TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2019

Exponential Softmax

Distributions on Exponentially Large Sets

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

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{y\sim \mathrm{Pop}} \; - \ln \; P(y)$$

The structured case: $y \in \mathcal{Y}$ where $\mathcal{Y}$ is discrete but iteration over $\hat{y} \in \mathcal{Y}$ is infeasible.

Semantic Segmentation

We want to assign each pixel to one of C semantic classes.

For example "person", "car", "building", "sky" or "other".

Constructing a Graph

We construct a graph whose nodes are the pixels and where there is an edges between each pixel and its four nearest neighboring pixels.

$$\begin{array}{ccccc}
 & & j(i,\mathrm{up}) & & \\
 & & | & & \\
j(i,\mathrm{left}) & - & i & - & j(i,\mathrm{right}) \\
 & & | & & \\
 & & j(i,\mathrm{down}) & &
\end{array}$$

Labeling the Nodes of the Graph

$\hat{y}$ assigns a semantic class $\hat{y}[i]$ to each node (pixel) i.

We assign a score to $\hat{y}$ by assigning a score to each node and each edge of the graph.

$$s(\hat{y}) = \underbrace{\sum_{i \in \mathrm{Nodes}} s_n[i, \hat{y}[i]]}_{\text{Node Scores}} + \underbrace{\sum_{\langle i,j\rangle \in \mathrm{Edges}} s_e[\langle i,j\rangle, \hat{y}[i], \hat{y}[j]]}_{\text{Edge Scores}}$$

Computing the Node and Edge Tensors

For input x we use a network to compute the score tensors.

$$s_n[I, C] \;=\; f^n_\Phi(x)$$

$$s_e[E, C, C] \;=\; f^e_\Phi(x)$$

Exponential Softmax

$$\text{for } \hat{y} \quad s(\hat{y}) = \sum_i \; s_n[i, \hat{y}[i]] + \sum_{\langle i, j\rangle \in \text{Edges}} \; s_e[\langle i, j\rangle, \hat{y}[i], \; \hat{y}[j]]$$

$$\text{for } \hat{y} \;\; P_s(\hat{y}) = \text{softmax}_{\hat{y}} \;\; s(\hat{y}) \;\; \text{all possible } \hat{y}$$

$$\mathcal{L} = -\ln P_s(y) \quad \text{gold label (training label) } y$$

Exponential Softmax is Typically Intractable

$\hat{y}$ assigns a label $\hat{y}[i]$ to each node i.

$s(\hat{y})$ is defined by a sum over node and edge tensor scores.

$P_s(\hat{y})$ is defined by an exponential softmax over $s(\hat{y})$.

Computing Z in general is #P hard (there is an easy direct reduction from SAT).

Compactly Representing Scores on Exponentially Many Labels

The tensor $s_n[I, C]$ holds IC scores.

The tensor $s_e[E, C, C]$ holds EC^2 scores where e ranges over edges $\langle i, j \rangle \in \text{Edges}$.

Back-Propagation Through Exponential Softmax

$$\begin{aligned} s_n[I,C] &= f^n_\Phi(x) \\ s_e[E,C,C] &= f^e_\Phi(x) \end{aligned}$$

$$s(\hat{y}) = \sum_i s_n[i, \hat{y}[i]] + \sum_{\langle i,j\rangle \in \mathrm{Edges}} s_e[\langle i,j\rangle, \hat{y}[i],\ \hat{y}[j]]$$

$$P_s(\hat{y}) = \operatorname*{softmax}_{\hat{y}}\ s(\hat{y}) \ \ \text{all possible } \hat{y}$$

$$\mathcal{L} = -\ln P_s(y) \quad \text{gold label } y$$

We want the gradients $s_n.\mathrm{grad}[I,C]$ and $s_e.\mathrm{grad}[E,C,C]$.

END