Sampled Softmax with Random Fourier Features

Ankit Singh Rawat, Jiecao Chen, Felix Xinnan Yu, Ananda Theertha Suresh, Sanjiv Kumar

Google Research New York

Problem

- Computational cost of training with softmax based cross-entropy loss scales linearly with number of classes.
- Infeasible for many real-life applications.
- **Sampled softmax** computes loss based on a small subset of sampled negative classes.
- Sampling distribution plays a crucial role in the training speed and the quality of the final model.

**Objective:** Design provably accurate sampling methods with low computational cost.

Background

- Let \( \theta \) denote the model parameters and \( h \) be the embedding generated by the model for input \( x \).
- Let \( c_1, \ldots, c_n \in \mathbb{R}^d \) be the class embeddings.

**Softmax cross-entropy loss**

- The model assigns probability to the \( i \)-th class according to the softmax distribution:
  \[
  p_i = \frac{e^{h \cdot c_i}}{Z}
  \]
  where \( Z = \sum_{i=1}^{n} e^{h \cdot c_i} \).
- The full softmax cross-entropy loss is defined as:
  \[
  L(x, t) = -\log p_t = -\log Z + \sum_{i=1}^{n} (e^{h \cdot c_i}) 
  \]
-\( t \in [n] \): true class for the input \( x \).

**Sampled softmax** [Bengio and Sennetaal'08]

- Sample \( m \) negative classes \( s_1, \ldots, s_m \sim q \).
- Define the sampled softmax distribution:
  \[
  p_i = \frac{e^{h \cdot c_i}}{Z'}
  \]
  where \( Z' = \sum_{i=1}^{n} e^{h \cdot c_i} \).
- Sampled softmax loss:
  \[
  L'(x, t) = -\log p_t = -\log Z' + \sum_{i=1}^{n} (e^{h \cdot c_i})
  \]

**Gradient bias of sampled softmax**

\[
LB \leq \mathbb{E}[\nabla_x L'] - \nabla_x L \leq UB
\]

where

\[
LB = -\mathbb{E}
\]

**Kernel-based sampling (I)**

- Given a kernel \( K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \) that can be linearized by a mapping \( \phi : \mathbb{R}^d \rightarrow \mathbb{R}^D \), define
  \[
  g_i = \phi(K(h, c_i)) = \phi(h)^T \phi(c_i)
  \]
- [Blanc and Steffen'18] use a quadratic kernel
  \[
  K_{quad}(x, y) = \phi(h)^T \phi(c) + 1
  \]
  where \( \phi(z) = [\sqrt{\gamma}(z \circ z), 1] \in \mathbb{R}^d \).
- Poor approximation of the exponential kernel and prohibitively large \( O(d^2n) \) computational cost.

**Kernel-based sampling (II)**

- Define the sampled softmax distribution
  \[
  p_i = \frac{e^{h \cdot c_i}}{Z'}
  \]
  where \( Z' = \sum_{i=1}^{n} e^{h \cdot c_i} \).
- Sampling cost \( O(D \log n) \) per class.

**Random Fourier softmax (RF-softmax)**

- Given an input embedding \( h \), RF-softmax picks the \( i \)-th class with probability
  \[
  q_i \propto \phi(h)^T \phi(c_i)
  \]
- Quality of approximation: For normalized inputs and class embeddings, \( e^{h \cdot c_i} \leq \frac{e^{h \cdot c_i}}{e^{h \cdot c_j}} \leq 1 - O\left(\frac{1}{\gamma} \right) \).
- With large enough \( D \),
  \[
  q_i \propto (1 + \alpha \phi(1)) \cdot p_i
  \]

**Experiments**

- **Wall time:** Batch size \( =10, m = 10, d = 64 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>Wall time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp</td>
<td>0.42 ms</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.36 ms</td>
</tr>
<tr>
<td>Quadratic</td>
<td>0.36 ms</td>
</tr>
<tr>
<td>RF-softmax</td>
<td>0.37 ms</td>
</tr>
</tbody>
</table>
- **Performance:** NLP datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Wall time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp</td>
<td>0.42 ms</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.36 ms</td>
</tr>
<tr>
<td>Quadratic</td>
<td>0.36 ms</td>
</tr>
<tr>
<td>RF-softmax</td>
<td>0.37 ms</td>
</tr>
</tbody>
</table>
- **Design choices:** Penn tree bank, \( m = 100 \).

**Table 1. MSE for approximating \( e^{h \cdot c} \) using different methods.**

\[
\begin{array}{cccc}
\text{Method} & \text{Exp} & \text{Uniform} & \text{Quadratic} \\
RF-softmax & 0.56 & 0.35 & 0.26 \\
RF-softmax & 0.41 & 0.37 & 0.36 \\
Quadratic & 0.40 & 0.36 & 0.34 \\
\end{array}
\]

**Figure 1. Wall time vs. training steps (Penn tree bank, \( m = 100 \)).**

**Figure 2. Validation perplexity vs. training steps (Treebank, \( m = 100 \)).**

**Performance:** Extreme classification datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Prec@1</th>
<th>Prec@3</th>
<th>Prec@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmazonCat-13k</td>
<td>Exp</td>
<td>0.85</td>
<td>0.62</td>
<td></td>
</tr>
<tr>
<td>n = 13,330</td>
<td>Uniform</td>
<td>0.83</td>
<td>0.69</td>
<td>0.55</td>
</tr>
<tr>
<td>n = 205,443</td>
<td>Quadratic</td>
<td>0.84</td>
<td>0.74</td>
<td>0.60</td>
</tr>
<tr>
<td>RF-softmax</td>
<td>0.87</td>
<td>0.75</td>
<td>0.61</td>
<td></td>
</tr>
<tr>
<td>Delicious-200k</td>
<td>Exp</td>
<td>0.75</td>
<td>0.47</td>
<td>0.36</td>
</tr>
<tr>
<td>n = 205,443</td>
<td>Uniform</td>
<td>0.42</td>
<td>0.38</td>
<td>0.21</td>
</tr>
<tr>
<td>n = 782,585</td>
<td>Quadratic</td>
<td>0.36</td>
<td>0.31</td>
<td>0.32</td>
</tr>
<tr>
<td>RF-softmax</td>
<td>0.37</td>
<td>0.36</td>
<td>0.34</td>
<td></td>
</tr>
<tr>
<td>WikiLBHTC</td>
<td>Exp</td>
<td>0.58</td>
<td>0.37</td>
<td>0.29</td>
</tr>
<tr>
<td>n = 1,147,899</td>
<td>Uniform</td>
<td>0.47</td>
<td>0.29</td>
<td>0.12</td>
</tr>
<tr>
<td>n = 573,404</td>
<td>Quadratic</td>
<td>0.57</td>
<td>0.37</td>
<td>0.28</td>
</tr>
<tr>
<td>RF-softmax</td>
<td>0.56</td>
<td>0.35</td>
<td>0.26</td>
<td></td>
</tr>
</tbody>
</table>