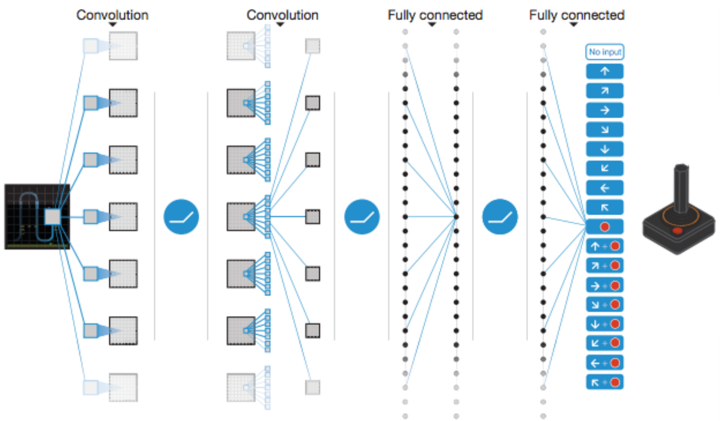
TTIC 31230, Fundamentals of Deep Learning

David McAllester, April 2017

Deep Graphical Models

Review: Deep Q Networks (DQN)

Q-network $Q_\Theta(s, a)$



$$\Theta \mathrel{-}= \eta \nabla_\Theta \left(Q_\Theta(s_t, a_t) - R_t\right)^2$$

$$\pi(s) \leftarrow \underset{a}{\text{argmax}}\ Q_\Theta(s, a)$$

Review: Asynchronous Advantage Actor-Critic (A3C)

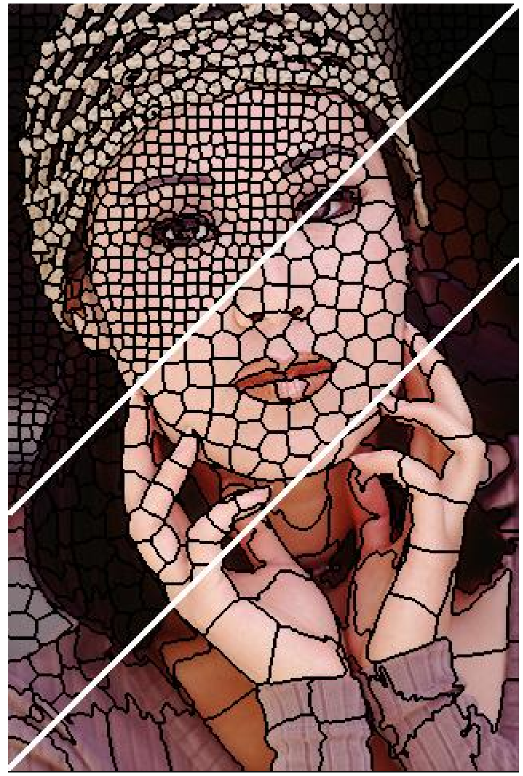
Policy network $\pi_\Phi(a|s)$
State value network $V_\Psi(s)$

$$\Phi \mathrel{+}= \eta \left(\nabla_\Phi \ \ln \pi_\Phi(a_t|s_t)\right) \left(R_t - V_\Psi(s_t)\right)$$

$$\Psi \mathrel{-}= \eta \nabla_\Psi \ \left(V_\Psi(s_t) - R_t\right)^2$$

Deep Graphical Models

Image Segmentation with Superpixels



[Achanta et al.]

We want to assign each superpixel a semantic label. Maybe "face", "hand", or "hat". (More typically "person", "car", "road", "building" or "background".)

Exponential Softmax

If we have K superpixels and N possible semantic labels we have N^K possible semantic labelings.

We will define a probability distribution over the semantic labelings with an **exponential softmax**.

Segmentation Features

We have K superpixels and N possible semantic labels.

We define KN features $U_{k,n}$ where $U_{k,n} = 1$ if segment k has label n and 0 otherwise.

If each superpixel has D neightbors we define $DK/2$ features $B_{k,k'}$ where $B_{k,k'} = 1$ (with $k < k'$) if neighboring segments k and k' have different labels, and 0 otherwise.

Let $\Phi(y)$ be the feature vector of segmentation y.

Scoring by Weighting Features

We can score a segmentation y by providing a vector of weights for the features.

$$s_w(y) = w \cdot \Phi(y)$$

We can then define an exponential softmax over the semantic assignments.

$$P_w(y) = \operatorname*{softmax}_{y} s_w(y)$$

Weight Networks

We will consider a weight network $W_\Theta(x)$ which assigns weights to the features of y.

$$s_\Theta(y|x) = W_\Theta(x) \cdot \Phi(y)$$

$$P_\Theta(y|x) = \operatorname*{softmax}_y \; s_\Theta(y|x)$$

Cross Entropy Training

$$\Theta^* = \underset{\Theta}{\text{argmin}}\ \mathrm{E}_{(x,y)\sim D}\left[-\ln P_\Theta(y|x)\right]$$

$$P_\Theta(y|x) = \underset{y}{\text{softmax}}\ s_\Theta(y|x)$$

Note that the same equations apply whether y is drawn from a small set or an exponentially large set.

It Suffices to Compute w.grad

$$w = W_\Theta(x)$$

$$P = \mathrm{softmax}\, s(\cdot|w)$$

$$\ell = -\ln P(y)$$

$$\Theta.\mathrm{grad} = w.\mathrm{grad}\;\; \nabla_\Theta W(x,\Theta)$$

Computing w.grad

$$\ell(y|w) = -\ln P(y|w)$$

$$P(y|w) = \frac{1}{Z(w)} e^{w \cdot \Phi(y)}$$

$$Z(w) = \sum_y e^{w \cdot \Phi(y)}$$

$$\ell(y|w) = \ln Z(w) - w \cdot \Phi(y)$$

$$\nabla_w \ell(y|w) = \nabla_w \ln Z(w) - \Phi(y)$$

Negative Sampling

$$
\begin{aligned}
-w.\text{grad} &= \Phi(y) - \nabla_w \ln Z(w) \\
&= \Phi(y) - \frac{1}{Z}\sum_{y'} e^{w\cdot\Phi(y')}\;\Phi(y') \\
&= \Phi(y) - \sum_{y'} \left(\frac{1}{Z} e^{w\cdot\Phi(y')}\right)\;\Phi(y') \\
&= \Phi(y) - \mathrm{E}_{y'\sim P(\cdot|w)}\left[\Phi(y')\right]
\end{aligned}
$$

We can estimate $\mathrm{E}_{y'\sim P(\cdot|w)}\left[\Phi(y')\right]$ by sampling. This is called **negative sampling**.

We move toward $\Phi(y)$ and away from $\Phi(y')$.

Monte Carlo Markov Chain (MCMC) Sampling

Metropolis Algorithm

Assume that each y has a set of N "neighbors" where the neighbor relation is symmetric.

Pick an initial y then repeat for the **mixing time**.

1. pick a neighbor y' of y uniformly at random.

2. If $s(y'|w) > s(y|w)$ update $y = y'$

3. If $s(y'|w) \leq s(y|w)$ then update $y = y'$ with probability $e^{-\Delta s}$, $\Delta s = s(y|w) - s(y'|w)$.

Markov Processes and Stationary Distributions

A Markov process is a process defined by a fixed state transition probability $P(y'|y) = M_{y',y}$.

Let P^t the probability distribution for time t.

$$P^{t+1} = MP^t$$

If every state can be reached form every state (ergodic process) then P^t converges to a unique **stationary distribution** P^∞

$$P^\infty = MP^\infty$$

Correctness of Metropolis

To verify that the Metropolis process has the correct stationary distribution we simply verify that $MP = P$ where P is the desired distribution.

This can be done by checking that under the desired distribution the flow from y to y' equals the flow from y' to y (**detailed balance**).

For $s(y) \geq s(y')$

$$\mathrm{flow}(y' \rightarrow y) = \frac{1}{Z} e^{s(y')} \frac{1}{N}$$

$$\mathrm{flow}(y \rightarrow y') = \frac{1}{Z} e^{s(y)} \frac{1}{N} e^{-\Delta s} = \frac{1}{Z} e^{s(y')} \frac{1}{N}$$

Detailed balance is not required in general.

Negative Sampling with MCMC

Sample $y' \sim P(\cdot|w)$ using MCMC.

$$-w.\mathrm{grad} \leftarrow \Phi(y) - \Phi(y')$$

We move toward $\Phi(y)$ and away from $\Phi(y')$.

Gibbs Sampling

The Metropolis algorithm wastes time by rejecting proposed moves.

Gibbs sampling avoids this move rejection.

Gibbs sampling applies when y is a tuple $(y_1, \ldots, y_K)$.

In semantic segmentation y_k is the class of superpixel k.

Gibbs Sampling

Initialize y

Repeat for the **mixing time**:

1. Select a component k

2. Update $y_k = y'_k$ where y'_k is drawn from

$$P(y'_k \,|y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K)$$

For the models we are considering $P(y'_k \,|y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K)$ is easily computed (the partition function Z cancels).

Correctness Proof

$P(y)$ is a stationary distribution of Gibbs Sampling:

$$P^{t+1}(y_1, \ldots, y_K) = \frac{1}{K} \sum_k \begin{array}{l} P^t(y_k \,|y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K) \\ P^t(y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K) \end{array}$$

$$= \frac{1}{K} \sum_k P^t(y_1, \ldots, y_k)$$

$$= P^t(y_1, \ldots, y_K)$$

Negative Sampling with Gibbs

Sample $y' \sim P(\cdot|w)$ using MCMC.

$$-w.\text{grad} \leftarrow \Phi(y) - \Phi(y')$$

Pseudolikelihood

In Pseudolikelihood we replace the objective $\ln P(y|w)$ with the objective $\ln \tilde{P}(y|w)$ where

$$\tilde{P}(y|w) = \prod_k P(y'_k \,|y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K, w)$$

As in Gibbs sampling, we note that these probabilities are easily computed.

$$-w.\mathrm{grad} \leftarrow \nabla_w \ln \tilde{P}(y|w)$$

Pseudolikelihood

Algorithm:

$$\Theta^* = \underset{\Theta}{\text{argmin}} \;\; \mathrm{E}_{(x,y)\sim D} \left[-\ln \tilde{P}(y|W(x;\Theta)) \right]$$

Theorem:

$$\underset{Q(y|x)}{\text{argmin}} \; \mathrm{E}_{(x,y)\sim D} \left[-\ln \tilde{Q}(y|x) \right] = D(y|x)$$

This is called **consistency**.

Proof of Consistency I

We have

$$\min_{Q(y|x)} \; \mathrm{E}_{(x,y)\sim D}\left[-\ln \tilde{Q}(y|x)\right] \leq \mathrm{E}_{(x,y)\sim D}\left[-\ln \tilde{D}(y|x)\right]$$

If we can show

$$\min_{Q(y|x)} \; \mathrm{E}_{(x,y)\sim D}\left[-\ln \tilde{Q}(y|x)\right] \geq \mathrm{E}_{(x,y)\sim D}\left[-\ln \tilde{D}(y|x)\right]$$

Then the minimizer (the argmin) is D as desired.

Proof of Consistency II

We will prove the case of $K = 2$.

Consider **unrelated** distributions $Q_1(y_1|y_2, x)$ and $Q_2(y_2|y_1, x)$.

$$\min_{Q(y|x)} \mathrm{E}_{(x,y)\sim D}\left[-\ln Q(y_1|y_2, x)\; Q(y_2|y_1, x)\right]$$

$$\geq \min_{Q_1,Q_2} \mathrm{E}_{(x,y)\sim D}\left[-\ln Q_1(y_1|y_2, x)\; Q_2(y_2|y_1, x)\right]$$

$$= \min_{Q_1} \mathrm{E}_{(x,y)\sim D}\left[-\ln Q_1(y_1|y_2, x)\right] + \min_{Q_2} \mathrm{E}_{(x,y)\sim D}\left[-\ln Q_2(y_2|y_1, x)\right]$$

$$= \mathrm{E}_{(x,y)\sim D}\left[-\ln D(y_1|y_2, x)\right] + \mathrm{E}_{(x,y)\sim D}\left[-\ln D(y_2|y_1, x)\right]$$

$$= \mathrm{E}_{(x,y)\sim D}\left[-\ln \tilde{D}(y|x)\right]$$

Contrastive Divergence

Algorithm (CDk): Run k steps of MCMC for $P_\Theta(y|x)$ **starting from** y to get y'.

$$-w.\mathrm{grad} \leftarrow \Phi(y) - \Phi(y')$$

Theorem: If $P_\Theta(y|x) = D(y|x)$ then

$$\mathrm{E}_{(x,y)\sim D}\left[\Phi(y) - \Phi(y')\right] = 0$$

Here we can take $k = 1$ — **no mixing time required**.

Handling Task Loss

Different Costs for Different Errors

In inexpensive cancer screening we want **high recall** of cancers but can tolerate **low precision**.

In other words, the cost of a **false negative** is generally considered higher than the cost of a **false positive**.

Intersection over Union

In visual detection problems one is typically evaluated by **intersection over union**.

$$\mathbf{IOU} = \frac{|\text{true positives} \cap \text{false positives}|}{|\text{true positives} \cup \text{false positives}|} = \frac{P - FN}{P + FP}$$

$$\frac{\partial IOU}{\partial \text{FP}} = \frac{-(P - FN)}{(P + FP)^2} = \frac{-\text{IOU}}{P + FP}$$

$$\frac{\partial IOU}{\partial \text{FN}} = \frac{-1}{P + FP}$$

Generic Task Loss

We can consider an arbitrary loss function $L(y, \hat{y})$ assigning a loss when the true label is y and the system guesses $\hat{y}$.

For example, If y and $\hat{y}$ are segmentations then $\hat{y}$ typically has many errors.

We can then assign a cost $C_{i,j}$ for labeling a pixel i when the true label is j.

Label Adjustment

$$s_\Theta(y|x) \;=\; W_\Theta(x) \cdot \Phi(y)$$

$$\hat{y}_\Theta(x) \;=\; \underset{y}{\text{argmax}}\;\; s_\Theta(y|x)$$

$$\tilde{y}_{\Theta,\epsilon}(x,y) \;=\; \underset{\tilde{y}}{\text{argmax}}\;\; s_\Theta(y|x) - \epsilon L(y,\tilde{y}) \quad \textbf{(adjusted label)}$$

$$-w.\text{grad} \leftarrow \Phi(\tilde{y}_\epsilon) - \Phi(\hat{y})$$

Theorem: For a continuous and smooth data distribution D

$$-\nabla_w \, \mathrm{E}_{(x,y)\sim D}\left[L(y,\hat{y}_w)\right] \;=\; \lim_{\epsilon \rightarrow 0}\; \frac{1}{\epsilon}\, \mathrm{E}_{(x,y)\sim D}\left[\Phi(\tilde{y}_{w,\epsilon}) - \Phi(\hat{y}_w)\right]$$

END