TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

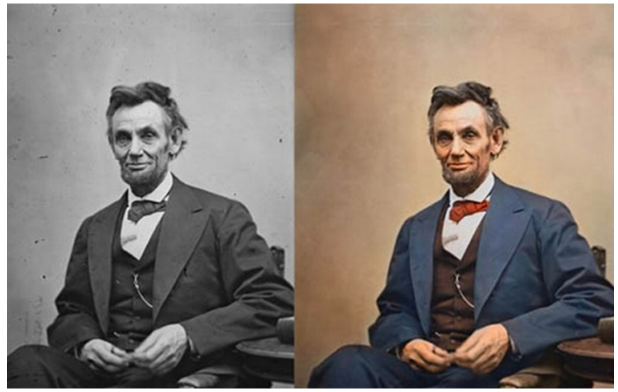
Deep Graphical Models I

Exponential Softmax

Sufficient Statistics

Belief Propagation

Consider Colorization



x is a black and white image.

y is a color image drawn from $\mathrm{Pop}(y|x)$.

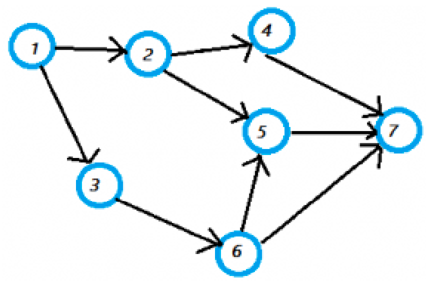
$\hat{y}$ is an arbitrary color image.

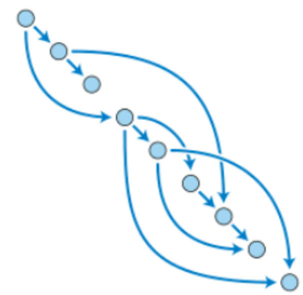
$Q_\Phi(\hat{y}|x)$ is the probability that model Φ assigns to the color image y given black and white image x.

Cross Entropy Training

$$\Phi^* = \underset{\Phi}{\mathrm{argmin}} \quad E_{(x,y)\sim \mathrm{Pop}} - \log Q_\Phi(y|x)$$

Auto-Regressive Models are Tractable





An auto-regressive model is locally normalized.

$$Q_f(\hat{y}) = \prod_i Q_f(\hat{y}[i] \mid \hat{y}[\mathrm{Parents}(i)])$$

$$Q_f(\hat{y}[i] \mid \hat{y}[\mathrm{Parents}(i)]) = \operatorname*{softmax}_{\tilde{y}} f(\tilde{y}|\hat{y}[\mathrm{Parents}(i)])$$

There are exponentially many possible values for $\hat{y}$ but each softmax is over a tractable-sized set.

General Markov Random Fields (MRFs) are More Challenging

We can run a CNN with parameters Φ on the black and white image x to get a Markov random field (MRF) $f_\Phi(x)$ on possible color images.

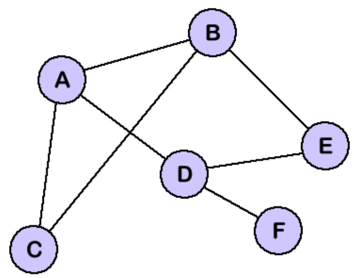
The MRF $f_\Phi(x)$ will determine the probabilities $Q_\Phi(\hat{y}|x) = Q_{f_\Phi(x)}(\hat{y})$.

Markov Random Fields (MRFs)

$\hat{y}[i]$ is the color value of pixel i in image $\hat{y}$.

$\hat{y}[(i,j)]$ is the pair $(\hat{y}[i], \hat{y}[j])$ for neighboring pixels i and j.

Markov Random Fields (MRFs)



$$f(\hat{y}) = \sum_{i \in \text{Nodes}} f[i, \hat{y}[i]] \; + \sum_{(i,j) \in \text{Edges}} f[(i,j), \hat{y}[(i,j)]]$$

Node Potentials Edge Potentials

Exponential Softmax

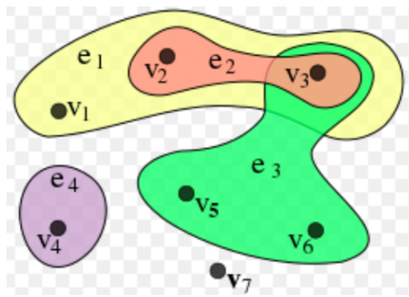
$$Q_f(\hat{y}) = \operatorname*{softmax}_{\hat{y}} \; f(\hat{y})$$

$$Q_f(\hat{y}) = \frac{1}{Z}\, e^{f(\hat{y})} \qquad Z = \sum_{\hat{y}} \; e^{f(\hat{y})}$$

$$f(\hat{y}) = \sum_{i \in \mathrm{Nodes}} \; f[i, \hat{y}[i]] \; + \sum_{(i,j) \in \mathrm{Edges}} \; f[(i,j), \hat{y}[(i,j)]]$$

Hyper-Graphs: More General and More Concise

A hyper-edge is a subset of nodes.



$$f(\hat{y}) = \sum_{i \in \mathrm{Nodes}} f[i, \hat{y}[i]] + \sum_{(i,j) \in \mathrm{Edges}} f[(i,j), \hat{y}[(i,j)]]$$

$$f(\hat{y}) = \sum_{\alpha \in \mathrm{HyperEdges}} f[\alpha, \hat{y}[\alpha]]$$

Back-Propagation Through An Exponential Softmax

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{(x,y)\sim\mathrm{Pop}} - \log Q_\Phi(y|x)$$

$$= \operatorname*{argmin}_{\Phi} \; E_{(x,y)\sim\mathrm{Pop}} - \log Q_{f_\Phi(x)}(y)$$

We need to back-propagate through the softmax to get f.grad.

f is a tensor containing the numbers $f[\alpha, \tilde{y}]$ where $\tilde{y}$ is a possible value of $\hat{y}[\alpha]$.

$$f.\mathrm{grad}[\alpha, \tilde{y}] = \frac{-\partial \log Q_f(y)}{\partial f[\alpha, \tilde{y}]}$$

Back-Propagation Through An Exponential Softmax

$$\begin{aligned}\mathrm{loss}(f, y) &= -\ln\left(\frac{1}{Z(f)}\, e^{f(y)}\right) \\ &= \ln Z(f) - f(y)\end{aligned}$$

$$f.\mathrm{grad}[\alpha, \tilde{y}] = \left(\frac{1}{Z}\sum_{\hat{y}} e^{f(\hat{y})}\,(\partial f(\hat{y})/\partial f[\alpha, \tilde{y}])\right) - (\partial f(y)/\partial f[\alpha, \tilde{y}])$$

Back-Propagation Through An Exponential Softmax

$$\begin{aligned}
f.\mathrm{grad}[\alpha,\tilde{y}] &= \left(\frac{1}{Z}\sum_{\hat{y}} e^{f(\hat{y})}\,(\partial f(\hat{y})/\partial f[\alpha,\tilde{y}])\right) - (\partial f(y)/\partial f[\alpha,\tilde{y}]) \\
&= \left(\sum_{\hat{y}} Q_f(\hat{y})\,(\partial f(\hat{y})/\partial f[\alpha,\tilde{y}])\right) - (\partial f(y)/\partial f[\alpha,\tilde{y}]) \\
&= E_{\hat{y}\sim Q_f}\mathbb{1}[\hat{y}[\alpha]=\tilde{y}] - \mathbb{1}[y[\alpha]=\tilde{y}] \\
&= P_{\hat{y}\sim Q_f}(\hat{y}[\alpha]=\tilde{y}) - \mathbb{1}[y[\alpha]=\tilde{y}]
\end{aligned}$$

Sufficient Statistics

$$f.\mathrm{grad}[\alpha, \tilde{y}] = P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) - \mathbb{1}[y[\alpha] = \tilde{y}]$$

To compute $f.\mathrm{grad}$ it suffices to compute $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y})$.

By (minor) abuse of terminology we will call the quantities $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y})$ the **sufficient statistics** for f.

We now focus on computing the sufficient statistics for a given MRF f.

An Aside: Features and Weights

The indicators $\mathbb{1}[\hat{y}[\alpha] = \tilde{y}]$ form a 0-1 feature vector $\Psi(\hat{y})$.

The tensor $f[\alpha,\ \tilde{y}]$ forms a weight vector.

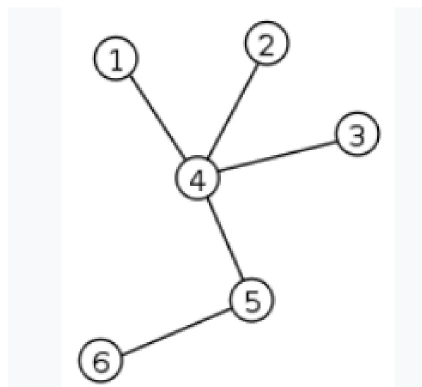
$$
\begin{aligned}
f(\hat{y}) &= \sum_{\alpha} f[\alpha, \hat{y}[\alpha]] \\
&= \sum_{\alpha,\tilde{y}} f[\alpha, \tilde{y}]\mathbb{1}[\alpha, \hat{y}[\alpha] = \tilde{y}] \\
&= f^{\top}\Psi(\hat{y})
\end{aligned}
$$

An Aside: Features and Weights

The sufficient statistics $P_{\hat{y}\sim Q_f}(\hat{y}[\alpha] = \tilde{y})$ are just the expected value of the features under the distribution defined by the MRF.

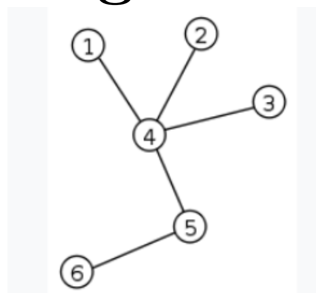
Belief Propagation

$$f.\mathrm{grad}[\alpha, \tilde{y}] = P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) - \mathbb{1}[y[\alpha] = \tilde{y}]$$



For trees we can compute $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y})$ exactly by message passing, aka, belief propagation.

Message Passing

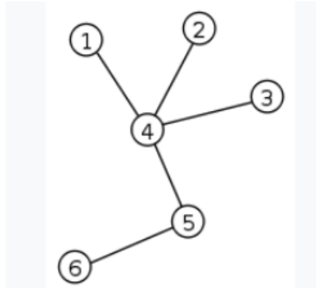


For each edge (i, j) there is a message $Z_{i \to j}$ and a message $Z_{j \to i}$.

Each message is assigns a weight to each node value of the target node.

$Z_{j \to i}[\tilde{y}]$ is the partition function for the subtree attached to i through j and with $\hat{y}[i]$ restricted to $\tilde{y}$.

Message Passing

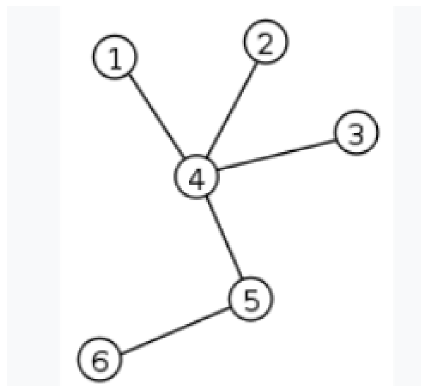


$$Z(i,\tilde{y}) \doteq \sum_{\hat{y}:\ \hat{y}[i]=\tilde{y}} e^{f(\hat{y})}$$

$$= e^{f[i,\tilde{y}]} \left(\prod_{j \in N(i)} Z_{j \to i}[\tilde{y}] \right)$$

$$P_{\hat{y} \sim Q_f}(\hat{y}[i] = \tilde{y}) = Z(i,\tilde{y})/Z, \qquad Z = \sum_{\tilde{y}} Z(i,\tilde{y})$$

Message Passing



$$Z_{j \to i}[\tilde{y}] = \sum_{\tilde{y}'} e^{f[j,\tilde{y}'] + f[\{j,i\},\{\tilde{y}',\tilde{y}\}]} \left(\prod_{k \in N(j),\, k \neq i} Z_{k \to j}[\tilde{y}'] \right)$$

Message Passing

$$
\begin{aligned}
Z(\{i,j\},\tilde{y}) &\doteq \sum_{\hat{y}:\ \hat{y}[\{i,j\}]=\tilde{y}} e^{f(\hat{y})} \\
&= e^{f[i,\tilde{y}[i]]+f[j,\tilde{y}[j]]+f[\{i,j\},\tilde{y}]} \\
&\quad \prod_{k\in N(i),\ k\neq j} Z_{k\to i}[\tilde{y}[i]] \\
&\quad \prod_{k\in N(j),\ k\neq i} Z_{k\to j}[\tilde{y}[j]] \\
P_{\hat{y}\sim Q_f}(\hat{y}[\{i,j\}]=\tilde{y}) &= Z(\{i,j\},\tilde{y})/Z
\end{aligned}
$$

Loopy BP

Message passing is also called belief propagation (BP).

In a graph with cycles it is common to do **Loopy BP**.

This is done by initializing all message $Z_{i \to j}[\tilde{y}] = 1$ and then repeating (until convergence) the updates

$$Z_{j \to i}[\tilde{y}] = \sum_{\tilde{y}'} e^{f[j,\tilde{y}'] + f[\{j,i\},\{\tilde{y}',\tilde{y}\}]} \left(\prod_{k \in N(j),\, k \neq i} Z_{k \to j}[\tilde{y}'] \right)$$

END