End-To-End Imitation Learning of Lane Following Policies Using Sum-Product Networks

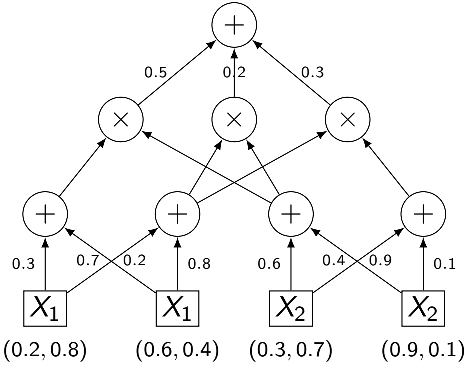
Renato Lui Geh   Denis Deratani Mauá

Institute of Mathematics and Statistics
University of São Paulo

ENIAC 2019

Sum-product networks

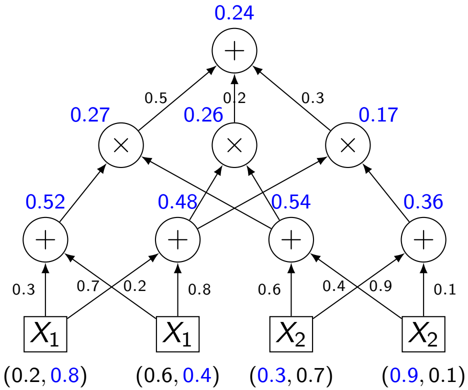
Sum-product networks (SPNs) are deep tractable density estimators with a neural network-like structure subject to only sums and products as activation functions.



In the above example, leaves are binomial distributions over each RV X_i.

Computing probability of evidence

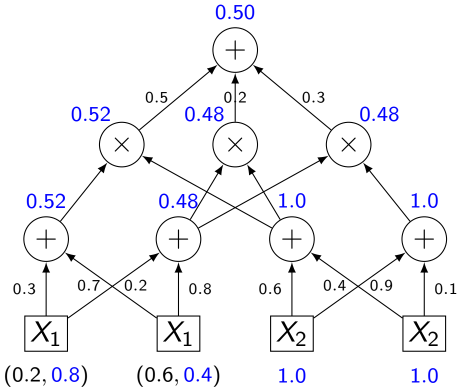
For some evidence $\mathbf{e} = \{X_1 = 1, X_2 = 0\}$, $P(\mathbf{e})$ is the value of the SPN at its root.



$$P(\mathbf{e} = \{X_1 = 1, X_2 = 0\}) = 0.24$$

Computing marginals

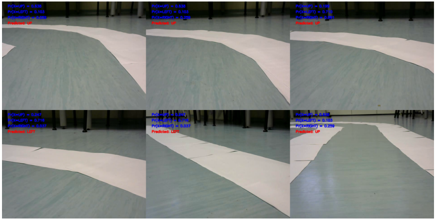
We compute marginals by summing out to 1 missing variables' leaves. Let $\mathbf{X} = \{X_1 = 1\}$. We want to compute $P(\mathbf{X})$:



$$P(\mathbf{X} = \{X_1 = 1\}) = 0.5$$

The task

Given a track, bot must autonomously complete the whole course without going off road. The only available input is a single frontal camera.

Decision making must be done in real-time under a resource constrained, low-budget robot environment.

Why SPNs?

- Probabilistic semantics (efficiently compute probability of input and detect outliers);
- Easy to build and debug (efficient algorithms for learning structure from data; probabilistic semantics);
- No need for heavy packages (e.g. Tensorflow);
- Good performance with much smaller models;
- Not been tried “in the wild” yet.

Robot

Raspberry Pi 3 Model B

CPU: Quad Core 1.2GHz Broadcom BCM2837 64bit ARMv7

Memory: 1GB RAM

Storage: 16GB SSD

Lego Mindstorms NXT

CPU: Atmel AT91SAM7S256 48MHz 32bit ARMv4

Memory: 64KB RAM

Storage: 256KB Flash

+

=

Modelling

Every pixel X_i is a variable in the distribution represented by the SPN, i.e. no additional feature extraction, end-to-end.

Two architectures:

- GD: LearnSPN with k-means and G-test
- DV: Clustering on Variables

Three weight setups:

- g: Generative gradient descent
- d: Discriminative gradient descent
- s: Proportional weights for GD, random weights for DV

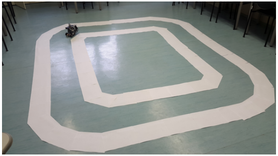
Experiments

Training

- Trained models with samples from a single simple track;
- 500 training samples (corresponding to 0.9% of the dataset);
- Training took from 30 mins to 9 hours (time dependent on parameters, pre-processing and learning algorithms) on an Intel i7-4500U 1.8GHz CPU.

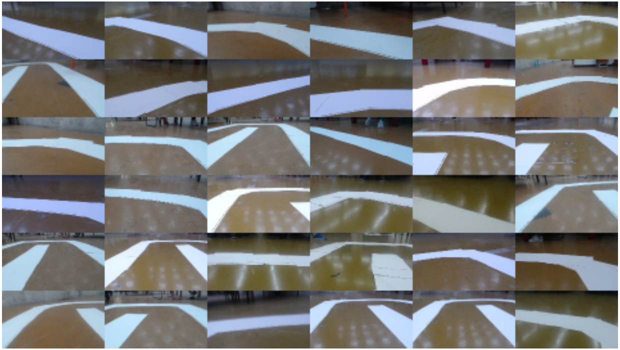
Evaluation

- Tests run on three different tracks;
- Different floor and lighting conditions from training dataset.



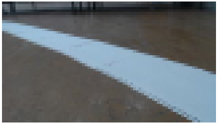
Dataset

Dataset used: Moraes and Salvatore 2018



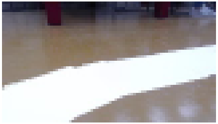
Lane tracking dataset with 80×45 RGB images. Each labeled with either UP, LEFT or RIGHT.

Self-driving as image classification



LEFT

UP

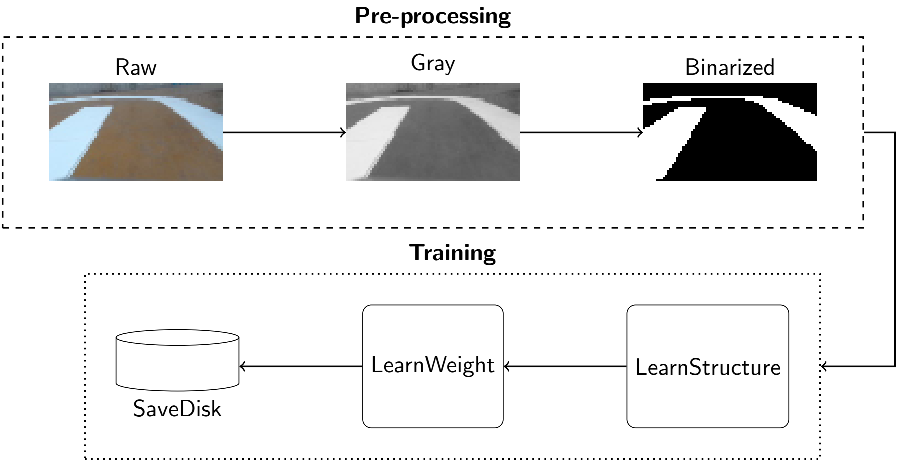


RIGHT

Training track in Moraes and Salvatore 2018.

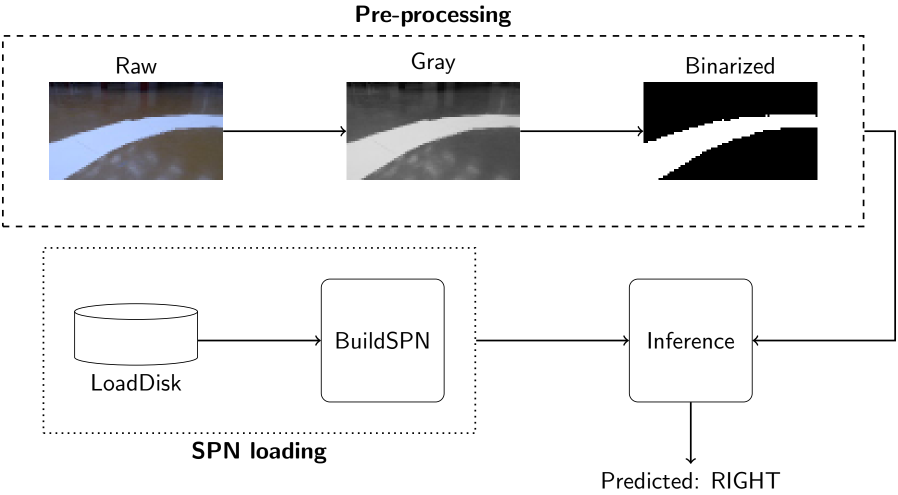
Training

Training was done on an Intel i7-4500U CPU 1.80 Hz.



Saved SPN was then passed to the Raspberry.

Prediction


Pre-processing
Raw
Gray
Binarized
LoadDisk
BuildSPN
SPN loading
Inference
Predicted: RIGHT

Chosen models

Model 1: Q_4, **GD+d**

Accuracy: 78.2%

Desktop time: 170ms

Berry time: 700ms

Model 2: Q_6, **GD+d**

Accuracy: 74.4%

Desktop time: 50ms

Berry time: 150ms

Model 3: $\emptyset$, **GD+d**

Accuracy: 62.4%

Desktop time: $<$ 10ms

Berry time: 75ms

Comparison to neural networks

Comparison to Moraes and Salvatore 2018:

Model	**Accuracy (%)**	**Speed (seconds)**
DFN	81.3	$\approx$1.35
CNN	80.6	$\approx$1.35
Q_4, GD-SPN+d	78.2	$\approx$0.70
Q_6, GD-SPN+d	74.4	$\approx$0.15
GD-SPN+d	62.4	$\approx$0.07

Neural networks were slightly more accurate, but real-time prediction with them is unfeasible.

Our implementation did not make use of the GPU, which could increase speed dramatically.

Real-time decision making poses new challenges: timely decisions are often more important than accurate decisions.

“Real world” scenario

Mobile Robot Self-Driving Through Image Classification Using Discriminative Learning of Sum-Product Networks — YouTube (`https://youtu.be/vhpWQDX2cQU`)

Implementation

Inference and learning: GoSPN
(`https://github.com/RenatoGeh/gospn`)

Mobile robot implementation: GoDrive
(`https://github.com/RenatoGeh/godrive`)

Thank you.

Questions?

Sum-product networks

Definition 1 (Sum-product network).

A sum-product network (SPN) is a DAG where each node n is either:

1. A tractable univariate probability distribution;
2. A product of SPNs: $v_n = \prod_{j \in \text{Ch}(n)} v_j$; or
3. A weighted sum of SPNs: $v_n = \sum_{j \in \text{Ch}(n)} w_{n,j} v_j$.

Where v_n is the value of node n, $\text{Ch}(n)$ its set of children and $w_{n,j}$ the weight of edge $n \to j$.

Accuracy

Accuracy (%)	DV+g	DV+d	DV+s	GD+g	GD+d	GD+s
B	78.8	78.8	78.8	82.8	83.8	85.0
Q_2	78.6	78.0	78.0	78.6	80.4	79.4
$Q_2 + E$	76.6	76.6	76.8	79.6	82.8	81.8
Q_3	77.4	77.4	77.4	77.6	80.2	79.8
$Q_3 + E$	70.4	76.6	76.6	79.2	81.2	77.4
Q_4	78.2	78.4	78.2	76.0	**78.2**	76.4
$Q_4 + E$	76.6	76.6	76.8	76.0	74.6	80.6
Q_5	77.8	78.4	78.4	77.6	74.0	73.8
$Q_5 + E$	76.6	76.6	76.6	72.0	72.8	72.0
Q_6	77.4	78.4	78.4	75.2	**74.4**	72.0
$Q_6 + E$	76.0	76.4	76.4	73.0	75.0	73.6
Q_7	78.2	78.4	78.4	62.8	72.2	71.4
$Q_7 + E$	76.2	76.4	76.4	70.6	71.4	71.6
$\emptyset$	78.0	78.4	78.4	62.4	**62.4**	62.4
E	76.4	76.4	76.4	60.4	60.0	61.2

B: binarization, Q_n: n-bit quantization, E: histogram equalization.

Inference time

Inference (secs)	DV+g	DV+d	DV+s	GD+g	GD+d	GD+s
B	0.23	0.25	0.25	0.38	0.37	0.31
Q_2	0.22	0.24	0.23	0.28	0.34	0.16
$Q_2 + E$	0.22	0.23	0.23	0.38	0.30	0.27
Q_3	0.22	0.23	0.22	0.22	0.32	0.17
$Q_3 + E$	0.22	0.23	0.22	0.34	0.32	0.31
Q_4	0.22	0.22	0.23	0.16	**0.17**	0.13
$Q_4 + E$	0.23	0.27	0.29	0.13	0.14	0.13
Q_5	0.22	0.26	0.28	0.07	0.05	0.02
$Q_5 + E$	0.22	0.29	0.25	0.05	0.05	0.02
Q_6	0.23	0.24	0.23	0.04	**0.05**	0.01
$Q_6 + E$	0.22	0.24	0.28	0.03	0.04	0.02
Q_7	0.23	0.23	0.26	0.03	0.01	0.01
$Q_7 + E$	0.22	0.26	0.24	0.01	0.01	0.01
$\emptyset$	0.22	0.26	0.23	0.02	**0.01**	0.01
E	0.23	0.23	0.22	0.01	0.01	0.02

B: binarization, Q_n: n-bit quantization, E: histogram equalization.

References I

(N.d.).

Bojarski, Mariusz et al. (2016). “End to End Learning for Self-Driving Cars”. In: *CoRR* abs/1604.07316.

Chen, Z. and X. Huang (2017). “End-to-end learning for lane keeping of self-driving cars”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1856–1860.

Conaty, Diarmaid, Cassio Polpo de Campos, and Denis Deratani Mauá (2017). “Approximation Complexity of Maximum A Posteriori Inference in Sum-Product Networks”. In: *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*.

Dennis, Aaron and Dan Ventura (2012). “Learning the Architecture of Sum-Product Networks Using Clustering on Variables”. In: *Advances in Neural Information Processing Systems 25*. NIPS, pp. 2033–2041.

References II

- Ester, Martin et al. (1996). "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. AAAI Press, pp. 226–231.
- Gens, Robert and Pedro Domingos (2012). "Discriminative Learning of Sum-Product Networks". In: *Advances in Neural Information Processing Systems 25*. NIPS, pp. 3239–3247.
- Gens, Robert and Domingos Pedro (17–19 Jun 2013). "Learning the Structure of Sum-Product Networks". In: *Proceedings of the 30th International Conference on Machine Learning*. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, pp. 873–880.
- Hussein, Ahmed et al. (2017). "Imitation Learning: A Survey of Learning Methods". In: *ACM Computing Surveys* 50.2, 21:1–21:35. ISSN: 0360-0300. DOI: 10.1145/3054912.

References III

Levinson, Jesse et al. (2011). "Towards Fully Autonomous Driving: Systems and Algorithms". In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 163–168.

Mei, Jun, Yong Jiang, and Kewei Tu (2018). "Maximum A Posteriori Inference in Sum-Product Networks". In: *AAAI Conference on Artificial Intelligence*.

Moraes, Paula and Felipe Salvatore (2018). *Self-Driving Pi Car*. https://github.com/felipessalvatore/self_driving_pi_car.

Otsu, Nobuyuki (Jan. 1979). "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1, pp. 62–66. ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076.

Paden, Brian et al. (2016). "A survey of motion planning and control techniques for self-driving urban vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1.1, pp. 33–55.

References IV

Pan, Yunpeng et al. (2018). "Agile Autonomous Driving using End-to-End Deep Imitation Learning". In: *Proceedings of Robotics: Science and Systems XIV*.

Peharz, Robert (2015). "Foundations of Sum-Product Networks for Probabilistic Modeling". PhD thesis. Graz University of Technology.

Peharz, Robert et al. (2018). "Probabilistic Deep Learning using Random Sum-Product Networks". In: *CoRR* abs/1806.01910.

Pfeiffer, Mark et al. (2017). "From Perception to Decision: A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots". In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1527–1533.

Pomerleau, Dean A. (1989). "Alvinn: An autonomous land vehicle in a neural network". In: *Advances in Neural Information Processing Systems*, pp. 305–313.

References V

Poon, Hoifung and Pedro Domingos (2011). "Sum-Product Networks: A New Deep Architecture". In: *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*. Corvallis, Oregon: AUAI Press, pp. 337–346.

Rashwan, Abdullah, Pascal Poupart, and Chen Zhitang (Nov. 2018). "Discriminative Training of Sum-Product Networks by Extended Baum-Welch". In: *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*. Vol. 72. Proceedings of Machine Learning Research. Prague, Czech Republic: PMLR, pp. 356–367.

Sguerra, B. M. and F. G. Cozman (2016). "Image Classification Using Sum-Product Networks for Autonomous Flight of Micro Aerial Vehicles". In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 139–144.

Vergari, Antonio, Nicola Di Mauro, and Floriana Esposito (2015). "Simplifying, Regularizing and Strengthening Sum-Product Network Structure Learning". In: *ECML/PKDD*.

References VI

Zhao, Han, Tameem Adel, et al. (2016). "Collapsed Variational Inference for Sum-Product Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR, pp. 1310–1318.

Zhao, Han, Pascal Poupart, and Geoffrey J Gordon (2016). "A Unified Approach for Learning the Parameters of Sum-Product Networks". In: *Advances in Neural Information Processing Systems 29*. NIPS, pp. 433–441.