Training Feature Discovery Networks

① So, given the causal model

$$\begin{tikzcd} & \theta \arrow[d] & & y^{(n)} \arrow[d] \\ x^{(n)} \arrow[r] & \pi^{(n)} \arrow[r] & o^{(n)} \arrow[r] & \ell^{(n)} \end{tikzcd} \qquad \left\} \begin{aligned} \ell^{(n)} &= L^{(n)}(o^{(n)}, y^{(n)}) \\ \pi^{(n)} &= f(x^{(n)}, \theta) \end{aligned} \right.$$

we want to compute $\frac{\partial \ell^{(n)}}{\partial \theta}$.

By the Chain Rule:

$$\frac{\partial \ell^{(n)}}{\partial \theta} = \frac{\partial \ell^{(n)}}{\partial o^{(n)}} \frac{\partial o^{(n)}}{\partial \theta} = \underbrace{\frac{\partial \ell^{(n)}}{\partial o^{(n)}}}_{②} \underbrace{\frac{\partial o^{(n)}}{\partial \pi^{(n)}}}_{②b} \frac{\partial \pi^{(n)}}{\partial \theta}$$

② This part should be straightforward to compute (assuming we chose some reasonable loss function).

e.g. $\frac{\partial}{\partial o^{(n)}} \ell^{(n)} = \frac{\partial}{\partial \pi^{(n)}} (y^{(n)} - o^{(n)})^2$

$$= -2(y^{(n)} - \pi^{(n)}) \quad \left(\text{for } \ell^{(n)} = L^{(n)}_{lin}(\pi^{(n)}, y^{(n)}) \right)$$

②b This part is also straightforward. For instance, with logistic regression, $o^{(n)} = \sigma(\pi^{(n)})$, thus:

$$\frac{\partial o^{(n)}}{\partial \pi^{(n)}} = \sigma'(\pi^{(n)}) = \sigma(\pi^{(n)})(1 - \sigma(\pi^{(n)}))$$

Training Feature Discovery Networks

(3) The challenge lies in computing $\dfrac{\partial}{\partial\theta}\pi^{(n)} = \dfrac{\partial}{\partial\theta} f(x^{(n)}, \theta)$

Let's suppose it's our "feature discovery" network:

θ: $\dot{W}$, $\ddot{W}$

$$\begin{tikzcd}
 & \dot{W} \arrow[d] & & \ddot{W} \arrow[d] & & y^{(n)} \arrow[d] \\
x^{(n)} \arrow[r] & \dot{\pi}^{(n)} \arrow[r] & \dot{x}^{(n)} \arrow[r] & \ddot{\pi}^{(n)} \arrow[r] & o^{(n)} \arrow[r] & \ell^{(n)}
\end{tikzcd}$$

(4) Recall that $\theta = \{\ddot{w}_{ij} \in \ddot{W}\} \cup \{\dot{w}_{ij} \in \dot{W}\}$,

so to compute $\dfrac{\partial}{\partial\theta}\ddot{\pi}^{(n)}$, we need to compute:

(a) for each $\ddot{w}_{ij} \in \ddot{W}$:

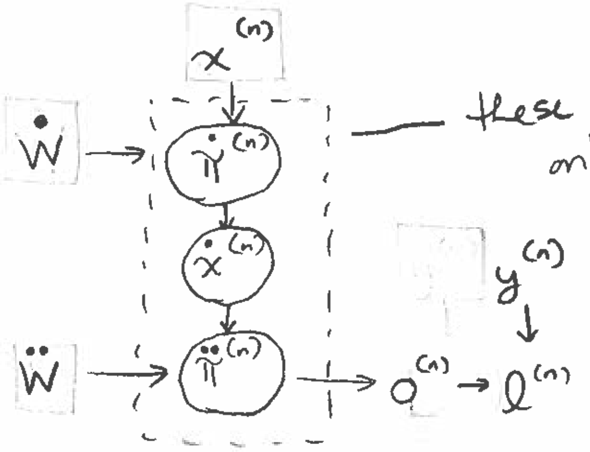
$$\frac{\partial}{\partial \ddot{w}_{ij}} \ddot{\pi}^{(n)} = \frac{\partial}{\partial \ddot{w}_{ij}} \dot{x}^{(n)} \ddot{W} = \dot{x}_i^{(n)}$$

(b) for each $\dot{w}_{ij} \in \dot{W}$:

$$\frac{\partial}{\partial \dot{w}_{ij}} \ddot{\pi}^{(n)}$$

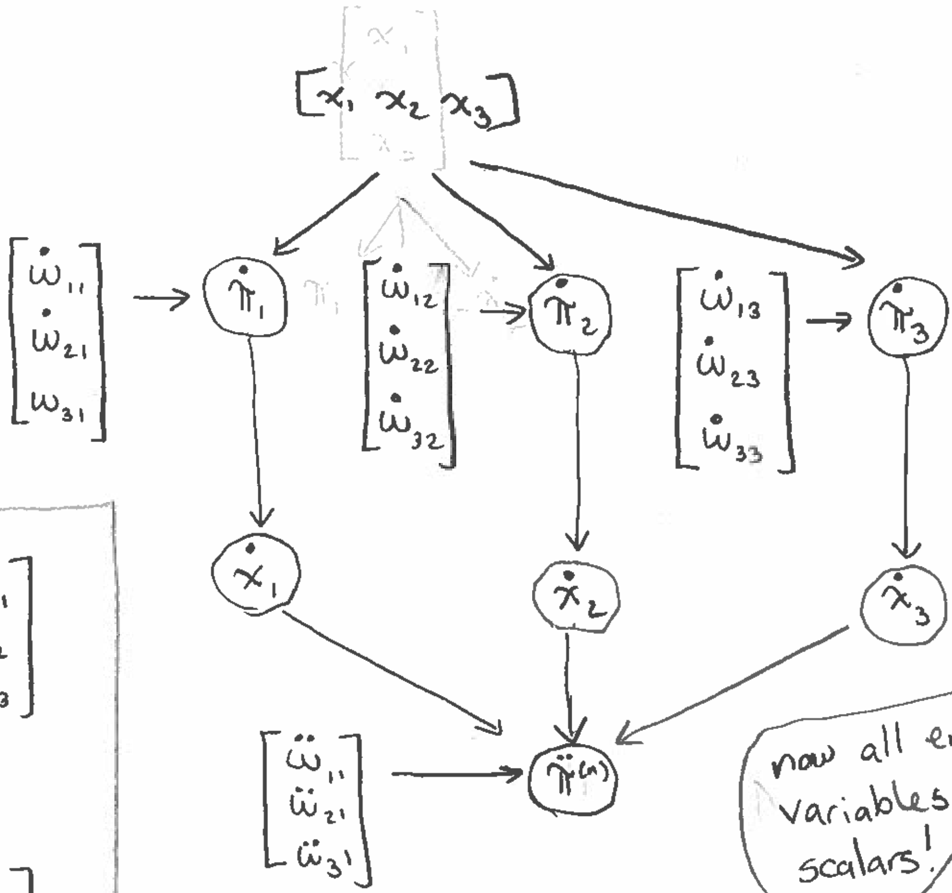
Training Feature Discovery Networks

⑤ Sounds like a job for the Chain Rule of Partial Derivatives, but there's one issue:



these are vectors, and we've only shown how to use the Chain Rule on a causal diagram when these are scalars.

⑥ No problem though, let's just explicitly represent the components of these vectors and matrices:



$$\dot{\pi}_j \leftarrow \begin{bmatrix}\dot{w}_{1j}\\ \dot{w}_{2j}\\ \dot{w}_{3j}\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$$

$$\dot{x}_j \leftarrow a(\dot{\pi}_j)$$

$$\ddot{\pi}^{(n)} \leftarrow \begin{bmatrix}\ddot{w}_{11}\\ \ddot{w}_{21}\\ \ddot{w}_{31}\end{bmatrix} \cdot \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}$$

now all endogenous variables are scalars!

Training Feature Discovery Networks

⑦ Now we can compute $\frac{\partial \ddot{\pi}^{(n)}}{\partial \dot{\omega}_{ij}}$ by repeated application of the Chain Rule:

$$\frac{\partial \ddot{\pi}^{(n)}}{\partial \dot{\omega}_{ij}} = \frac{\partial \ddot{\pi}^{(n)}}{\partial \dot{x}_j} \cdot \frac{\partial \dot{x}_j}{\partial \dot{\omega}_{ij}}$$

[$\dot{x}_j$ separates $\ddot{\pi}^{(n)}$ from $\dot{\omega}_{ij}$, so Chain Rule applies]

$$= \frac{\partial \ddot{\pi}^{(n)}}{\partial \dot{x}_j} \cdot \frac{\partial \dot{x}_j}{\partial \dot{\pi}_j} \cdot \frac{\partial \dot{\pi}_j}{\partial \dot{\omega}_{ij}}$$

[$\dot{\pi}_j$ separates $\dot{x}_j$ from $\dot{\omega}_{ij}$, so Chain Rule applies]

$$= \left(\frac{\partial}{\partial \dot{x}_j} \dot{x}^{(n)} \begin{bmatrix} \ddot{\omega}_{11} \\ \ddot{\omega}_{21} \\ \ddot{\omega}_{31} \end{bmatrix}\right) \left(\frac{\partial}{\partial \dot{\pi}_j} a(\dot{\pi}_j)\right) \left(\frac{\partial}{\partial \dot{\omega}_{ij}} \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} \dot{\omega}_{1j} \\ \dot{\omega}_{2j} \\ \dot{\omega}_{3j} \end{bmatrix}\right)$$

$$= \ddot{\omega}_{j1} \cdot a'(\dot{\pi}_j) \cdot x_i$$

this is just the standard derivative of the ReLU function.

⑧ Putting this together with the result from ①:

$$\frac{\partial \ell^{(n)}}{\partial \dot{w}_{ij}} = \frac{\partial \ell^{(n)}}{\partial o^{(n)}} \cdot \frac{\partial o^{(n)}}{\partial \ddot{\pi}^{(n)}} \cdot w_j \, a'(\dot{\pi}_j^{(n)}) x_i^{(n)}$$

and
$$\frac{\partial \ell^{(n)}}{\partial \ddot{w}_{ij}} = \frac{\partial \ell^{(n)}}{\partial o^{(n)}} \frac{\partial o^{(n)}}{\partial \ddot{\pi}^{(n)}} \dot{x}_i^{(n)}$$

So we can compute the partial derivative of the loss with respect to any of our weights. Thus we can use gradient descent to compute the weights that minimize our loss.