LSTM算法原理及其在***中应用

莲华

一、背景

LSTM（Long Short-Term Memory）算法作为深度学习方法的一种，在介绍LSTM算法之前，有必要介绍一下深度学习（Deep Learning）的一些基本背景。

目前在机器学习领域，最大的热点毫无疑问是深度学习，从谷歌大脑（Google Brain）的猫脸识别[1]，到ImageNet比赛中深度卷积神经网络的获胜[2]，再到Alphago大胜李世石[3]，深度学习受到媒体、学者以及相关研究人员越来越多的热捧。这背后的原因无非是深度学习方法的效果确实超越了传统机器学习方法许多。从2012年Geoffrey E. Hinton的团队在ImageNet比赛（图像识别中规模最大影响最大的比赛之一）中使用深度学习方法获胜[4]之后，关于深度学习的研究就呈井喷之势；在2012年以前，该比赛结果的准确率一直处于缓慢提升的状态，这一年突然有质的飞越，而从此之后深度学习方法也成为了ImageNet比赛中的不二选择。同时，深度学习的影响却不仅局限于图像识别比赛，也深刻影响了学术界和工业界，顶级的学术会议中关于深度学习的研究越来越多，如CVPR、ICML等等，而工业级也为深度学习立下了汗马功劳，贡献了越来越多的计算支持或者框架，如Nivdia的cuda、cuDnn，Google的tensorflow，Facebook的torch和微软的DMTK等等。

深度学习技术发展的背后是广大研究人员的付出，目前该领域内最著名的研究人员莫过于Yoshua Bengio，Geoffrey E. Hinton，Yann LeCun以及Andrew Ng。最近Yoshua Bengio等出版了《Deep Learning》[5]一书，其中对深度学习的历史发展以及该领域内的主要技术做了很系统的论述，其关于深度学习历史发展趋势的总结非常精辟，书中总结的深度学习历史发展趋势的几个关键点分别：

a）深度学习本身具有丰富悠久的历史，但是从不同的角度出发有很多不同得名，所以历史上其流行有过衰减趋势。

b）随着可以使用的训练数据量逐渐增加，深度学习的应用空间必将越来越大。

c）随着计算机硬件和深度学习软件基础架构的改善，深度学习模型的规模必将越来越大。

d）随着时间的推移，深度学习解决复杂应用的精度必将越来越高。

而深度学习的历史大体可以分为三个阶段。一是在20世纪40年代至60年代，当时深度学习被称为控制论；二是在上世纪80年代至90年代，此期间深度学习被誉为联结学习；三是从2006年开始才以深度学习这个名字开始复苏（起点是2006年，

Geoffrey Hinton发现深度置信网可以通过逐层贪心预训练的策略有效地训练）。总而言之，深度学习作为机器学习的一种方法，在过去几十年中有了长足的发展。随着基础计算架构性能的提升，更大的数据集和更好的优化训练技术，可以预见深度学习在不远的未来一定会取得更多的成果。

二、LSTM算法

LSTM算法全称为Long short-term memory，最早由 Sepp Hochreiter和Jürgen Schmidhuber于1997年提出[6]，是一种特定形式的RNN（Recurrent neural network，循环神经网络），而RNN是一系列能够处理序列数据的神经网络的总称。这里要注意循环神经网络和递归神经网络（Recursive neural network）的区别。

一般地，RNN包含如下三个特性：

a）循环神经网络能够在每个时间节点产生一个输出，且隐单元间的连接是循环的；

b）循环神经网络能够在每个时间节点产生一个输出，且该时间节点上的输出仅与下一时间节点的隐单元有循环连接；

c）循环神经网络包含带有循环连接的隐单元，且能够处理序列数据并输出单一的预测。

RNN还有许多变形，例如双向RNN（Bidirectional RNN）等。然而，RNN在处理长期依赖（时间序列上距离较远的节点）时会遇到巨大的困难，因为计算距离较远的节点之间的联系时会涉及雅可比矩阵的多次相乘，这会带来梯度消失（经常发生）或者梯度膨胀（较少发生）的问题，这样的现象被许多学者观察到并独立研究。为了解决该问题，研究人员提出了许多解决办法，例如ESN（Echo State Network），增加有漏单元（Leaky Units）等等。其中最成功应用最广泛的就是门限RNN（Gated RNN），而LSTM就是门限RNN中最著名的一种。有漏单元通过设计连接间的权重系数，从而允许RNN累积距离较远节点间的长期联系；而门限RNN则泛化了这样的思想，允许在不同时刻改变该系数，且允许网络忘记当前已经累积的信息。

LSTM就是这样的门限RNN，其单一节点的结构如下图1所示。LSTM的巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样一来在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

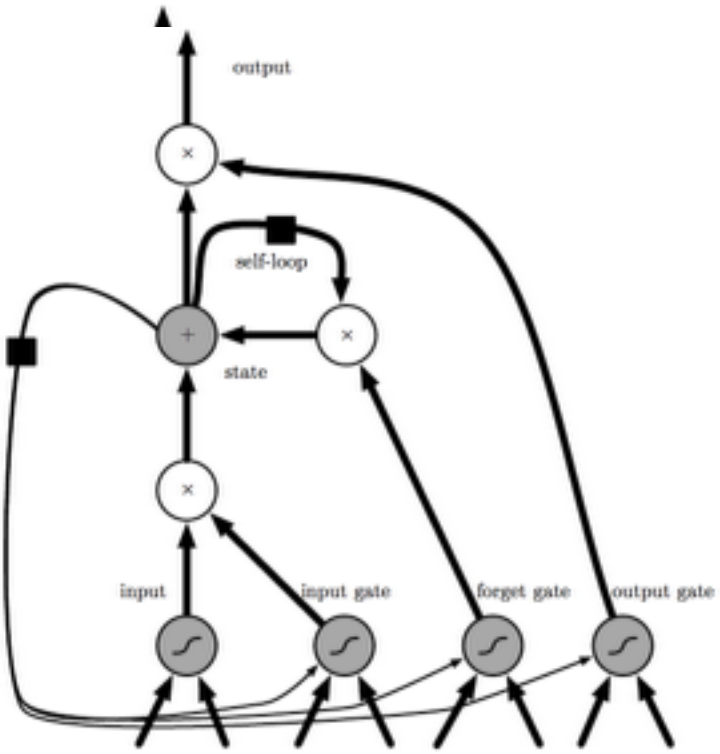

图1 LSTM的CELL示意图[5]

根据LSTM网络的结构，每个LSTM单元的计算公式如下图2所示，其中Ft表示遗忘门限，It表示输入门限，˜Ct表示前一时刻cell状态、Ct表示cell状态（这里就是循环发生的地方），Ot表示输出门限，Ht表示当前单元的输出，Ht-1表示前一时刻单元的输出。

$$
\begin{aligned}
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right) \\
i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t &= \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right) \\
h_t &= o_t * \tanh\left(C_t\right)
\end{aligned}
$$

图2 LSTM计算公式

三、BPTT

介绍完LSTM算法的原理之后，自然要了解如何训练LSTM网络。与前馈神经网络类似，LSTM网络的训练同样采用的是误差的反向传播算法（BP），不过因为LSTM处理的是序列数据，所以在使用BP的时候需要将整个时间序列上的误差传播回来。LSTM本身又可以表示为带有循环的图结构，也就是说在这个带有循环的图上使用反向传播时我们称之为BPTT（back-propagation through time）。下面我们通过图3和图4来解释BPTT的计算过程。从图3中LSTM的结构可以看到，当前cell的状态会受到前一个cell状态的影响，这体现了LSTM的recurrent特性。同时在误差反向传播计算时，可以发现h(t)的误差不仅仅包含当前时刻T的误差，也包括T时刻之后所有时刻的误差，即back-propagation through time的含义，这样每个时刻的误差都可以经由h(t)和c(t+1)迭代计算。

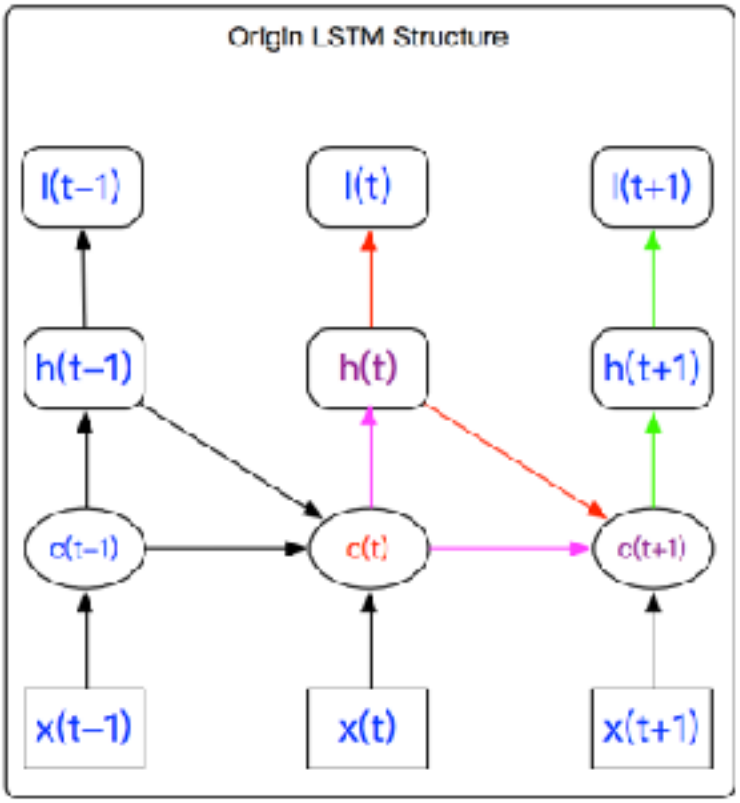

图3 LSTM网络示意图

为了直观地表示整个计算过程，在参考神经网络计算图的基础上，LSTM的计算图如图4所示，从计算图上面可以清晰地看出LSTM的forward propagation和back propagation过程。如图，H(t-1)的误差由H(t)决定，且要对所有的gate layer传播回来的梯度求和，c(t-1)由c(t)决定，而c(t)的误差由两部分，一部分是h(t)，另一部分是

c(t+1)。所以在计算c(t)反向传播误差的时候，需要传入h(t)和c(t+1)，而h(t)在更新的时候需要加上h(t+1)。这样就可以从时刻T向后计算任一时刻的梯度，利用随机梯度下降完成权重系数的更新。

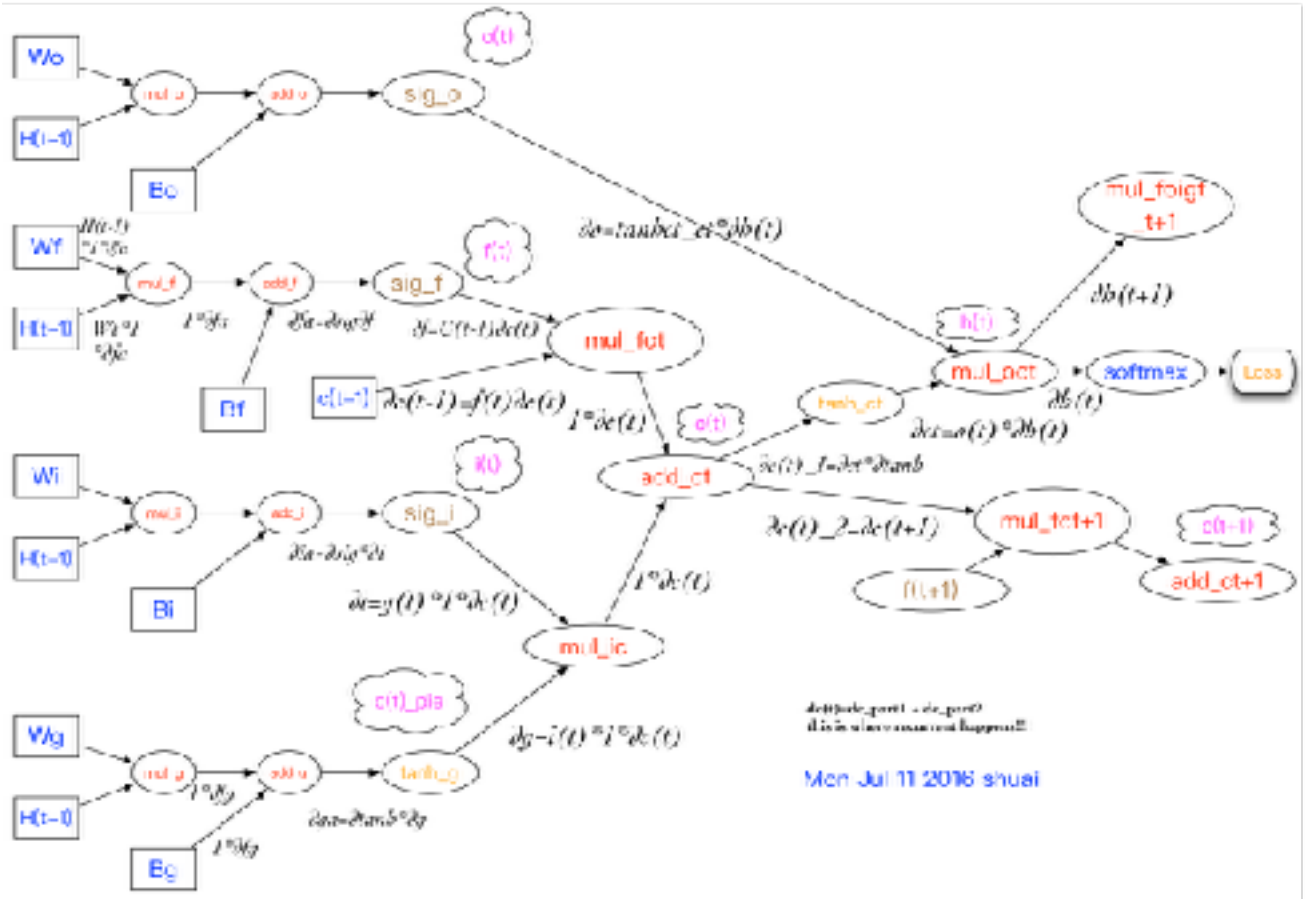

图4 BPTT示意图

四、LSTM算法的一些变形

LSTM算法的变形有很多，最主要的有两种，分别如下：

a）GRU

LSTM算法的变形里面GRU（Gated Recurrent Unit）是使用最为广泛的一种，最早由Cho等人于2014年提出[7]。GRU与LSTM的区别在于使用同一个门限来代替输入门限和遗忘门限，即通过一个“更新”门限来控制cell的状态，该做法的好处是计算得以简化，同时模型的表达能力也很强，所以GRU也因此越来越流行。

b）Peephole LSTM

Peephole LSTM由Gers和Schmidhuber在2000年提出[8]，Peephole的含义是指允许当前时刻的门限Gate“看到”前一时刻cell的状态，这样在计算输入门限，遗忘

门限和输出门限时需要加入表示前一时刻cell状态的变量。同时，另外一些Peephole LSTM的变种会允许不同的门限“看到”前一时刻cell的状态。

不同的研究者提出了许多LSTM的改进，然而并没有特定类型的LSTM在任何任务上都能够由于其他变种，仅能在部分特定任务上取得最佳的效果。更多LSTM算法的改进可以参考《Deep Learning》一书中的第10.10章节。

五、实际应用

在交易反作弊的应用中，我们使用了LSTM算法来识别作弊交易。因为交易反作弊实际上可以归纳为一个二分类的问题，所以我们使用的是仅包含一个隐层的LSTM二分类网络（一阶LSTM网络的表达能力已经足够强，诚然仅包含一个隐层的网络也许不能称之为深度学习算法）。

a）数据

在构建端到端的机器学习应用的过程中，最基本也是最重要的就是训练数据集的获得经过一年多的积累已经沉淀下来大量的***标签数据，而LSTM算法所需的序列数据的选择则是较为困难的，因为确定序列数据的一个约束条件是数据本身要能够描述***的行为。从数据的丰富程度考虑，最终我们选择了***数据，并根据这两类序列数据训练了两个LSTM网络。

b）模型

LSTM模型包含如下几个部分：Embedding层，LSTM层，Sigmoid输出层，损失函数采用binary交叉熵，其中Embedding层的尺寸根据词典数据的大小而变动。

c）训练

从性能以及现有的计算资源考虑，最终我们采用了以Theano为计算核心的Keras深度学习框架完成了模型的训练。核心计算设备为NVIDIA K40，训练数据在五百万左右，模型参数十万左右。使用随机梯度下降进行训练，其中minibatch设置为1024，迭代次数为10轮，学习率的自动更新采用adam算法，dropout参数设置为0.4。基于GPU的LSTM网络训练耗时总计10个小时左右。

d）结果

训练集的正负样本比例约为1：3，交叉验证的比例约为2：1，基于***训练与验证的准确率分别为0.9023和0.9017；基于***训练与验证的准确率分别为0.9608和0.9632，人工抽样校验通过，达到上线标准。

六、总结

本文回顾了LSTM算法诞生的背景与原因，详细分析了LSTM网络训练过程中使用BPTT的细节，并介绍了LSTM算法在***中的应用。从目前模型上线后的表现效果看来，LSTM算法的表现超过了传统算法（SVM，RF，GBDT等等），也从侧面印证了深度学习的强大之处，值得算法同学更多的探索。诚然，深度学习这一新兴的机器学习领域内包罗万象，上述的理解仅是个人的一些涉猎和体会，如有纰漏在所难免，欢迎对深度学习感兴趣的同学一起探讨，共同提高。

七、感谢：

本文主要参考了如下资料，深表感谢：

a）Understanding LSTM Networks，http://colah.github.io/posts/2015-08-Understanding-LSTMs/；

b）Deep Learning，Ian Goodfellow Yoshua Bengio and Aaron Courville，Book in preparation for MIT Press，2016；

c）Simple LSTM，http://nicodjimenez.github.io/2014/08/08/lstm.html。

参考资料：

【1】https://googleblog.blogspot.com/2012/06/using-large-scale-brain-simulations-for.html;

【2】http://image-net.org/challenges/LSVRC/2012/supervision.pdf;

【3】https://en.wikipedia.org/wiki/AlphaGo;https://deepmind.com/research/alphago/;http://sports.sina.com.cn/go/2016-09-13/doc-ifxvukhx4979709.shtml;

【4】https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf;

【5】http://www.deeplearningbook.org/;

【6】http://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735#.V9fMNZN95TY;

【7】http://arxiv.org/pdf/1406.1078v3.pdf;

【8】ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf.