

文章编号: 1003-0077(2018)08-0053-07

训练语料的不同利用方式对神经机器翻译模型的影响

邝少辉, 熊德意

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

摘要: 神经机器翻译(NMT)是近两年刚出现的一种新型机器翻译方法, 是一种端到端的翻译模型。目前, 影响NMT模型效果的因素有很多, 其一, 当训练语料规模较大时, 梯度下降更新方法会对机器的内存要求很高, 因此大多研究工作中采用随机梯度下降(SGD)的方法来更新模型的训练参数, 即每输入一定数量(批:batch)的训练样例, 就利用局部的训练样例更新一次模型参数; 其二, 参数 dropout 可以防止系统训练时出现过拟合, 提高系统泛化能力; 其三, 数据打乱(shuffle)也对翻译结果有着重要影响。因此, 该文的研究内容主要是探索批、dropout 和打乱这三个因素在训练神经机器翻译模型中对模型翻译质量的影响, 并得出以下三条结论: 一是批的大小将影响神经机器翻译(NMT)模型的收敛速度, 二是 dropout 可以提升神经机器翻译模型的性能, 三是数据打乱可以在一定程度上提升神经机器翻译(NMT)系统的翻译质量。

关键词: 神经机器翻译; 批; dropout; 数据打乱

中图分类号: TP391

文献标识码: A

The Influence of Different Use of Training Corpus on Neural Machine Translation Model

KUANG Shaohui, XIONG Deyi

(School of Computer Sciences and Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract: Neural machine translation (NMT) is an emerging end-to-end machine translation paradigm. In NMT, the stochastic gradient descent (SGD) is used to update the model parameters. This paper explores the influence on NMT system resulted from the batch, the dropout and the shuffle in SGD. The results show that the size of batch affects the convergence speed of NMT model, hyper parameter dropout has a huge impact on the performance of the NMT model, and data shuffle can improve the translation quality of NMT system.

Key words: neural machine translation; batch; dropout; data shuffle

0 引言

随着互联网和社交网络的发展, 机器翻译在社会发展和信息传播中的作用越来越突出。为了满足人们对机器翻译的强烈需求, 国内外许多研究机构和公司, 对机器翻译进行了深入的研究, 如百度翻译、有道翻译等。机器翻译方法包括基于规则的机器翻译、基于实例的机器翻译、统计机器翻译, 以及当前的神经网络机器翻译等。

随着计算能力的提高, 可用训练数据量的增加, 基于深度学习的神经网络已在多个领域取得较好的结果, 如图像识别、语音识别等。神经机器翻译

(neural machine translation)^[1-3]作为目前一种主流的机器翻译建模方法, 也是利用神经网络来构建翻译模型, 在多个语言对上^[4]的翻译效果已经赶超传统统计机器翻译(statistical machine translation, SMT)模型。神经机器翻译系统采用“端到端”(end-to-end)的思想, 分别使用两个不同的神经网络作为编码器(encoder)和解码器(decoder)来搭建翻译模型。和统计机器翻译相比, 神经机器翻译实现了源语言到目标语言的直接翻译, 并在性能上取得进一步的提升。

在本文中, 我们会详细介绍神经机器翻译系统的原理构成, 并实现一套基本的神经机器翻译系统作为实验的基准系统。同时, 由于神经机器翻译训

收稿日期: 2017-09-18 定稿日期: 2017-10-28

基金项目: 国家自然科学基金优秀青年基金(61622209)

练过程中,会设置很多的超参,这些参数会影响系统的整体性能和训练时间。我们在文中对这些参数的设置进行实验对比,分析这些参数的影响。对比试验主要集中在批、打乱、dropout 这三个参数设置上。

目前很多的相关工作在训练神经机器翻译系统时,均会提到对数据进行随机打乱。因此,我们针对打乱这一因素进行实验证,来深入分析打乱是如何影响神经机器翻译系统的翻译质量的,并给出具体的统计实验结果。

批的大小是神经机器翻译系统训练过程中必不可少的一个参数,在训练时对神经机器翻译系统的收敛速度影响较大。本文通过设计一系列对比实验来验证批的大小对神经机器翻译系统训练时间的影响。

参数 dropout^[5]可以有效防止系统训练过程中的过拟合现象,并在一定程度上影响模型的性能。在本文中,我们进一步地利用对比实验来验证不同 dropout 设置时,系统性能的变化。

通过一系列的对比实验证,我们得出以下结果:通过打乱训练数据。可以一定程度上提高神经机器翻译的性能;改变批的大小对神经机器翻译系统的训练时间有较大影响,批设置越大,神经机器翻译系统训练时间越短,反之,训练时间越长。参数 dropout 可以显著提升神经机器翻译系统的性能,不同的值对神经机器翻译系统有着不同程度的影响。

1 神经机器翻译

1.1 基于注意力机制的神经机器翻译

在神经机器翻译中,一般采用编码器-解码器(encoder-decoder)框架^[1,6]来实现翻译的过程,具体流程如图 1 所示。

对训练语料中的每一个词,我们都为其初始化一个词向量^[7-8],语料中所有词的词向量构成了词向量词典。词向量,一般是一个多维的向量,向量中每一维都是一个实数。例如,对于单词“咱们”,它的词向量可能是 $\{0.12, -0.23, \dots, 0.99\}$ 。

编码器由循环神经网络(recurrent neural network)^[3,9-10]构成。在编码阶段,编码器读入一个句子,并将句子编码成一系列的向量。具体过程如下,首先将一个句子表示为词向量的序列,即 $x = \{x_1, x_2, \dots, x_T\}$

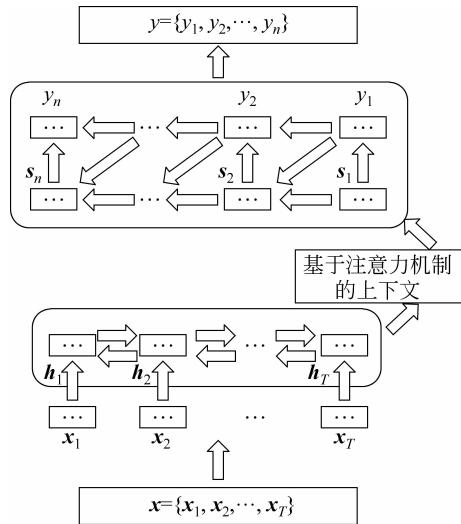


图 1 神经机器翻译框架图

$x_2, \dots, x_{T_x}\}$, 其中 x 为输入的句子, x_i 为句子中第 i 个词的词向量, 即一个 m 维的向量。根据式(1)我们可以获得一个由隐藏向量组成的向量序列 $\{h_1, h_2, \dots, h_{T_x}\}$ 。由这个隐藏向量序列, 我们可以获得上下文向量 $c = q(\{h_1, h_2, \dots, h_{T_x}\})$ 。其中 $h_j \in \mathbb{R}^n$, 是时序 t 时刻的编码器隐藏状态, f 和 q 是非线性的激活函数, 其中 f 一般采用 GRU^[11] 或者 LSTM^[12], q 一般采用注意力^[13-14]网络。

$$h_j = f(x_j, h_{j-1}) \quad (1)$$

神经机器翻译系统中, 编码器一般采用双向的循环神经网络(BI-RNN)网络来实现, 分别为正向循环神经网络(forward recurrent neural network)和反向循环神经网络(backward recurrent neural network)。单向循环神经网络仅能够捕捉一个顺序方向的序列信息, 而双向循环神经网络可以从两个不同的方向来捕捉序列信息, 使生成的语义向量含有的语义信息更为丰富。

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j \quad (2)$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T_x} e_{tk}} \quad (3)$$

$$e_{tj} = a(s_{t-1}, h_j) \quad (4)$$

基于注意力机制(attention-based)的神经机器翻译系统中, 上下文向量 c 一般利用注意力网络来获得, 注意力网络可以通过式(2)~式(4)表示, 其中 a 是一个一层的前向网络, α_{tj} 是编码器的每一个隐藏状态 h_j 的权重。注意力机制如图 2 所示。

神经机器翻译系统中, 解码器通常也由循环神

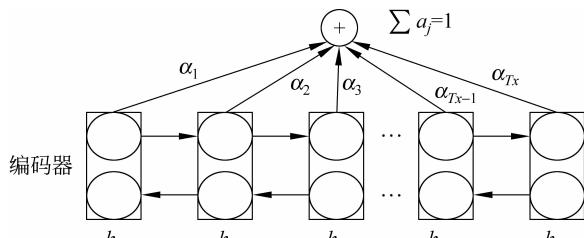


图 2 注意力网络结构

经网络构成。在解码器阶段,给定上下文向量 c_t ,以及所有已经预测生成的词 $\{y_1, y_2, \dots, y_{t-1}\}$,解码器可以根据式(5)预测生成下一个单词 y_t 的概率。

$$p(y_t | \{y_1, y_2, \dots, y_{t-1}\}, c_t) = g(y_{t-1}, s_t, c_t) \quad (5)$$

其中, g 是非线性激活函数,一般采用 softmax 函数。 s_t 为循环神经网络中的隐藏状态,可以通过式(6)获得。

$$s_t = f(y_{t-1}, s_{t-1}, c_t) \quad (6)$$

编码器和解码器都采用循环神经网络网络,主要是因为循环神经网络网络的特点在于隐藏状态由当前的输入和上一个隐藏状态共同决定。如在神经机器翻译过程中,编码器阶段隐藏状态由源端语句当前词的词向量和上一个隐藏状态共同决定。解码器阶段的隐藏状态由前一步骤中计算得到的目标端语句的词向量和上一个隐藏状态共同决定。图 1 中 h 为隐藏状态, x 为源语端词向量序列, y 为目标语端词向量序列。

模型的训练一般采用最大化对数似然作为损失函数,利用随机梯度下降方法来进行迭代训练。其目标函数,如式(7)所示。

$$L(\theta) = \max_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(y_n | x_n) \quad (7)$$

其中, θ 是模型的参数, (y_n, x_n) 表示双语训练语料句对。

1.2 增强的注意力机制

受到开源机器翻译系统 DL4MT^[15] 的启发,我们在工作中实现了一个带有反馈机制(feedback attention)^[16]的神经机器翻译系统。在反馈机制中, e_{ij} 可以通过式(8)计算得到。

$$e_{ij} = a(\tilde{s}_{t-1}, h_j) \quad (8)$$

其中,状态 \tilde{s}_{t-1} 通过式(9)更新,解码器的隐藏状态 s_t 按照式(10)更新。

$$\tilde{s}_{t-1} = GRU(s_{t-1}, y_{t-1}) \quad (9)$$

$$s_t = GRU(\tilde{s}_{t-1}, c_t) \quad (10)$$

2 实验

2.1 实验设置

我们针对中英翻译任务开展实验,使用中英双语平行语料作为模型的训练语料,双语平行语料包含有 125 万句对,其中中文单词 8 090 万,英文单词 8 640 万。语料主要来自于宾夕法尼亚大学的语言数据联盟发布的 LDC 双语语料的部分子集: LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, LDC2004T08, LDC2005T06。实验中采用 NIST06 作为开发集, NIST02, NIST03, NIST04, NIST05, NIST08 作为测试集,并且选择 BLEU-4^[17] 作为翻译模型的质量评估标准。

系统实现中,我们采用 TensorFlow^[18],一个开源的深度学习框架,来实现 1.2 中介绍的带有反馈注意力机制的神经机器翻译模型,称之为 TF-NMT。在 TF-NMT 模型中,编码器采用双向循环神经网络,隐层单元(hidden unit)个数设置为 1 000。同样,解码器的隐层单元(hidden unit)个数也设置为 1 000。词向量(Wordembedding)的维度设置为 620。

训练 TF-NMT 模型的语料的中文和英文句子长度均限制在 50 个单词以内,长度大于 50 个单词的句子将被过滤掉。中文端和英文端词典大小均设定为 16 000, 经统计, 源端词典在中文端训练语料中的覆盖率达到 95.8%, 目标端词典在英文端训练语料中的覆盖率达到 98.2%, 用单词“UNK”取代其他不在词表中的低频词。实验中的其他参数设定,均与开源神经机器翻译系统 GroundHog^[1] 保持一致。我们使用随机梯度下降算法和 Adadelta^[19] 算法来训练我们的模型。Adadelta 算法的参数 ρ 和 ϵ 分别设定为 0.95 和 10^{-6} 。

我们的实验主要集中在以下三方面:

(1) 批大小对神经机器翻译系统训练过程的影响。

(2) 数据打乱对神经机器翻译系统质量的影响。

(3) Dropout 技术对神经机器翻译系统质量的影响。

为了验证批大小对神经机器翻译系统训练过程的影响,我们将批设定为不同的大小,分别为 40, 80, 120, 180。批的值设定过大时,会造成 GPU 显

存不足,从而无法进行训练。为了使 TF-NMT 在不同的批设定下都能够正常工作,我们利用 TensorFlow 框架,实现了基于数据并行的多 GPU (multi-GPU) 的 TF-NMT 系统,我们统称为 TF-NMT 系统。

为了验证数据打乱对神经机器翻译系统翻译质量的影响,我们设定了两种不同的数据迭代方式:

(1) 首先对训练语料进行一次打乱。训练语料读取后,不再进行打乱,按照固有顺序进行循环迭代。

(2) 整个训练语料每迭代完成一次,进行一次打乱,然后再次迭代。

另外,我们训练 GroundHog 系统作为对比系统,用以验证 TF-NMT 系统的翻译效果。GroundHog 系统中编码器和解码器的隐层单元数量配置和 TF-NMT 系统相同,词向量的维度也和 TF-NMT 系统保持一致,其他参数采用其默认配置(未

采用 dropout 技术)。

在 TF-NMT 中的基准系统训练,批和数据打乱的对比实验中,我们将 dropout 概率统一设置为 0.5。

为了验证 dropout 对神经机器翻译系统的影响,我们将 dropout 大小分别设置为 1,0.2,0.5,0.8 进行了四组实验。其中 dropout 设置为 1 时,即代表不使用 dropout。

2.2 TF-NMT 模型验证

我们依据现有已发表论文^[1]中常用的设定,设置批为 80,来验证我们实现的 TF-NMT 系统。表 1 给出了统计结果。当词典大小设定为 16 000,从表 1 中可以看出,当数据不进行打乱,我们实现的带有反馈注意力的 TF-NMT 系统,在 BLEU 值上面超过 GroundHog 系统平均 3.5 个点。当数据进行打乱时,BLEU 值超过 GroundHog 系统平均 3.78 个点。

表 1 GroundHog 和 TF-NMT 实验结果

Model	Batch	Shuffle	Voc	NIST06	NIST02	NIST03	NIST04	NIST05	NIST08	Mean
GroundHog	80	No	16 000	30.41	34.60	32.01	34.52	30.66	23.17	30.89
TF-NMT	80	No	16 000	34.41	37.68	36.14	38.36	34.39	25.39	34.39
TF-NMT	80	Yes	16 000	34.97	37.67	35.91	38.67	34.54	26.30	34.67
TF-NMT	80	Yes	30 000	36.68	39.78	37.45	40.31	36.59	28.22	36.53

注: 我们采用 BLEU 作为评测标准。Voc 表示源端词典和目标端词典大小,Mean 代表在 6 个测试集上面的平均 BLEU 值。

为了进一步验证系统效果,我们在词典大小设置为 30 000 时也进行了实验。从表 1 可以看出,词典设置为 30 000 时,TF-NMT 系统在 NIST 各个测试集上,平均 BLEU 值为 36.53。实验结果证明,我们实现的基准系统 TF-NMT 可以达到并超过目前已公布和开源的神经机器翻译系统的效果。

2.3 Batch 对神经机器翻译系统的影响

为了验证批这一变量对神经机器翻译系统的影响,我们将批设置为不同的大小,来训练多个 TF-NMT 系统,并统计各个 TF-NMT 系统的 BLEU 值变化。为了验证的准确性,我们分别在训练数据进行打乱和不进行打乱的情况下,各进行了一系列实验。表 2 和表 3 分别给出了详细的实验结果。

表 2 当数据进行打乱时,批设置为 40,80,120,180 时,TF-NMT 在各个测试集上的 BLEU 值

Batch	Shuffle	NIST06	NIST02	NIST03	NIST04	NIST05	NIST08	Mean	Iters
40	Yes	34.95	38.15	35.90	39.13	35.44	26.01	34.93	250 000
80	Yes	34.97	37.67	35.91	38.67	34.54	26.30	34.67	181 000
120	Yes	34.59	37.79	36.06	39.05	35.39	26.46	34.89	126 000
180	Yes	34.58	38.54	36.73	39.14	35.72	25.94	35.10	93 500

注: Mean 代表在 6 个测试集上的平均 BLEU 值。Iters 表示 TF-NMT 在开发集 NIST06 上达到最大 BLEU 值时候,训练的批数量。

表3 当数据不进行打乱时,批设置为40,80,120,180时,TF-NMT在各个测试集上的BLEU值

Batch	Shuffle	NIST06	NIST02	NIST03	NIST04	NIST05	NIST08	Mean	Iters
40	No	34.49	37.59	35.86	38.21	34.60	26.03	34.46	277 000
80	No	34.41	37.68	36.14	38.36	34.39	25.39	34.39	190 000
120	No	34.24	37.82	35.16	38.41	34.17	24.88	34.11	110 000
180	No	33.84	38.26	34.77	37.64	34.22	25.15	33.98	127 000

注: Mean 代表在 6 个测试集上的平均 BLEU 值。Iters 表示 TF-NMT 在开发集 NIST06 上达到最大 BLEU 值时候,训练的批数量。

由表 2 和表 3 的实验结果可知,在保持打乱条件一致的前提下,批变化的大小对 BLEU 值的影响并不明显。从表 2 中可以发现,在训练数据进行打乱的条件下,平均 BLEU 值波动范围在 0.45 个点之内。从表 3 也可以发现,平均 BLEU 值的波动范围在 0.48 个点之内。

另外,从表 2 和表 3 中我们可以发现,在不同的批条件下,模型收敛速度不同。比如,在表 2 中,随着批的大小从 40 增加到 180,模型在开发集上达到最好效果需要迭代的批数量(iters)依次降低,模型训练时间依次减少。

为了进一步直观地比较不同批条件下,模型的

收敛速度。在打乱一致的前提下,我们每训练完成 500 个批,就进行一次 BLEU 验证。在数据进行打乱这一条件下,图 3 给出了模型在开发集 NIST06 上随着训练的进行,BLEU 值变化的情况。从图 3 中可以看出,批-180 的曲线最先收敛,BLEU 值增长迅速。批-120 和批-180 相比,批-120 收敛较慢,但比批-80 和批-40 较快。批-40 收敛最慢,在训练了 150 000 个批之后,仍未达到 BLEU 最好的收敛点。结合表 2 可以看出,批-80 和批-40 要达到最好 BLEU 值收敛点,要分别训练 181 000 和 250 000 个批。而批-180 只需要训练 90 000 个批左右,就可以达到最好收敛点,图 3 中批-180 曲线很好的体现了这一点。

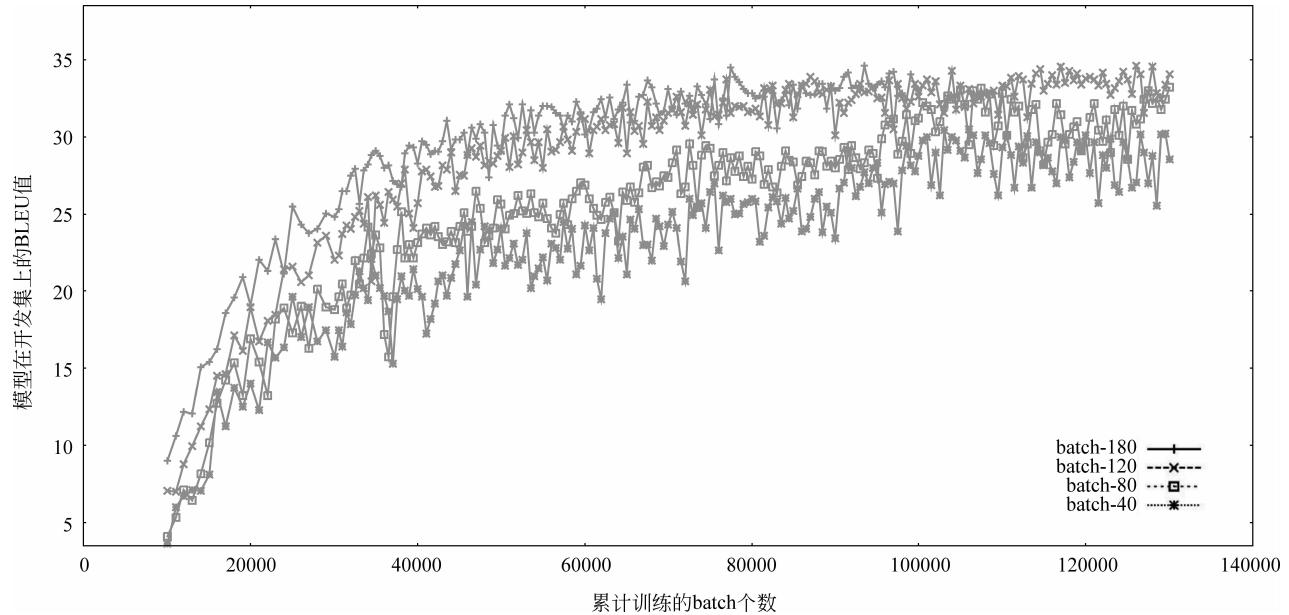


图3 不同批大小设定下 BLEU 值随着训练批个数增加的变化

2.4 打乱对神经机器翻译系统的影响

为了验证数据打乱对神经机器翻译系统的影响,我们对不同打乱条件下,模型在六个 NIST 测试集上的平均结果进行统计,结果如表 4 所示。

表4 不同批大小设置下 TF-NMT 在六个 NIST 测试集上面的平均值。

Shuffle	Batch-40	Batch-80	Batch-120	Batch-180	Mean
Yes	34.93	34.67	34.89	35.10	34.90
No	34.46	34.39	34.11	33.98	34.23

注: Mean 表示在打乱条件相同的前提下,模型的平均值。

从表 4 中我们可以看出,无论批大小设置为多少,训练数据进行打乱之后训练的神经机器翻译系统 BLEU 值上,总是比不进行打乱的情况要好。打乱之后,平均 BLEU 值提高 0.67 个点。在训练神经机器翻译系统过程中,随机梯度下降方法的一个特点就是每次只需要一个批的数据就可以进行梯度更新,十分简单有效。但是这种方法也有一定的缺陷,举例来说:假设训练语料由三种不同领域(例如新闻、教育、军事)的数据按照顺序组成,神经机器翻译模型开始进行训练之后,会先利用新闻语料进行模型参数更新,接着利用教育领域语料更新参数,当神经机器翻译模型训练将要结束时,使用军事领域的语料来更新参数。而神经机器翻译模型在一定程度上就是利用最近输入的语料来进行模型优化,这样会促使神经机器翻译模型朝着更为有利于军事领域的参数空间进行优化,导致神经机器翻译模型在语料中其他领域的适应性降低。这种情况,可以看作是模型参数的一种偏爱(bias)^[20]。为了解决利用随机梯度下降方法训练神经机器翻译模型潜在的这一问题,训练语料迭代一次,就对训练语料进行打乱,再进行模型训练,这是一种值得推荐而且有效的方式。

从表 4 结果可以发现,当不进行数据打乱时,批越大,数据之间的规律更容易影响神经机器翻译系统的性能。而数据进行打乱时,系统性能趋于稳定。这也从另一个方面说明了数据打乱对 NMT 系统的影响。

2.5 TF-NMT 系统训练时间

我们进一步统计了不同批条件和不同 GPU(型

号: GPU GeForce GTX 1080)个数配置下,神经机器翻译系统的训练时间。表 5 给出了详细的数据。从表 5 可以看出,在利用单 GPU 训练时,批 -40 和批 -80 训练时间在 40 至 45 个小时左右。利用数据并行, GPU 个数配置为 3 时,批 -120 和批 -180 只需要 20 个小时左右就可以达到最佳性能,训练时间缩短两倍。

表 5 不同批大小设置下 TF-NMT 达到最佳性能所用的训练时间

Params	Batch-40	Batch-80	Batch-120	Batch-180
GPU	1	1	3	3
Iters	250 000	181 000	126 000	93 500
Batch_Time/s	0.62	0.79	0.64	0.71
Parallel_Time/h	43.06	39.72	22.4	18.44

注: GPU 代表训练时使用的 GPU 数量;Iters 代表达到最佳性能时训练的批数量;Batch_Time 代表每训练一个批所用时间,单位为秒;Parallel_Time 代表在当前 GPU 个数设置下系统训练的时间,单位为小时。

同时,系统达到最佳性能的训练时间,随着批的增大而逐渐缩小,也再次证明了批对系统收敛速度的影响。

2.6 Dropout 对神经机器翻译系统质量的影响

为了验证 dropout 对神经机器翻译系统性能的影响,我们分别对不同 dropout 设置下,模型在六个 NIST 测试集上的结果进行统计,模型结果如表 6 所示。

表 6 不同 dropout 设置下,TF-NMT 模型的效果。

Model	Batch	Shuffle	Dropout	NIST06	NIST02	NIST03	NIST04	NIST05	NIST08	Mean
TF-NMT	80	Yes	1	33.28	36.28	34.52	36.99	33.31	23.84	33.02
TF-NMT	80	Yes	0.8	33.97	37.15	34.89	38.09	33.92	24.88	33.81
TF-NMT	80	Yes	0.5	34.97	37.67	35.91	38.67	34.54	26.30	34.67
TF-NMT	80	Yes	0.2	35.53	37.96	35.98	39.13	35.55	26.58	35.12

注: 我们采用 BLEU 作为评测标准。Mean 代表在六个测试集上面的平均 BLEU 值。

从表 6 中可以看出,在 dropout 设置为 1 时(即 dropout 不被使用),模型在六个 NIST 测试集上的平均 BLEU 值为 33.02。dropout 设置为 0.5 时(baseline 系统),模型的平均 BLEU 值为 34.67。我们可以得出结论:不使用 dropout,模型的平均 BLEU 值降低了 1.64。

在本文中,dropout 设置为 0.5 时,我们在所有

的批和打乱实验上面,都取得了较好的模型性能,这可以进一步地证明 dropout 对神经机器翻译系统性能的提升。另外,当 dropout 分别设置为 0.8,0.5,0.2 时,BLEU 值依次提升。当 dropout 设置为 0.8 和 0.2 时,BLEU 值相差 1.31 个点。根据这四组实验可以看出:dropout 值设置越小,神经机器翻译系统倾向于表现出更好的性能。

3 总结与展望

本文详细论述了神经机器翻译的基本原理，并基于TensorFlow深度学习框架，实现了带有反馈注意力网络的神经机器翻译系统TF-NMT。为了测试不同批大小对神经机器翻译系统的影响，进一步实现了基于数据并行的multi-GPU神经机器翻译模型。通过对比开源神经机器翻译系统Ground-Hog，发现我们的TF-NMT系统效果能够到达目前神经机器翻译研究领域已公布的基于注意力网络的神经机器翻译系统的效果。

在TF-NMT模型上面，我们验证了batch、dropout和打乱这三个因素对神经机器翻译系统的影响。实验证明，批大小会影响模型训练时的收敛速度。在一定程度上，批的值越大，模型收敛速度越快。从实验结果中我们也发现，对训练数据进行打乱，在一定程度上能够提高神经机器翻译系统的翻译性能。另外，在训练神经机器翻译系统过程中，dropout可以有效地提升神经机器翻译系统的性能。

本次研究主要集中在神经机器翻译模型保持不变的情况下，超参数、打乱和dropout对神经机器翻译系统的影响。未来我们将会进行更多实验，来验证神经机器翻译系统训练中其他参数的影响。

参考文献

- [1] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv. 1409.0473, 2014.
- [2] Cho K, Merriënboer B V, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv. 1409.1259, 2014.
- [3] Cho K, Merriënboer B V, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv. 1406.1078, 2014.
- [4] Firat O, Cho K, Sankaran B, et al. Multi-way, multilingual neural machine translation [J]. Computer Speech and Language, 2017, 45:236-252.
- [5] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
- [6] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proceedings of the Advances in Neural Information Processing Systems. 2014: 3104-3112.
- [7] Siwei Lai, Kang Lin, Shizhu He, et al. How to Generate a Good Word Embedding[J]. IEEE Intelligent Systems, 2016, 31(6):5-14.
- [8] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of the International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:3111-3119.
- [9] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Proceedings of the IEEE International Conference on IEEE, 2013: 6645-6649.
- [10] Koutník J, Greff K, Gomez F, et al. A Clockwork RNN[J]. arXiv preprint arXiv. 1402.3511, 2014.
- [11] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[C]//Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop, 2014.
- [12] Hochreiter S, Jürgen S. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.
- [13] Luong Minh-Thang, Hieu Pham, Christopher D. Manning. Effective approaches to attention-based neural machine translation[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015.
- [14] Cheng Y, Shen S, He Z, et al. Agreement-based joint training for bidirectional attention-based neural machine translation[C]//Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016.
- [15] Hao Zhou, Zhaopeng Tu, Shujian Huang, et al. Chuhk-Based Bi-Scale Decoder for Neural Machine Translation[C]//Proceedings of the 55th annual meeting on association for computational linguistics, 2017.
- [16] Xing Wang, Zhengdong Lu, Zhaopeng Tu, et al. Neural machine translation advised by statistical machine translation[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
- [17] PPapineni K, Roukos S, Ward T, et al. BLEU: A method for automatic evaluation of machine translation[C]//Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002: 311-318.
- [18] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning[C]//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA. 2016.

(下转第 67 页)

translation of rare words with subword units[C]// Proceedings of ACL 2016:1715-1725.

[17] 李良友,贡正仙,周国栋.机器翻译自动评价综述[J].中文信息学报,2014,28(03):81-91.



包乌格德勒(1979—),博士,副教授,主要研究领域为自然语言处理、机器翻译等。
E-mail: wgd2827@163.com



赵小兵(1967—),博士,教授,主要研究领域为自然语言处理、舆情分析等。
E-mail: nmzxb_cn@163.com

~~~~~

(上接第 59 页)

- [19] Zeiler M D. ADADELTA: An Adaptive learning rate method[J]. arXiv preprint arXiv:1212.5701, 2012.  
[20] Neubig G. Neural machine translation and sequence-

to-sequence models: A Tutorial[J], arXiv preprint arXiv: 1703.01619, 2017.



邝少辉(1991—),硕士研究生,主要研究领域为自然语言处理、机器翻译。  
E-mail: shaohuikuang@foxmail.com



熊德意(1979—),通信作者,博士,教授,主要研究领域为自然语言处理、人工智能。  
E-mail: dyxiong@suda.edu.cn