# Logistic Circuits

Kushagra Chandak

4th July, 2019

# Introduction to logistic circuits

- Layered circuit representation, comprised of OR and AND gates. Circuit representations achieve SOTA in density estimation.
- Used as classifiers; discriminative counterparts to probabilistic circuits.
- Parameter learning can be reduced to logistic regression, therefore convex optimization.
- Structure learning by local search from probabilistic circuit learning.
- Image classification tasks: less parameters and data efficient.

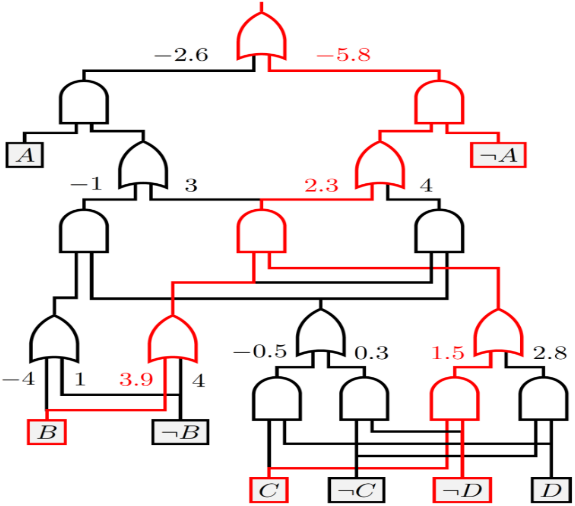
# Representation and some properties

- Represented as a logical circuit (which represents a logical sentence).
- Inner nodes: AND/OR gates; leaf node: a Boolean literal.
- Every AND gate is *decomposable (disjoint inputs)* and every OR gate is *deterministic (at most one input set to 1)*.
- Real-valued parameters $\theta_i$ associated with input wires to every OR gate.

# Flow and weight functions

- Boolean circuit flow of $\mathbf{x}$ between node n (OR) and child c:
  $f(n, \mathbf{x}, c) = \begin{cases} 1 & \text{if } \mathbf{x} = c \\ 0 & \text{otherwise} \end{cases}$
- Weight function for node n, $g_n(x)$:
  - Leaf node: $g_n(\mathbf{x}) = 0$
  - AND gate: $g_n(\mathbf{x}) = \sum_{i=1}^{m} g_{c_i}(\mathbf{x})$
  - OR gate: $g_n(\mathbf{x}) = \sum_{i=1}^{m} f(n, \mathbf{x}, c_i) \cdot (g_{c_i}(\mathbf{x}) + \theta_i)$
- Posterior distribution on class variable Y:
  $\Pr(Y = 1|\mathbf{x}) = \frac{1}{1+\exp(-g_r(\mathbf{x}))}$
- Real-valued (Bernoulli) inputs: probabilities of the corresponding Boolean RVs.
- Probabilistic circuit flow:
  $f(n, \mathbf{q}, c) = \Pr_{\mathbf{q}}(c|n) = \frac{\Pr_{\mathbf{q}}(c \wedge n)}{\Pr_{\mathbf{q}}(n)} = \frac{\Pr_{\mathbf{q}}(c)}{\Pr_{\mathbf{q}}(n)}$

# Example: $A = 0$, $B = 1$, $C = 1$, $D = 0$, $Pr(Y = 1|A, B, C, D)$



# Learning $\theta_i$

- Any logistic circuit model can be reduced to a logistic regression model over a particular feature set.
- $\Pr(Y = 1|\mathbf{x}) = \frac{1}{1+\exp(-\mathcal{X}\cdot\boldsymbol{\theta})}$
- Decomposing $g_n(x)$ into $\mathcal{X}.\theta$ $\forall n$? (proof by induction)
- Learning $\theta_i$ equivalent to logistic regression on features $\mathcal{X}$. Can use gradient descent now.

# Global circuit flow features

- $\mathcal{X}$ = global circuit flow features.
- Global circuit flow $f_r(n, \mathbf{x}, c)$ (top-down):
  - Between root r and its child c: $f_r(r, \mathbf{x}, c) = f(r, \mathbf{x}, c)$
  - AND gate, global flow from c: $f_r(n, \mathbf{x}, c) = \sum_{i=1}^{m} f_r(v_i, \mathbf{x}, n)$
  - OR gate, global flow from c: $f_r(n, \mathbf{x}, c) = f(n, \mathbf{x}, c) \cdot \sum_{i=1}^{m} f_r(v_i, \mathbf{x}, n)$
- Top-down weight function: $g_r(\mathbf{x}) = \sum_{(n,\theta,c)\in\mathcal{W}} f_r(n, \mathbf{x}, c) \cdot \theta$
- Gates in logistical circuit correspond to some logical sentence. Hence, values of $\mathcal{X}$ correspond to probability of these logical sentences according to input $\mathbf{x}$.
- Computing global flow features (offline, before parameter learning):
  - Bottom-up linear pass to calculate node probabilities.
  - Top down approach to calculate $\mathcal{X}$.