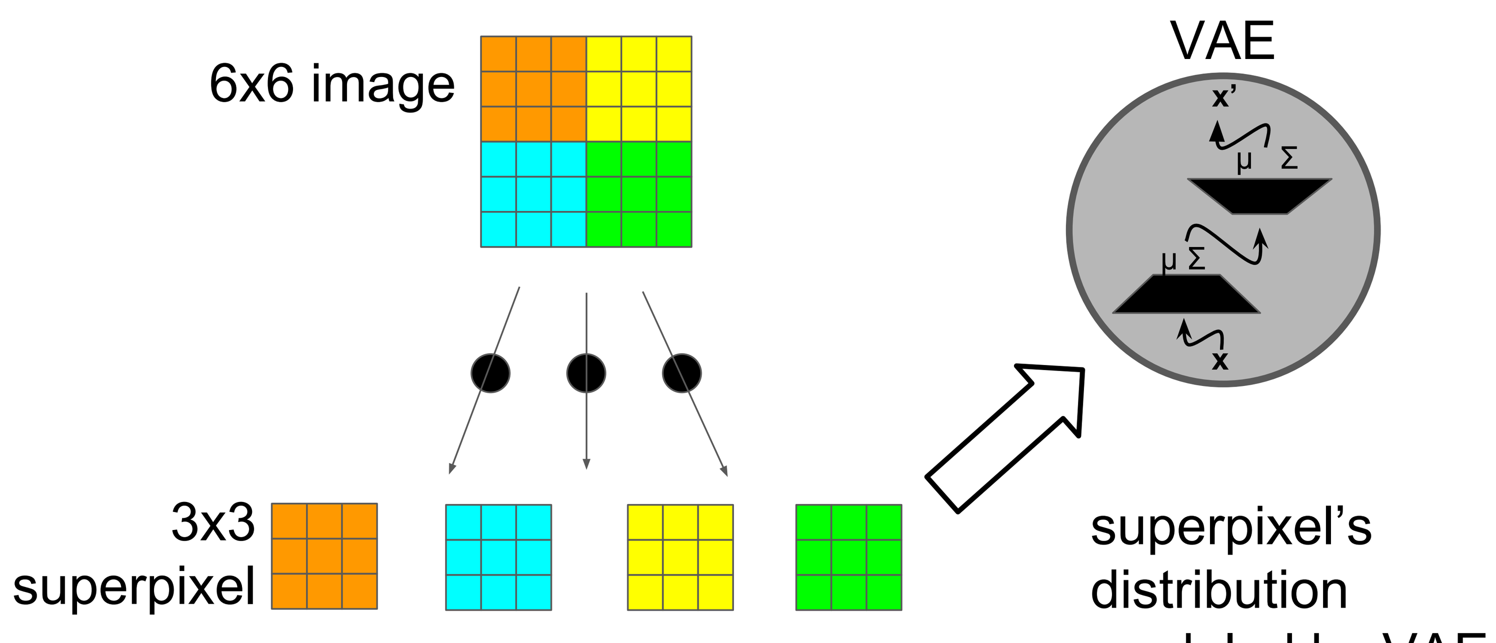
Num of **sum nodes**: $K\frac{m(m+1)n(n+1)}{4}$

Num of **product nodes**: $K^2\frac{(m+1)m(n+1)n(m+n-2)}{12}$

SPN with VAE Leaves

Proposed Solution:

- Keep denseSPN small by decomposing image down to superpixel only.
- Neural generative model (like VAEs) handles distribution of superpixel

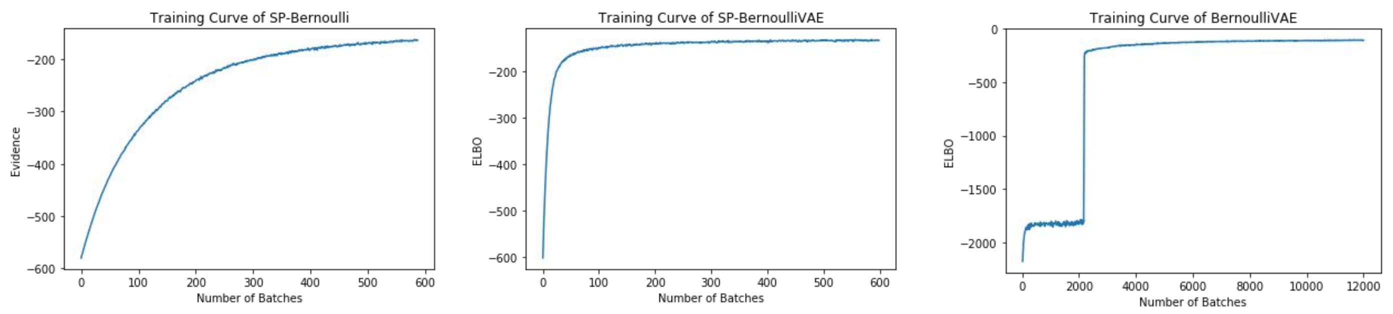


Region Graph
(did not partition smaller than superpixel)

Initial Experiment

Typical Training Curves on MNIST:

- Compare SPN-VAE against vanilla SPN and vanilla VAE
- Vanilla VAE has same size as all VAE leaves combined
- Same learning rate (0.01) and algorithm (Adam)



Vanilla SPN	**SPN + VAE leaves**	Vanilla VAE
1568 bern leaves 9702 **sum nodes** 51744 **product nodes**	98 VAE leaves 1470 **sum nodes** 12544 **product nodes**	Latent dim: 196 1568 hidden recog nodes 1568 hidden gener nodes

- SPN training curves have consistent logarithmic shape
- SPN-VAE training curves have more consistent and faster convergence than Vanilla SPN, though still slower and than vanilla VAE
- VAE training often initially gets stuck in local optimum

Further Work

1. Investigate SPN's regularization effect
2. Evaluate model on more datasets
3. Try other neural generative models like PixelRNN
4. Show that theoretical benefits of SPN are preserved when using VAE leaves
5. Propose and validate a convolutional architecture of SPN with mix of Gaussian and VAE leaves

Contact

plt28@cam.ac.uk

1. H. Poon and P. M. Domingos, *Sum-Product Networks: A New Deep Architecture*, 2012, http://arxiv.org/abs/1202.3732