使用《全宋词》训练写诗机器人

1 实验结果

Word embedding的结果如图1所示。由图可见，意思相近的词位置相近，如霜露、别离、烟雨雾云霞、二三四五六八九十千百万等。其中“一”与许多词都可以搭配，如 “一枝”、“一片”、“一心”等，与“半”、“几”和“数”距离较近。

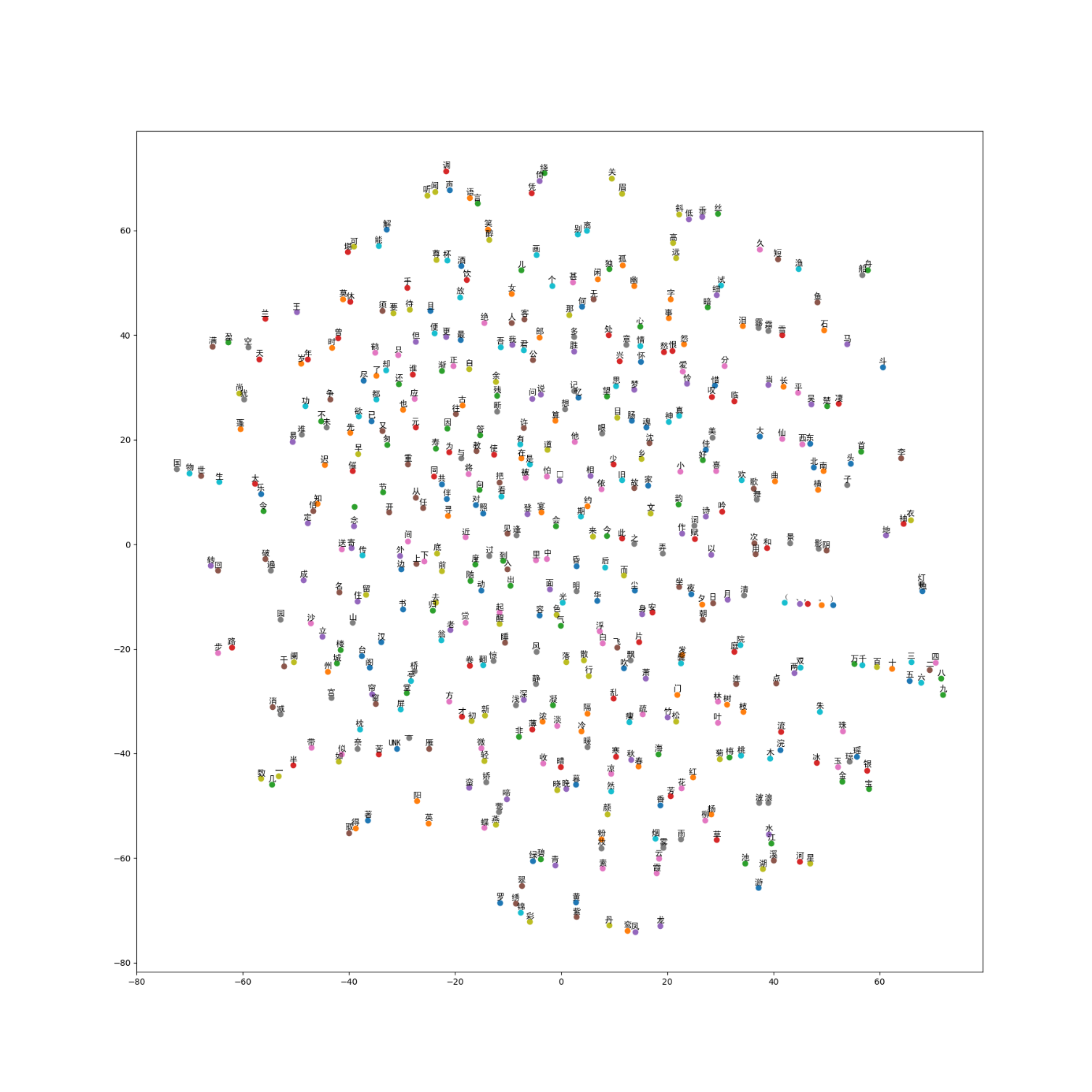


图1 word embedding

LSTM经过20个epoch的训练，损失函数从5.2以上降到3.8左右，如图2所示。

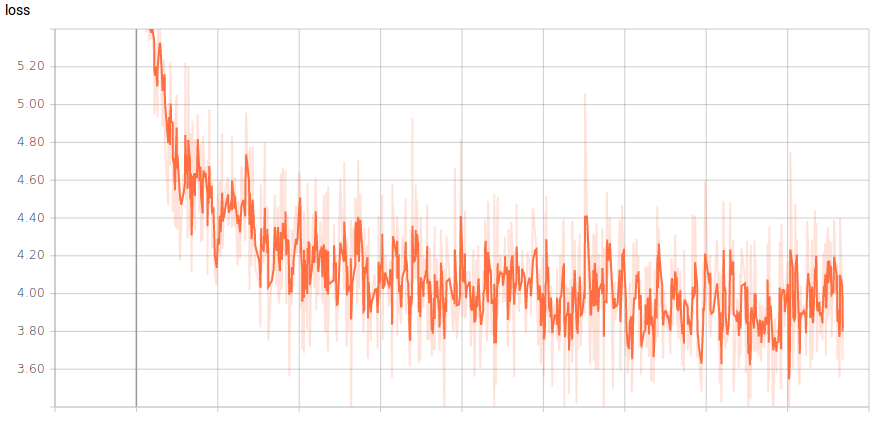


图2 损失函数变化

验证结果如图3所示。可以看到在word embedding的作用下，LSTM成功使用了一些词语，如秋色、秋光、明月、琉璃等；出现了略有意境的句子，如一声声断、一片愁肠、一枝春色满江南等；但是词牌的格式还不太正确，词牌名后未能正确预测回车，有的词语搭配很别扭，如一枝一笑、一枝明月、一点春风等。

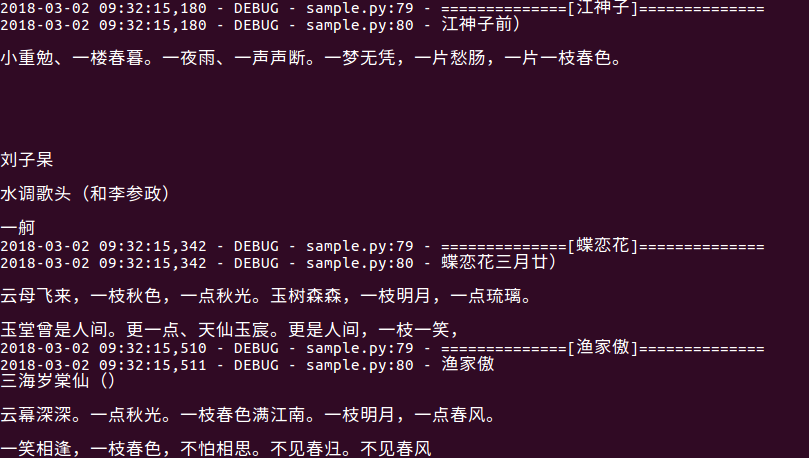


图3 训练结果

2 word embedding

生成词向量的传统方式是将汉字或单词编码成onehot向量的形式，这样做最大的问题是我们的词汇表一般都非常大，比如达到百万级别，这样每个词都用百万维的向量来表示简直是内存的灾难。考虑通过训练将每个词映射为一个较短的词向量，这些词向量构成连续的向量空间，在那里语义相近的词位于相邻的位置。传统的神经网络词向量语言模型有2种：CBOW模型根据输入周围n-1个词来预测出这个词本身；skip-gram模型根据词本身来预测周围有哪些词。也就是说，CBOW模型的输入是某个词A周围的n个单词的词向量之和，输出是词A本身的词向量；而skip-gram模型的输入是词A本身，输出是词A周围的n个单词的词向量。

skip-gram model由Mikolov 于2013年提出，模型根据词本身来预测周围有哪些词，经优化后单机每天可处理1000亿字，并且在词汇的向量表示中显现出了某种线性关系：法国-巴黎=中国-北京。但是由于它使用的softmax函数的计算复杂度与词汇数目成正比，而词汇数目通常在105-107，因此训练起来不切实际。之后Mikolov提出了3个改进：Hierarchical Softmax，Negative Sampling，Subsampling of Frequent Words。

word2vec 是 Google 于 2013 年开源推出的一个用于获取词向量的工具包，它简单、高效，因此引起了人们的广泛关注。

3 LSTM

传统的RNN是从输入的角度上看，非常深的神经网络，即t时刻的输出不仅由t时刻的输入决定，而且通过state由t-1, t-2,…,0时刻的输入共同决定。t时刻的输出产生的梯度，在通过BPTT向前传播以更新与0时刻输入相关联的w时，要经过一个sigmoid激活函数（导数取值0~0.25）和多个tanh激活函数（导数取值0~1），而激活函数的取值很容易饱和，在取值饱和的地方梯度值会非常小，在给定学习率的情况下，梯度可能发生剧烈变化，从而产生梯度爆炸和梯度消失的问题，导致网络难以训练。

LSTM(https://colah.github.io/posts/2015-08-Understanding-LSTMs/)是RNN的一个变种，属于反馈神经网络的范畴。理解LSTM的关键在于把握Ct(cell state)的变化。Ct-1通过遗忘门丢掉一些历史信息，通过输入门加入一些当前信息得到Ct，然后让Ct通过输出门得到ht。通过加入这些门控单元，LSTM的梯度得以以常数级别向后传播，不会因非线性激活函数而导致梯度以指数级衰减，解决了传统的RNN由于梯度爆炸和梯度消失导致训练效率低下甚至无法训练的问题，相较于标准RNN具有较长期的短期记忆。