学界 | 规范和优化 LSTM 语言模型

2017-11-17 机器海岸线

选自 arXiv

作者：Stephen Merity，Nitish Shirish Keskar，Richard Socher 等

机器海岸线编译

参与：方建勇

Regularizing and Optimizing LSTM Language Models

Stephen Merity [1] Nitish Shirish Keskar [1] Richard Socher [1]

论文链接：https://arxiv.org/pdf/1708.02182

摘要：递归神经网络（RNN），如长短期记忆网络（LSTM），可作为许多序列学习任务（包括机器翻译，语言建模和问答）的基本构建模块。在本文中，我们考虑了字级语言建模的具体问题，并研究了基于 LSTM 的模型的正则化和优化策略。我们提出了使用 DropConnect 作为一种反复调节形式的权重下降 LSTM 的隐藏到隐藏的权重。进一步，我们引入平均随机梯度法的变量 NT-ASGD，其中平均触发器是使用非单调条件确定的，而不是由用户进行调整。使用这些和其他 reg-ularization 策略，我们在两个数据集上达到最新的字级复杂度：Penn Treebank 上的 57.3 和 WikiText-2 上的 65.8。在研究神经网络缓存与我们提出的模型相结合的有效性方面，我们在 Penn Treebank 上得到了更低的 52.8 的最新复杂度，WikiText-2 上得到了更低的 52.0。

Model	Parameters	Validation	Test
Mikolov & Zweig (2012) - KN-5	2M‡	–	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M‡	–	125.7
Mikolov & Zweig (2012) - RNN	6M‡	–	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M‡	–	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M‡	–	92.0
Zaremba et al. (2014) - LSTM (medium)	20M	86.2	82.7
Zaremba et al. (2014) - LSTM (large)	66M	82.2	78.4
Gal & Ghahramani (2016) - Variational LSTM (medium)	20M	81.9 ± 0.2	79.7 ± 0.1
Gal & Ghahramani (2016) - Variational LSTM (medium, MC)	20M	–	78.6 ± 0.1
Gal & Ghahramani (2016) - Variational LSTM (large)	66M	77.9 ± 0.3	75.2 ± 0.2
Gal & Ghahramani (2016) - Variational LSTM (large, MC)	66M	–	73.4 ± 0.0
Kim et al. (2016) - CharCNN	19M	–	78.9
Merity et al. (2016) - Pointer Sentinel-LSTM	21M	72.4	70.9
Grave et al. (2016) - LSTM	–	–	82.3
Grave et al. (2016) - LSTM + continuous cache pointer	–	–	72.1
Inan et al. (2016) - Variational LSTM (tied) + augmented loss	24M	75.7	73.2
Inan et al. (2016) - Variational LSTM (tied) + augmented loss	51M	71.1	68.5
Zilly et al. (2016) - Variational RHN (tied)	23M	67.9	65.4
Zoph & Le (2016) - NAS Cell (tied)	25M	–	64.0
Zoph & Le (2016) - NAS Cell (tied)	54M	–	62.4
Melis et al. (2017) - 4-layer skip connection LSTM (tied)	24M	60.9	58.3
AWD-LSTM - 3-layer LSTM (tied)	24M	60.0	57.3
AWD-LSTM - 3-layer LSTM (tied) + continuous cache pointer	24M	53.9	52.8

表 1：Penn Treebank 语言建模任务的验证和测试集中的单一模型困惑。 带有‡的参数编号是根据我们对模型的理解并参照 Merity 等人的估计。（2016）。 模型注意绑定使用重量绑定嵌入和 softmax 权重。 我们的型号 AWD-LSTM 代表 ASGD 称量的 LSTM。

Model	Parameters	Validation	Test
Inan et al. (2016) - Variational LSTM (tied) ($h = 650$)	28M	92.3	87.7
Inan et al. (2016) - Variational LSTM (tied) ($h = 650$) + augmented loss	28M	91.5	87.0
Grave et al. (2016) - LSTM	–	–	99.3
Grave et al. (2016) - LSTM + continuous cache pointer	–	–	68.9
Melis et al. (2017) - 1-layer LSTM (tied)	24M	69.3	65.9
Melis et al. (2017) - 2-layer skip connection LSTM (tied)	24M	69.1	65.9
AWD-LSTM - 3-layer LSTM (tied)	33M	68.6	65.8
AWD-LSTM - 3-layer LSTM (tied) + continuous cache pointer	33M	53.8	52.0

表 2：WikiText-2 上的单一模型困惑。 模型注意绑定使用重量绑定嵌入和 softmax 权重。 我们的型号 AWD-LSTM 代表 ASGD 称量的 LSTM。

Word	Count	Δloss	Word	Count	Δloss
.	7632	-696.45	<unk>	11540	5047.34
,	9857	-687.49	Meridian	161	1057.78
of	5816	-365.21	Churchill	137	849.43
=	2884	-342.01	-	67	682.15
to	4048	-283.10	Blythe	97	554.95
in	4178	-222.94	Sonic	75	543.85
<eos>	3690	-216.42	Richmond	101	429.18
and	5251	-215.38	Starr	74	416.52
the	12481	-209.97	Australian	234	366.36
a	3381	-149.78	Pagan	54	365.19
"	2540	-127.99	Asahi	39	316.24
that	1365	-118.09	Japanese	181	295.97
by	1252	-113.05	Hu	43	285.58
was	2279	-107.95	Hedgehog	29	266.48
)	1101	-94.74	Burma	35	263.65
with	1176	-93.01	29	92	260.88
for	1215	-87.68	Mississippi	72	241.59
on	1485	-81.55	German	108	241.23
as	1338	-77.05	mill	67	237.76
at	879	-59.86	Cooke	33	231.11

表 3：在引入连续缓存指针时，给定单词在 WikiText-2 的验证数据集中的所有实例中产生的总损失差异（日志复杂度）。 右列包含具有二十个最佳证明（即，高速缓存是有利的）的字，并且左列是二十恶化（即，高速缓存不利的地方）。

Model	PTB Validation	PTB Test	WT2 Validation	WT2 Test
AWD-LSTM (tied)	60.0	57.3	68.6	65.8
– fine-tuning	60.7	58.8	69.1	66.0
– NT-ASGD	66.3	63.7	73.3	69.7
– variable sequence lengths	61.3	58.9	69.3	66.2
– embedding dropout	65.1	62.7	71.1	68.1
– weight decay	63.7	61.0	71.9	68.7
– AR/TAR	62.7	60.3	73.2	70.1
– full sized embedding	68.0	65.6	73.7	70.7
– weight-dropping	71.1	68.9	78.4	74.9

表 4：我们最好的 LSTM 模型的消融模型报告结果在 Penn Treebank 和 WikiText-2 的验证和测试集上。 消融被分成优化和正则化变体，根据 WikiText-2 上实现的验证复杂度进行排序。

✂----------