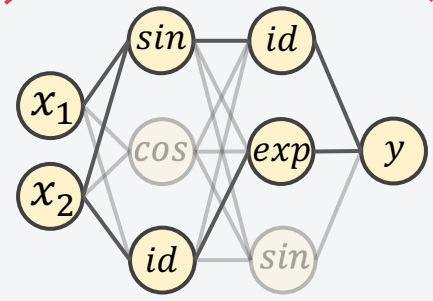
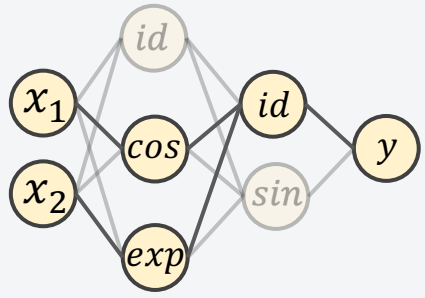
2. Instantiate and train DySymNet via backpropagation and weight pruning

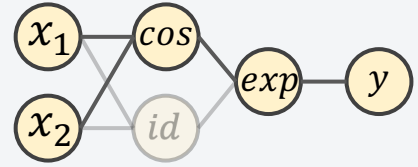
DySymNet:



$y_1 = sin(1.3x_1 + 0.5x_2) + 1.5x_2$

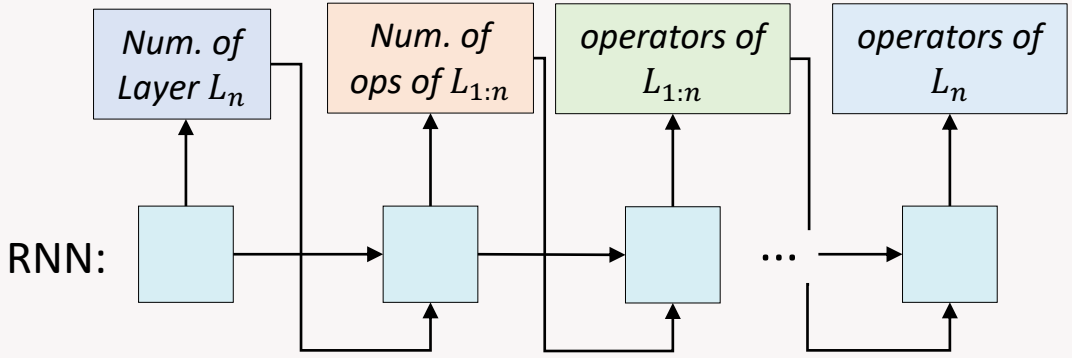


$y_i = cos(x_1) + exp(0.6x_2)$



$y_N = exp(cos(0.9x_1 + 1.2x_2))$

1. Sample N DySymNet descriptions autoregressively via RNN



3. Refine Constant and train RNN via risk-seeking policy gradient with entropy

$$\tilde{f}_i^{*}(x) = \mathrm{BFGS}(y_i)$$

$$\nabla_\theta J(\theta;\varepsilon) \approx \frac{1}{\varepsilon N}\sum_{i=1}^{N}\left[R\left(\tilde{f}_i^{*}|\mathcal{D}\right) - R_\varepsilon\right]\cdot \mathbf{1}_{R\geq R_\varepsilon}\nabla_\theta \log p\left(\tilde{f}_i^{*}|\theta\right)$$

$$\nabla_\theta \mathcal{H}(\theta) = \lambda_H \sum_{i=1}^{N}\nabla_\theta \mathcal{H}\left(\tilde{f}_i^{*}|\theta\right)$$