A Deep Architecture for Log-Linear Models

Simon Luo[1,2], Sally Cripps[1,2] and Mahito Sugiyama[3,4]

[1] School of Mathematics and Statistics, The University of Sydney [2] Data Analytics for Resources and Environments (DARE), Australian Research Council

[3] National Institute of Informatics [4] JST, PRESTO

Thirty-fourth Annual Conference on Neural Information Processing Systems (NeurIPS 2020) Workshop on Deep Learning through Information Geometry

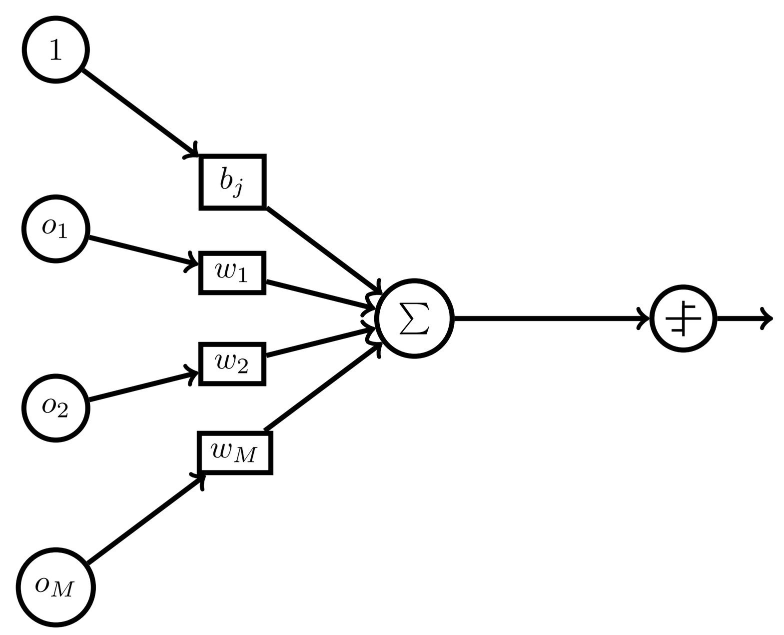
Summary

- Formulated a deep learning architecture using a partial order structure
- Bias and edge weights are realized on different layers
- Does not require gradients to update the parameters
- Minimizes Kullback-Leibler Divergence from a set of samples to our poset
- Uses statistical EM-Algorithm (Expectation-Maximization) for optimization
- Closed form formulae for both E-step and M-step

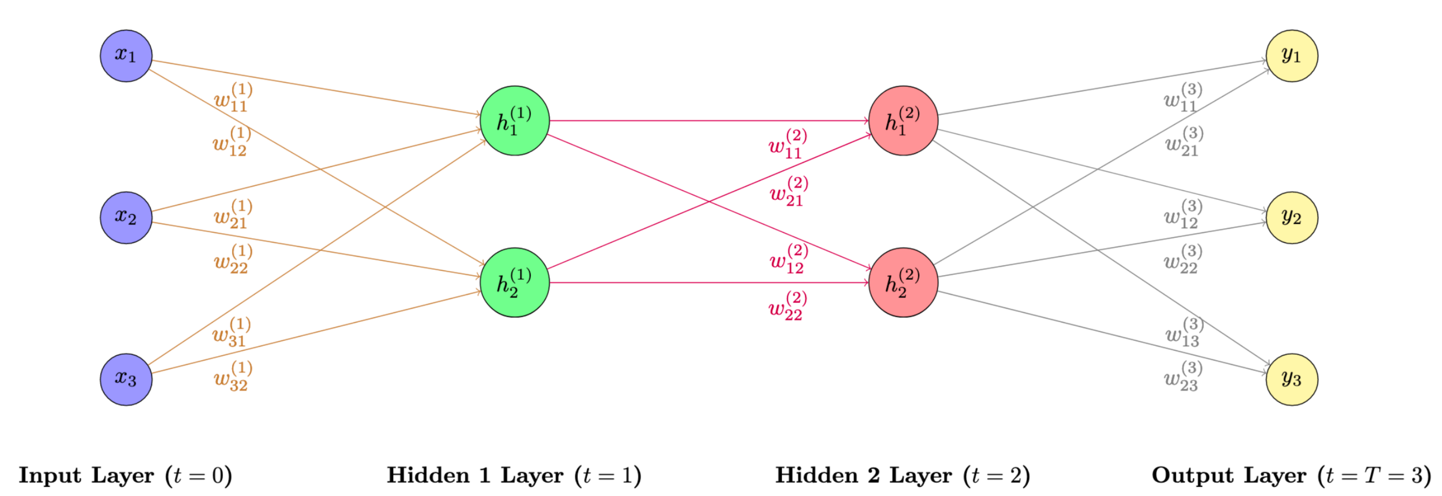
Perceptron

$$u_k^{(t)} = \sum_{i=1}^{M^{(t-1)}} w_{ij}^{(t-1)} o_i^{(t-1)} - b_j^{(t)} = \sum_{i=1}^{M^{(t-1)}} w_{ij}^{(t-1)} \sigma\left(u_i^{(t-1)}\right) - b_j^{(t)}$$

Inputs $(t-1)$ Integrated Input (t) Output (t)
Edge Weights $(t-1)$



Neural Network



Log-Linear Model on a Partially Ordered Set (poset)

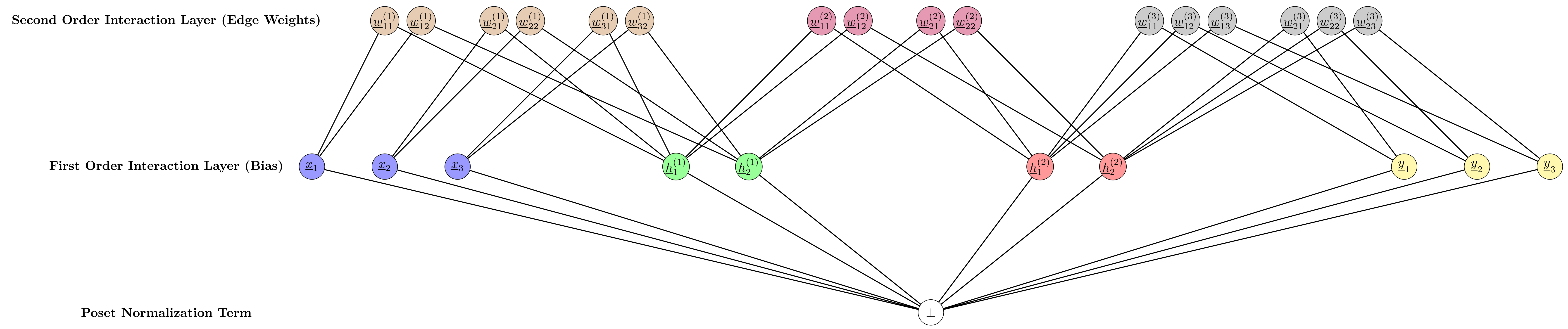
The log-linear model is defined over a partial order set (poset) $(S, \preccurlyeq)$
Dual coordinate system (θ, η) of a statistical manifold
- η Expectation parameter
- θ Natural parameter in the exponential family

$$\eta_\omega = \sum_{s \in \Omega} \mathbf{1}_{s \succcurlyeq \omega} p(s), \qquad \log p(\omega) = \sum_{s \in \Omega^+} \mathbf{1}_{s \preccurlyeq \omega} \theta_s - \psi(\boldsymbol{\theta}).$$

All parameters in the model are connected via the partition function given by

$$\psi(\boldsymbol{\theta}) = \log \sum_{\omega' \in \Omega^+} \exp\left[\sum_{\omega \in \Omega^+} \mathbf{1}_{\omega \preccurlyeq \omega'} \theta(\omega)\right] = -\theta(\perp).$$

Neural Network as a Partial Orders Structure



Representing Parameters as Partial Orders

Inputs: $\underline{x} \in \mathcal{X}$, Outputs: $\underline{y} \in \mathcal{Y}$

Hidden Nodes: $\underline{h} \in \mathcal{H}$, Edge Weights: $\underline{w} \in \mathcal{W}$

Integrated input: $u_k^{(t)} = \theta(\underline{n}_k)$

Edge Weight: $w_{ij}^{(t)} = \theta\left(\underline{n}_i^{(t-1)}\right) + \theta\left(\underline{w}_{ij}^{(t)}\right) + \theta\left(\underline{n}_j^{(t)}\right)$

Representing Inputs and Outputs

$$\hat{\eta}(\underline{x}_k) = \frac{\exp(\mathbb{E}[x_k])}{\sum_i \exp(\mathbb{E}[x_i])}, \qquad \hat{\eta}\left(\underline{y}_k\right) = \frac{\exp(\mathbb{E}[y_k])}{\sum_i \exp(\mathbb{E}[y_i])}.$$

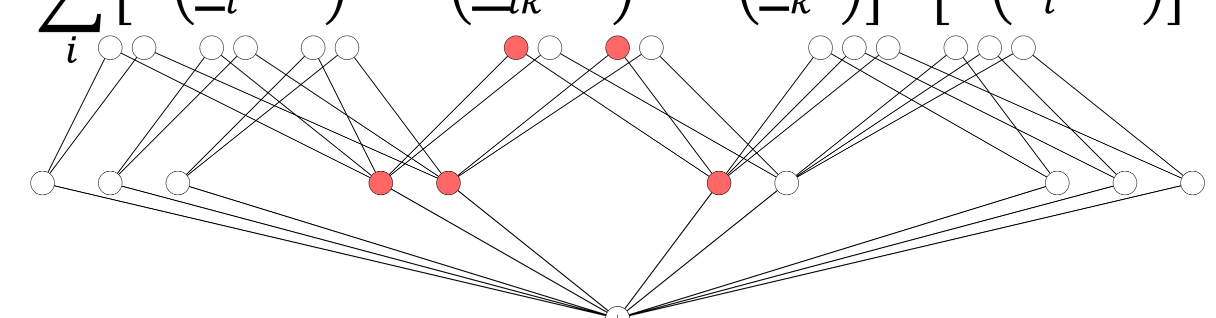
Expectation Step

Update probabilities for the input and output

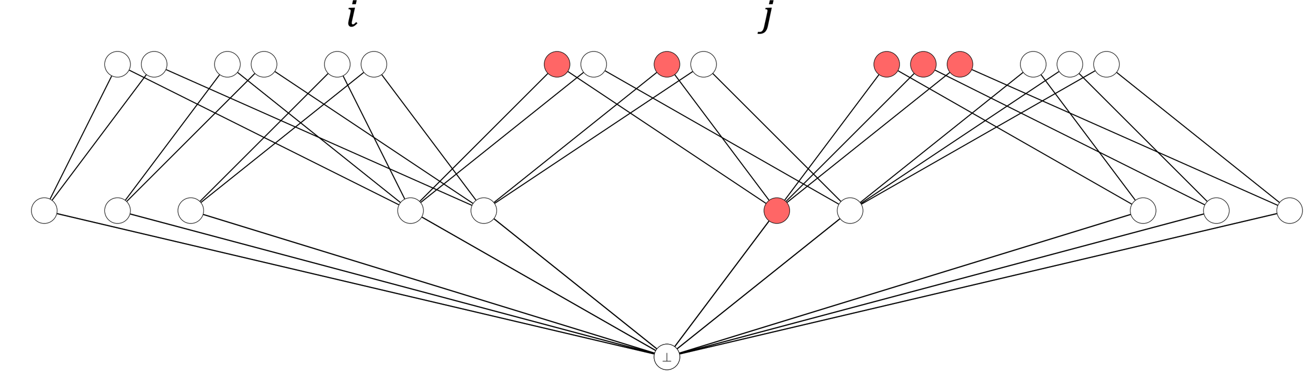
$$\hat{p}(\underline{x}_k) = \frac{\hat{\eta}(\underline{x}_k)}{\left[1 + \sum_j \exp\left[\hat{\theta}(\underline{w}_{kj}) + \hat{\theta}(\underline{h}_j)\right]\right]}, \hat{p}\left(\underline{y}_k\right) = \frac{\hat{\eta}\left(\underline{y}_k\right)}{\left[1 + \sum_j \exp\left[\hat{\theta}(\underline{w}_{kj}) + \hat{\theta}(\underline{h}_j)\right]\right]}$$

Forward Propagation on a Poset

$$u_k^{(t)} = \sum_i \left[\hat{\theta}\left(\underline{n}_i^{(t-1)}\right) + \hat{\theta}\left(\underline{w}_{ik}^{(t-1)}\right) + \hat{\theta}\left(\underline{n}_k^{(t)}\right)\right] \sigma\left[\hat{\theta}\left(n_i^{(t-1)}\right)\right] = \hat{\theta}\left(\underline{n}_k^{(t)}\right).$$



$$\hat{\eta}\left(\underline{h}_k^{(t)}\right) = \sum_i \hat{p}\left(w_{ik}^{(t-1)}; \hat{\theta}\right) + \sum_j \hat{p}\left(w_{kj}^{(t)}; \hat{\theta}\right) + \hat{p}\left(\underline{h}_k^{(t)}; \theta\right).$$



$$\hat{\eta}(\underline{w}_{ij}) = \hat{p}(\underline{w}_{ij}; \hat{\theta})$$

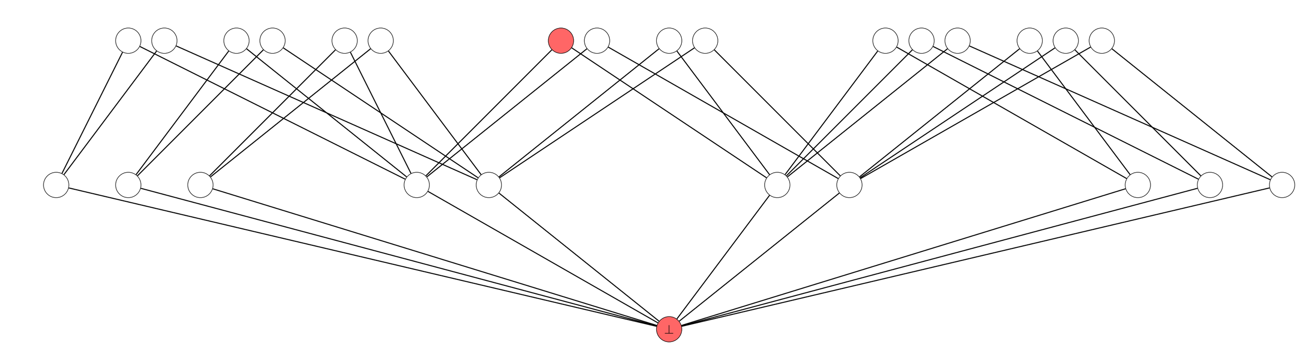
Optimization via EM-Algorithm

$$J(\boldsymbol{\theta}, \boldsymbol{\eta}) = -\mathbb{E}_{p(x,y)}[\log \hat{p}(\mathcal{H}, \mathcal{W} | \mathcal{X}, \mathcal{Y}; \boldsymbol{\theta})]$$

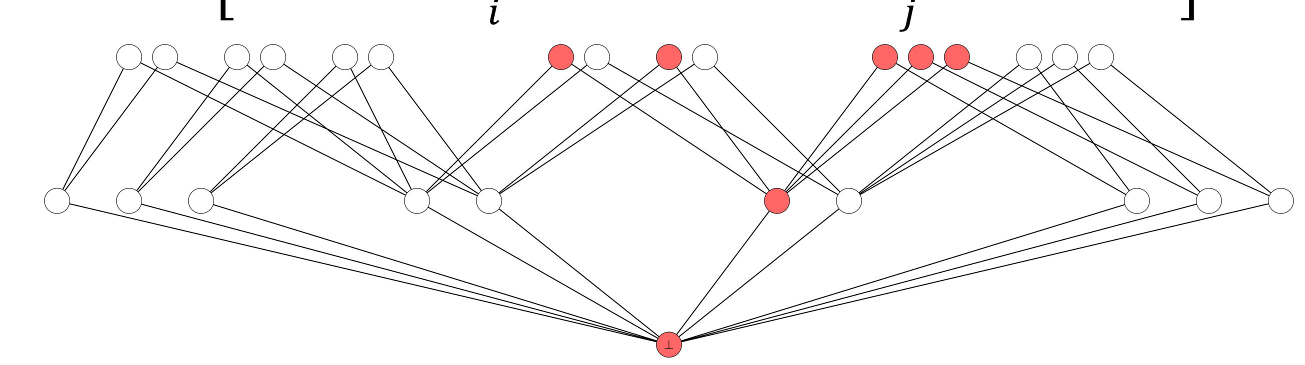
$$\boldsymbol{\eta}_{\text{next}} = \arg\min_{\boldsymbol{\eta}} J(\theta, \eta), \qquad \boldsymbol{\theta}_{\text{next}} = \arg\min_{\boldsymbol{\theta}} J(\theta, \eta).$$

Maximization Step

$$\hat{\theta}(\underline{w}_{ij}) = \log[\hat{\eta}(\underline{w}_{ij})] + \psi(\hat{\boldsymbol{\theta}})$$



$$\hat{\theta}\left(h_k^{(t)}\right) = \log\left[\hat{\eta}\left(h_k^{(t)}\right) - \sum_i \hat{p}\left(\underline{w}_{ik}^{(t-1)}; \hat{\theta}\right) - \sum_j \hat{p}\left(\underline{w}_{kj}^{(t)}; \hat{\theta}\right)\right] + \psi(\hat{\boldsymbol{\theta}})$$



References

- Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. **Tensor balancing on statistical manifold**, ICML 2017.
- Simon Luo and Mahito Sugiyama, **Bias-variance trade-off in hierarchical probabilistic models using higher-order feature interactions,** AAAI 2019.