神经网络语言模型
及其在语音识别中的应用进展

刘权
quanliu@mail.ustc.edu.cn
2014年01月14日

Outline

- 1. 统计语言模型概述及传统建模方法
- 2. 领域研究热点——神经网络语言模型
- 3. 神经网络语言模型在语音识别中的应用进展
- 4. 总结

1. 统计语言模型概述及传统建模方法

1.1 统计语言模型概述

■ **语言**在信息论中的定义(Abramson N. Information theory and coding. McGraw-Hill, 1963.)

语言是一个产生字符序列的源

“Language is considered an information source, which emits a sequence of symbols from a finite alphabet (the vocabulary).”

■ **Statistical Language Model** (SLM)：预测字符(词)序列产生的概率

词序列 $w_1^m = w_1, w_2, \ldots, w_m$

预测概率 $P(w_1^m) = \prod_{i=1}^{m} P(w_i \mid w_1^{i-1})$

■ 语言模型核心问题： $P(w_i \mid w_1^{i-1}) \Leftrightarrow P(w_i \mid h_i), h_i = history\ of\ word\ w_i$

■ 语言模型的性能度量

➢ 基本指标：困惑度

$$perplexity = \exp(-\frac{1}{N}\sum_{w_1^n} \log P(w_n \mid w_1^{n-1}))$$

➢ 任务指标：语音识别中的WER，机器翻译中的BLEU等

✓两个语言模型的困惑度，仅当都使用相同的词汇时才可以比较。

✓ 困惑度与实际任务中的指标并没有等价关系

1.2 传统建模方法
N-Gram Language Model

建模方法1 —— N元语法语言模型

- 思想：N-1阶马尔科夫模型，使用前N-1个单词估计下一个单词的概率

  发展于1980s，IBM的Thomas. J Watson研究中心应用N元语法在ASR中取得很大成绩

- 数学表示

$$P(w_i \mid w_1^{i-1}) = P(w_i \mid w_{i-N+1}^{i-1}) = \frac{C(w_{i-N+1}^{i-1}, w_i)}{C(w_{i-N+1}^{i-1})}$$

$C(w_i^j)$ 词序列在语料中的出现次数

- N-Gram语言模型的主要问题
  - 数据稀疏问题，导致存在大量的零概率N元语法
  - 不能使用长距离上下文，导致过低估计在语料中非临近出现符号序列的概率

- 问题解决
  - Discounting：Good-Turing，Kneser-Ney
  - Back-off：Katz

1.2 传统建模方法
Cache Language Model

建模方法2 —— Cache Language Model

■ 思想：从**动态的历史数据**中，训练Cache模型，与静态模型插值

Jelinek F, Merialdo B, Roukos S, et al. A Dynamic Language Model for Speech Recognition. HLT. 1991.

■ 数学表示

$$P_{adaptive}(w_i \mid w_1^{i-1}) = \lambda P_{static}(w_i \mid w_1^{i-1}) + (1-\lambda) P_{cache}(w_i \mid w_1^{i-1})$$

■ 核心问题

1. Cache模型确定

   训练数据通常可以由话题分/聚类分配，但要设定训练规模

2. 插值权重确定

   Cache模型的插值虽然操作简单，但权重的调整需额外的验证数据(held-out data)，若验证数据与实际的测试数据不匹配时，效果可能适得其反。

1.2 传统建模方法
Class Based Language Model

建模方法3 —— Class Based Language Model

- 思想：建立词类集合，将对词的概率估计拆分成与词类相关的两部分概率

Brown P F, Desouza P V, Mercer R L, et al. Class-based n-gram models of natural language. Computational linguistics, 1992.

- 数学表示

$$P_{new}(w_i \mid w_1^{i-1}) = P(c(w_i) \mid w_1^{i-1}) P(w_i \mid w_1^{i-1}, c(w_i))$$

- 应用领域

解决数据稀疏问题

神经网络语言建模中，为了提升速度而借鉴了Class Based的思想

- 核心问题

词类的建立(依赖语法知识)

2. 领域研究热点——神经网络语言模型

- 2.1 Feedforward Neural Network LM
- 2.2 Recurrent Neural Network LM

2.1 Feedforward Neural Network LM

前向型神经网络—语言模型(FFNN-LM)

■ 观点产生

➢ 1991年Risto Miikkulainen在UCLA提出PDP并行分布式处理网络用于NLP中

Miikkulainen R, Dyer M G. *Natural language processing with modular PDP networks and distributed lexicon*. Cognitive Science, 1991.

➢ 2000年**徐伟(now in IDL@Baidu)**在CMU提出神经网络用于语言建模的设想并做了简易实验

Xu W, Rudnicky A I. *Can artificial neural networks learn language models?*. CMU, 2000.

■ 经典论文

➢ **NPLM (Yoshua Bengio)**

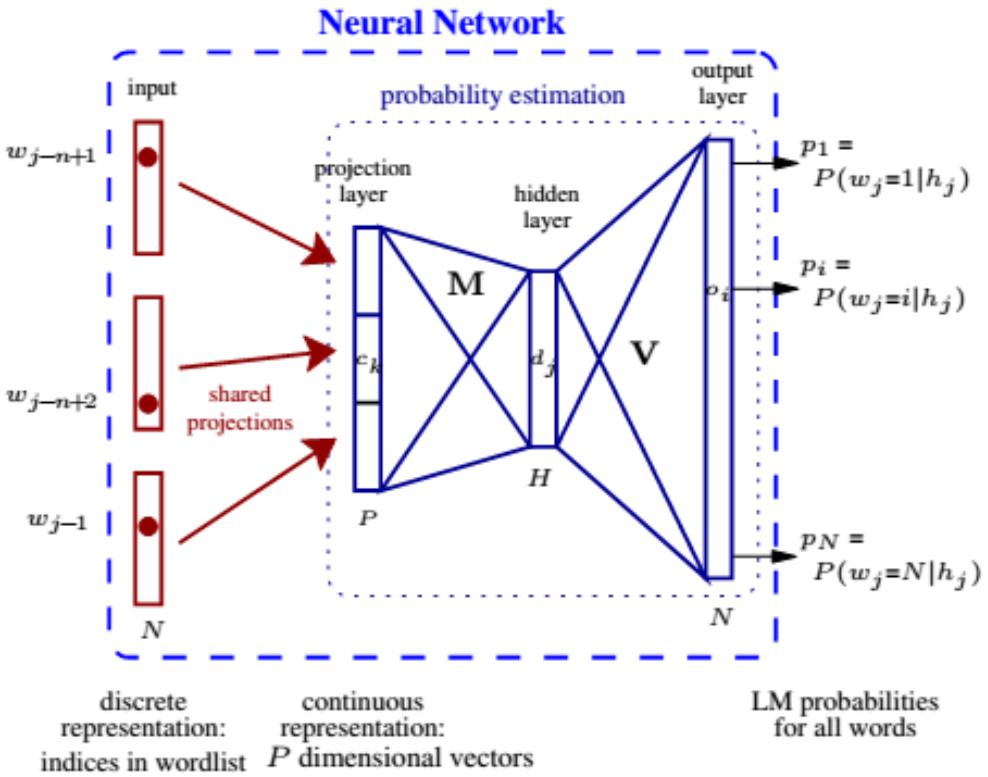
Y. Bengio, R. Ducharme, and P. Vincent, et al. *A Neural Probabilistic Language Model*. Journal of Machine Learning Research, 2003.

➢ **CSLM (Holger Schwenk)**

Schwenk H. *Continuous space language models*. Computer Speech & Language, 2007.

2.1.1 FFNN-LM 网络结构说明

➢ 网络结构示意图



单隐层FFNN-LM网络结构

➢ 核心要素

■ **Input ~ History**

$$h_j = (w_{j-1}, w_{j-2}, ..., w_{j-n+1})$$

■ **Output ~ Probability**

$$p_i = P(w_j = i \mid h_j) \quad \forall i \in [1, V]$$

■ **Objective Function**

$$E = \sum_{i=1}^{V} t_i \log p_i + \varepsilon(\theta)$$

■ **Notice**

✓ **1. 输入和输出层均为 1-of-V 向量；**

✓ **2. 输入到隐层之间，通常含有一个投影层，将每个词用低维的连续型向量表示。**

⇨ **Word Embedding Problem**

推荐https://code.google.com/p/word2vec/

2.1.2 FFNN-LM 模型训练

➢ 参数说明 $\theta = (\mathbf{b}, \mathbf{d}, \mathbf{U}, \mathbf{H}, \mathbf{C})$

维度说明：n = n-gram大小，m = 词向量维度，h = 隐层规模，V = 词典规模

b —— 输出偏置；**d** ——隐藏层偏置；**C** —— 词特征映射矩阵；

U —— 隐藏层到输出层的权重；**H** —— 词特征层到隐藏层的权重；

参数规模 = $(n-1)\times m\times h + h + (m+h+1)\times V$

■ **Feed Forward**

词映射输入 ⇨ $\mathbf{x} = (C(w_{j-1}), C(w_{j-2}), ..., C(w_{j-n+1}))$

前向传播 ⇨ $\mathbf{y} = \mathbf{b} + \mathbf{U}\tanh(\mathbf{d} + \mathbf{H}\mathbf{x})$

softmax ⇨ $P(w_j \mid w_{j-1}, ..., w_{j-n+1}) = \frac{e^{y_{w_j}}}{\sum_i e^{y_i}}$

高维的词典向量将导致训练极为耗时

IBM在一台8核机器上，使用50Million words规模的语料，词典规模为20000，训练一个词向量维度120，隐层数800的6-gram NNLM，耗时50天。

■ **Back Propagation**

$\theta \leftarrow \theta + r\frac{\partial E}{\partial \theta}$，$r = learning\ rate$

2.1.3 FFNN-LM 模型训练的提速策略

➢ 两类提速策略

提速策略A——基本方法

- FFNN只对小规模的常用词（shortlists）的建模
- 以Bunch为单位训练网络（性能表现不佳）
- 分块计算权重及梯度，优化矩阵乘法（GUDA, Intel CPU上使用SSE指令的MKL库等）

提速策略B——结构方法

基于Class-based思想，将输出层分类层次化后估计概率 $P(y \mid x) = P(c(y) \mid x)P(y \mid x, c(y))$

✓ Morin F, Bengio Y. Hierarchical probabilistic neural network language model. 2005.

✓ Kuo H K, Arisoy E, Emami A, et al. Large Scale Hierarchical Neural Network Language Models. INTERSPEECH. 2012.

IBM2012年使用该策略在rt04上的速度提升性能

	d	h	WER (%) + n-gram	training time
n-gram LM(350M word)			13.0	-
NNLM-55M	120	800	12.3	50 days
hNNLM-55M	120	800	12.3	2 days
hNNLM-350M	180	1200	12.0	24 days

2.1.3 FFNN-LM 模型训练的提速策略
并行网络训练

➢ 前向传播（预测词 w_j）

（1）word to vector

$$\mathbf{x} = (x(1), x(2), ..., x(n-1)),\ x(k) \leftarrow C(w_{j-k})$$

（2）传播至隐层

$$\mathbf{o} \leftarrow \mathbf{d} + \mathbf{Hx}$$

$$\boldsymbol{a} \leftarrow \tanh(\mathbf{o})$$

（3）激活对第 i 块的输出，计算softmax统计量

for block i $\quad s_i = 0$

for m in block i $\quad y_m = b_m + \boldsymbol{a}U_m$

$$p_m = e^{y_m}$$

$$s_i = s_i + p_m$$

（4）统计量合并共享，取softmax输出 $S = \sum_i s_i$

for m in block i $\quad p_m = p_m / S$

2.1.3 FFNN-LM 模型训练的提速策略
并行网络训练

- 反向传播

（1）清空梯度 $\frac{\partial E}{\partial \boldsymbol{a}}, \frac{\partial E}{\partial \mathbf{x}}$

（2）从输出层开始BP

for m in block i $\quad \frac{\partial E}{\partial y_m} = 1_{m==j} - p_m \qquad b_m = b_m + r\frac{\partial E}{\partial y_m} \qquad \frac{\partial E}{\partial \boldsymbol{a}} = \frac{\partial E}{\partial \boldsymbol{a}} + \frac{\partial E}{\partial y_m} U_m$

$$\Rightarrow \quad U_m = U_m + r\frac{\partial E}{\partial y_m}\boldsymbol{a}$$

（3）梯度合并共享，BP更新隐层权重

for l in hidden layer $\quad \frac{\partial E}{\partial o_l} = (1 - a_l^2)\frac{\partial E}{\partial a_l} \qquad \frac{\partial E}{\partial \mathbf{x}} = \frac{\partial E}{\partial \mathbf{x}} + \mathbf{H}^{\mathrm{T}}\frac{\partial E}{\partial \mathbf{o}}$

$$\Rightarrow \quad \mathbf{d} = \mathbf{d} + r\frac{\partial E}{\partial \mathbf{o}} \qquad \mathbf{H} = \mathbf{H} + r\frac{\partial E}{\partial \mathbf{o}}\mathbf{x}^{\mathrm{T}}$$

（4）更新特征向量矩阵

for k from 1 to n-1 $\quad C(w_{j-k}) = C(w_{j-k}) + r\frac{\partial E}{\partial x(k)}$

2.1.4 FFNN-LM 在语音识别系统中的应用

- 解码器需要什么？= 给定context下的LM概率

A LVCSR decoder usually requests many different **LM probabilities for the same context** when expanding the search trees and during LM look-ahead. (Holger Schwenk)

- 嵌入识别系统中的瓶颈

实际中，给定context，N-Gram LM直接寻找概率（a table look-up using hashing techniques）；而NNLM需要前向传播整个网络，为了得到一个输出节点上的概率信息，需要将所有节点求出，并取softmax，非常耗时.

- 1. shortlist ： 少量词的概率由NNLM估计

Schwenk H. **Continuous space language models** [J]. Computer Speech & Language, 2007.

$$\hat{P}(w_t|h_t) = \begin{cases} \hat{P}_N(w_t|h_t, L) \cdot \hat{P}_B(h_t|L) & if \ \ w_t \in short-list \\ \hat{P}_B(w_t|h_t) & else \end{cases}$$

$$P_B(h_t|L) = \sum_{w \in short-list} \hat{P}_B(w_t|h_t)$$

L代表一个事件：当前词在short-list表中

- 2. rescoring：不用于一遍解码，而是在基线系统解码结果的基础上，进行得分重估

2.1.5 FFNN-LM 近期研究进展

■ 近期研究进展(mainly concentrated by **IBM**)

✓ **DNNLM**

Arisoy E, Sainath T N, Kingsbury B, et al. *Deep neural network language models*. NAACL-HLT 2012.

✓ **HNNLM**

Kuo H K, Arisoy E, Emami A, et al. *Large Scale Hierarchical Neural Network Language Models*. INTERSPEECH. 2012

■ **IBM work on DNN-LM (WSJ task)**

Corpus： 1993 WSJ (CSR-III Text Corpus)
900K句话，23.5M个词
Baseline : 4-gram，词典大小20K
DNNLM：4-gram，词典大小10K

测试说明：ASR-WER中，DNNLM用于rescore)

实验结果

结论：NNLM中，将隐层层数增大，有助于提升PPL和WER，但效果并不明显（考虑计算代价）。PPL性能差于RNN LM（Mikolov）

Models	Perplexity	WER(%)
4-gram LM	114.4	22.3
DNN LM: h=500, d=30		
with 1 layer (NNLM)	115.8	22.0
with 4 layers	108.0	21.6
DNN LM: h=500, d=60		
with 1 layer (NNLM)	109.3	21.5
with 3 layers	105.0	21.3
DNN LM: h=500, d=120		
with 1 layer (NNLM)	104.0	21.2
with 3 layers	102.8	20.8
Model M (Chen, 2008)	99.1	20.8
RNN LM (h=200)	99.8	-
RNN LM (h=500)	83.5	-

2.1.5 FFNN-LM 近期研究进展
IBM work on DNN-LM

IBM work on DNN-LM

■ **增大层数的问题：易陷入局部最小值（local minima）**

⇨ **Pre-training**

➢ **Generative Pre-training (Hinton, et al., 2006)**

Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006.

➢ **Discriminative Pre-training, DPT (Seide et al., 2011)**

Seide F, Li G, Chen X, et al. Feature engineering in context-dependent deep neural networks for conversational speech transcription[C]. ASRU2011.

■ **DPT for DNNLM**

DPT 是一个“layer-wise BP”的过程，训练步骤如下

➢ 1. 训练单隐层DNN网络：随机初始化后利用部分监督信息训练。

➢ 2. 替换softmax层：将当前网络的softmax层用另一随机初始化的隐层+softmax层替换。

➢ 3. BP训练：在新的网络上，继续进行BP。

➢ 4. 迭代：按步骤2、3不断迭代，直到添加的隐层数量到设定值

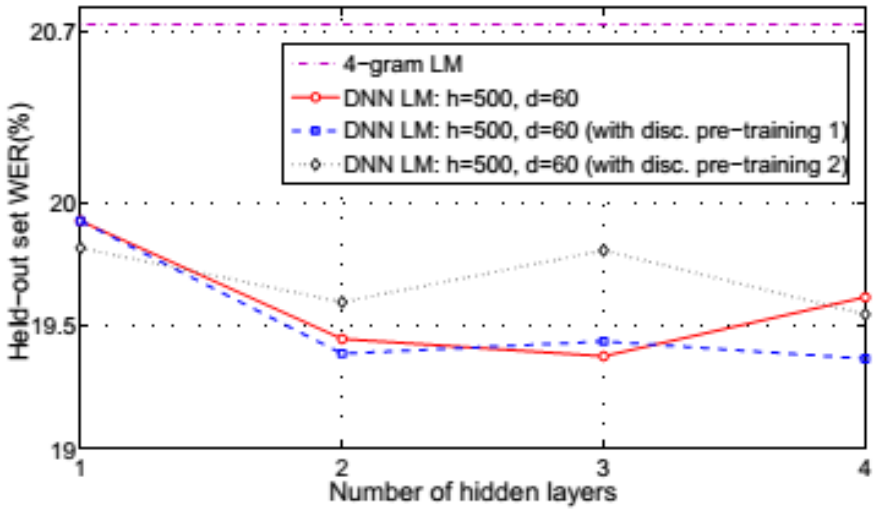
2.1.5 FFNN-LM 近期研究进展
IBM work on DNN-LM

IBM work on DNNLM

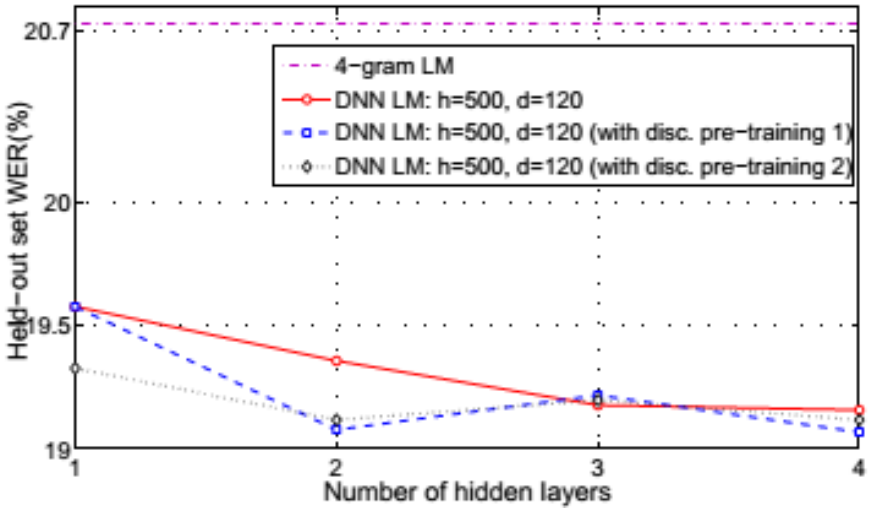
■ DPT for DNNLM result （在单独取出的数据上测试）

Pre-training 1：权重初始化完全依照DPT结果

Pre-training 2：除输出层权重外，其余权重的初始化完全依照DPT结果



隐层数=500，词向量位数=60



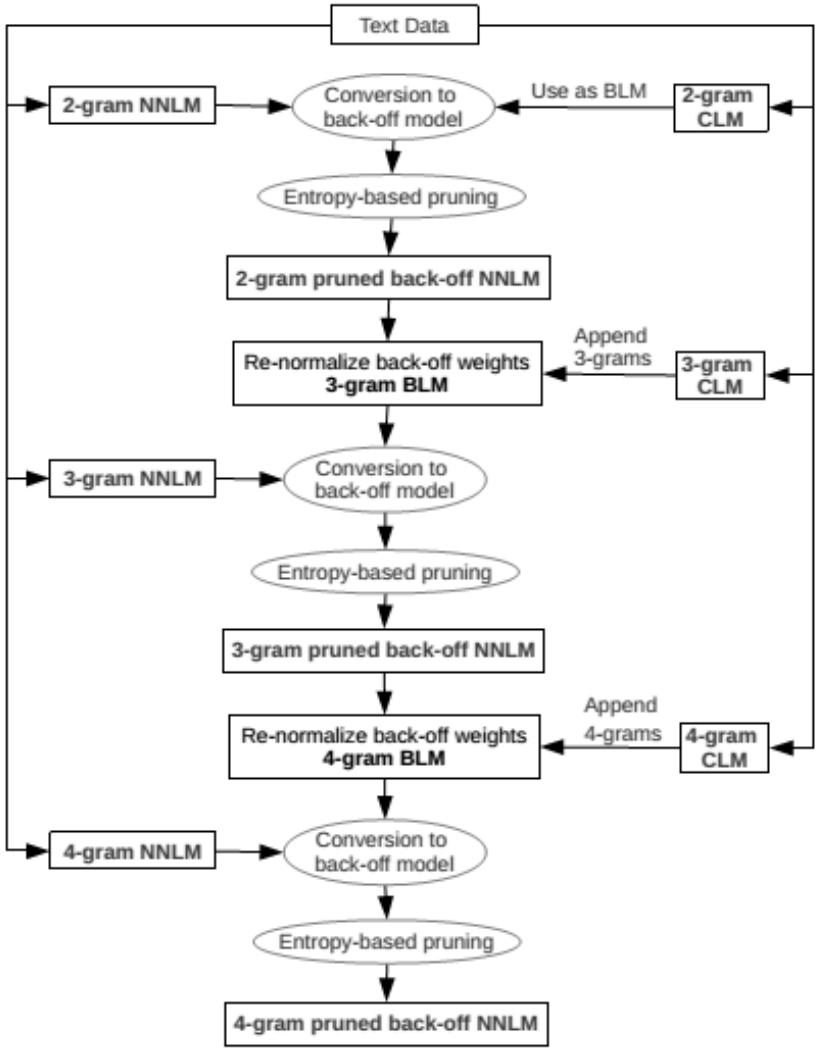
隐层数=500，词向量位数=120

IBM的结论

Pre-training did not give consistent gains for models with different number of hidden layers. We need to investigate discriminative pretraining and other pre-training strategies further for DNN LMs.

2.1.5 FFNN-LM 近期研究进展
IBM work on 《NNLM to Back-off LM》

- **IBM@ICASSP2013: 将FFNNLM转化为传统语言模型（back-off LM），再用于解码**



- **IBM T.J. Watson Research Center, ICASSP2013.**

Arisoy E, Chen S F, Ramabhadran B, et al. Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition.

- **实验结果（English Broadcast News Task）**

Model	WER (%)
Baseline LM	14.7
+ rescore lattices with 4-gram NNLM	13.3
+ rescore lattices with 6-gram NNLM	13.2
NNLM converted into a back-off LM	13.7
+ rescore lattices with 4-gram NNLM	13.0
+ rescore lattices with 6-gram NNLM	12.8

- **实验结论：较好的提升了系统性能**

2.1.6 FFNN-LM 工具资源

✓ CSLM 开源工具（2013）

LIUM在INTERSPEECH2013上发表的一篇文章，提出一个神经网络语言建模的开源工具

Schwenk H. CSLM-a modular open-source continuous space language modeling toolkit. Interspeech. 2013.

cslm_train this is the main tool to train an CSLM. The main command line arguments are the network to be trained, the training and development data as well various learning parameters (learning rate, number of iterations, etc).

cslm_eval evaluation of an existing CSLM on some development data.

nbest rescoring of n-best lists with a back-off or CSLM.

cslm_tool lattice rescoring with a back-off or CSLM.

In addition, the following tools are provided:

text2bin convert text files to the internal binary representation of the CSLM toolkit.

nn_info display information on a neural network.

➢ **Atrtractive：支持FFNN-LM训练，及rescore**

2.2 Recurrent Neural Network LM

2.2 Recurrent Neural Network LM Outline

技术起源：From **fixed size of context** To **long-term dependencies**

- 前向型NNLM(Bengio2003，Schwenk2005)，其估计概率的“历史”与n-gram相同，即由前 n-1 个词估计当前词的概率
- 构建历史长度不受限定，具有更强的记忆能力的网络

通常认为，网络的记忆能力存储在隐层中

Recurrent Neural Network (循环神经网络)

- 1. Simple Recurrent Neural Network （ Elman Network ）

  Mikolov在Brno博士期间的主要工作

- 2. Long Short-Term Memory

  TUM大学的提出，Toronto大学的Alex Grave

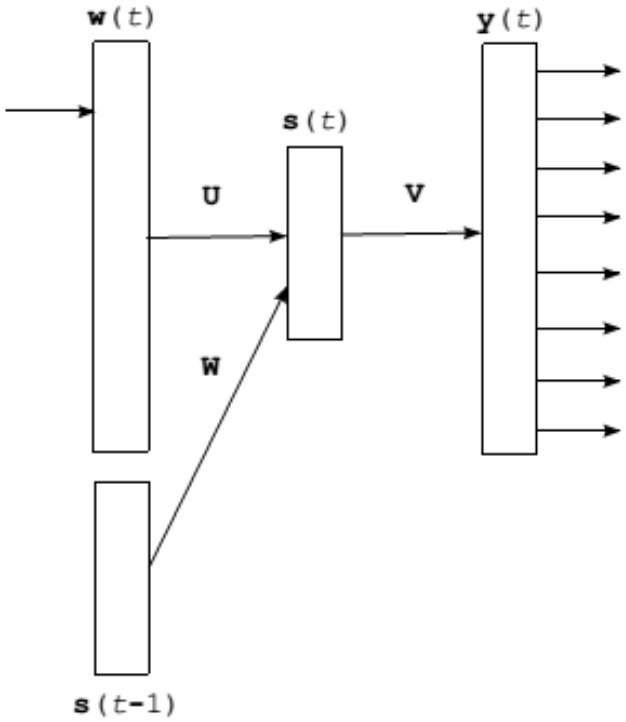
2.2.1 Elman Recurrent Neural Network

Elman Recurrent Neural Network简介

■ 结构提出

JL. Elman1990年在文章《Finding structure in time》中提出，现已被广泛应用（>5900引用量）

■ 网络结构：保留隐层信息至输入层，用于下一时刻对隐层的激活



- 用 t 表示时序，在LM中对应单词的前后顺序
- 当前输入 $w(t)$
- 当前隐层信息 $s(t)$
- 前一时刻隐层 $s(t-1)$
- 输出 $y(t)$

Tomas Mikolov
2012~now: researcher
Google Research
2007~2012: Phd student
BUT Speech@FIT
2011: visit student
Univ. Montreal, Y.Bengio

RNNLM

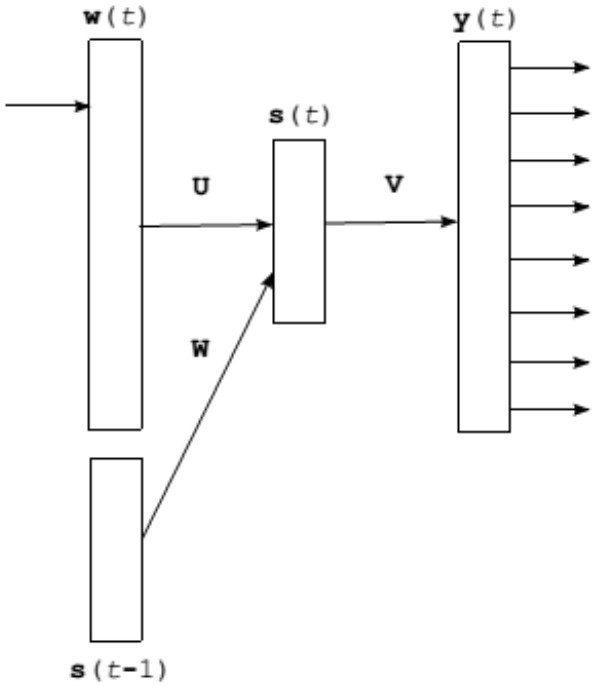
2.2.1 Elman Recurrent Neural Network
—— Mikolov work

Tomas Mikolov 's work

- Mikolov T. Statistical Language Models Based on Neural Networks. Ph. D. thesis, BUT, 2012.
- Mikolov T, Kombrink S, et al. Extensions of recurrent neural network language model. ICASSP 2011.
- Mikolov T, Karafi á t M, et al. Recurrent neural network based language model. INTERSPEECH. 2010.

结构说明

1. 输入和输出层为1-of-V 高维向量，V为词典规模
2. 单隐层，规模定义为H，循环连接



- **前向传播**

输出为基于输入和隐层的记忆得出的预测概率

$$\mathbf{s}(t) = f(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1))$$

$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{s}(t))$$

$$s_j(t) = f\left(\sum_i w_i(t)u_{ji} + \sum_l s_l(t-1)w_{jl}\right)$$

$$y_k(t) = g\left(\sum_j s_j(t)v_{kj}\right)$$

$$f(z) = \frac{1}{1+e^{-z}},\ g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

RNNLM 模型训练

RNN-LM网络训练步骤

目标：最大化训练数据的似然概率 $\sum_{t=1}^{N} \log y_{l_t}(t)$

N 为训练语料中的样本词数

l_t 为样本 t 对应的word Index

- 输出层梯度（根据交叉熵准则），d_t 为目标词

$$e_o(t) = d(t) - y(t)$$

- 隐层->输出层的权重更新，r 为学习速率

$$\mathbf{V}(t+1) = \mathbf{V}(t) + r \cdot \mathbf{s}(t)\mathbf{e}_o(t)^{\mathrm{T}}$$

- 梯度传递到隐层：

$$\mathbf{e}_h(t) = d_h(\mathbf{e}_o(t)^{\mathrm{T}}\mathbf{V}, t)$$

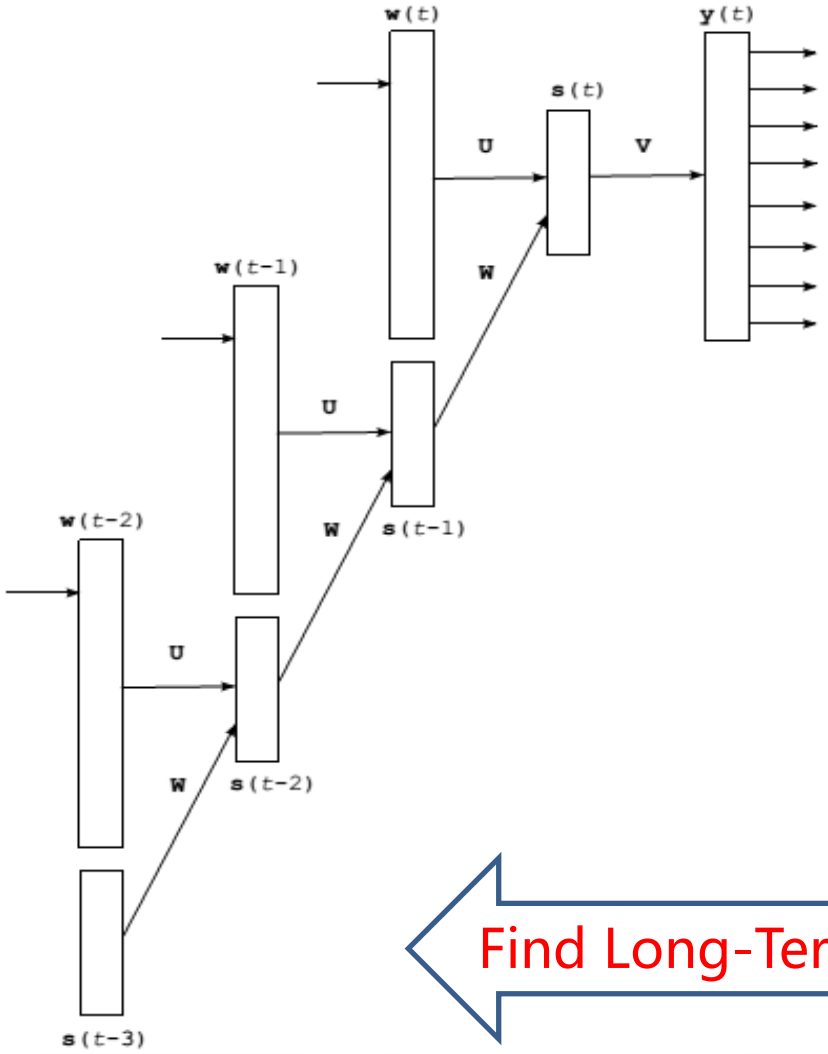
$$d_{hj}(x,t) = xs_j(t)(1-s_j(t))$$

- 输入层->隐层权重更新

$$\mathbf{U}(t+1) = \mathbf{U}(t) + r \cdot \mathbf{w}(t)\mathbf{e}_h(t)^{\mathrm{T}} \qquad \mathbf{W}(t+1) = \mathbf{W}(t) + r \cdot \mathbf{s}(t-1)\mathbf{e}_h(t)^{\mathrm{T}}$$

■ 如何考虑更长的历史？

Recurrent T times



Find Long-Term History

RNNLM —— BPTT

■ 估计更长历史信息：Backpropagation Through Time, BPTT算法

Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990.

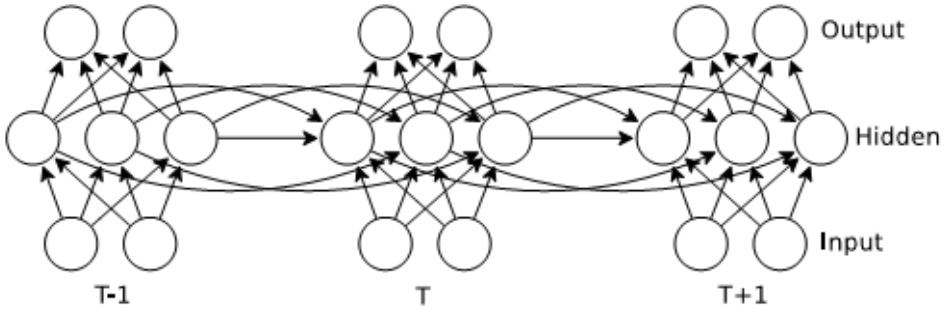
➢ **BP循环回退计算误差**

$$\mathbf{e}_h(t-\tau-1) = d_h(\mathbf{e}_h(t-\tau)^{\mathrm{T}}\mathbf{W}, t-\tau-1)$$

➢ **回退完成后更新权重**

$$\mathbf{U}(t+1) = \mathbf{U}(t) + \sum_{z=0}^{T_b} r \cdot \mathbf{w}(t-z)\mathbf{e}_h(t-z)^{\mathrm{T}}$$

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \sum_{z=0}^{T_b} r \cdot \mathbf{s}(t-z-1)\mathbf{e}_h(t-z)^{\mathrm{T}}$$



RNN网络示意图

注意

1. 循环回退的次数 T_b 可在模型训练的时候设定，回退的次数并不是越大越好.
2. 回退前需要在程序中保留前T个时间段的隐层状态.
3. 回退过程中不更新权重，待回退完成后再更新.

RNNLM 计算复杂度及改进策略

- RNNLM计算复杂度

$$O = (1+H)\times H \times T_b + H \times V$$

词典规模太大，导致训练或测试的计算代价非常高

- 改进策略

Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model. ICASSP, 2011.

Mikolov T, Deoras A, Povey D, et al. Strategies for training large scale neural network language models. ASRU2011.

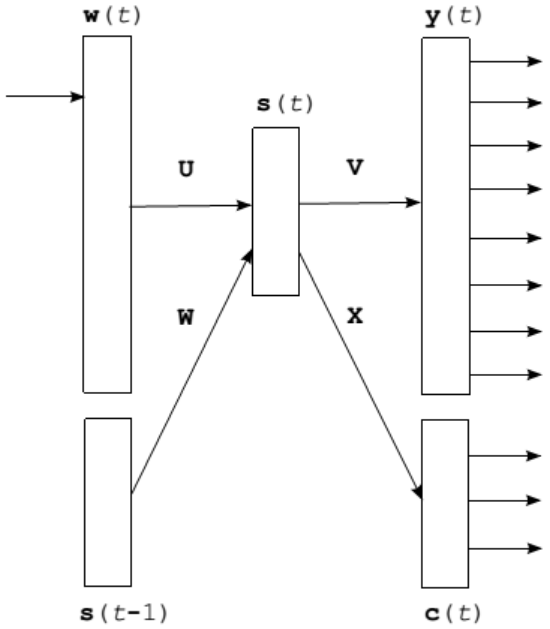
输出层加一个词类层，估计联合条件概率

$$P(w_i \mid history) = P(c_i \mid history)P(w_i \mid c_i)$$

词类规模为C时，计算复杂度转变为

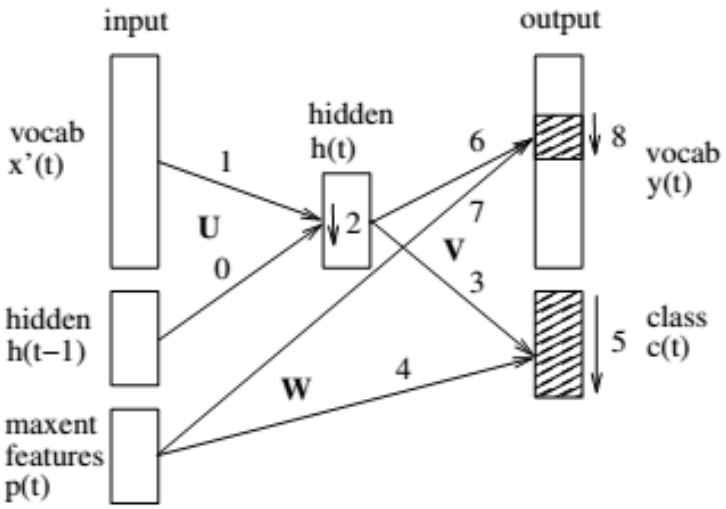
$$O = H \times H \times T_b + H \times (C + \frac{V}{C})$$

词的分类Mikolov使用最简单的按词频分类，效果较佳



RNNLM 补充说明

- RNNLM如何避免Over-fitting：添加额外的输入特征，将其与输出层直连



思路：在小数据集上的RNNLM极易过拟合，考虑引入额外特征。

额外特征

- 最大熵特征：最大熵模型等价去除隐层的NNLM
- N-Gram特征（RNNLM Toolkit中实现）

- 相关资源

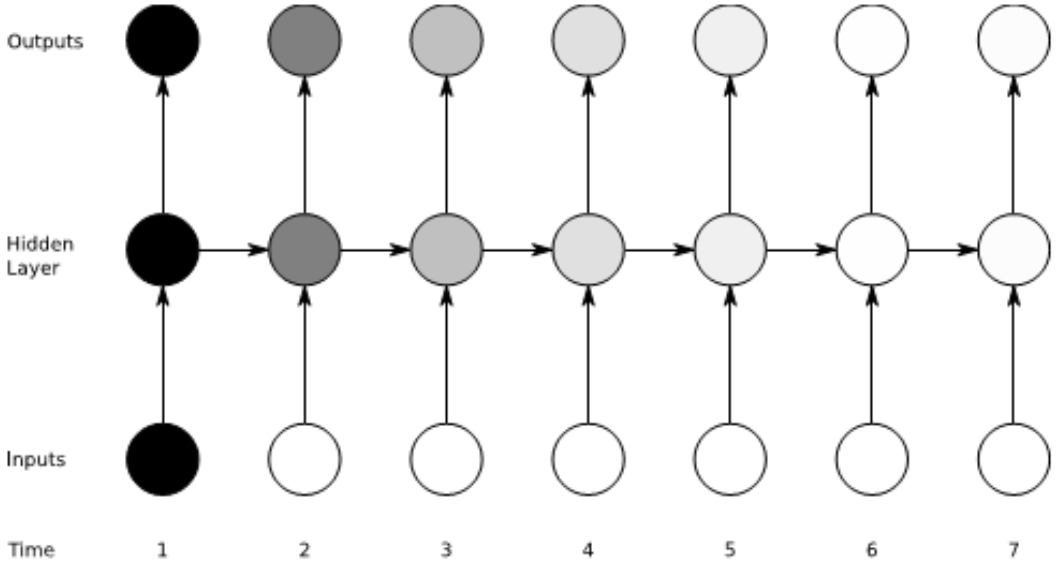
RNNLM Toolkit http://www.fit.vutbr.cz/~imikolov/rnnlm/

该工具已经包含到Kaldi开源系统中，用以完成RNN语言模型训练，lattice-rescore，但并没有在一遍解码中实现。

重新审视Elman RNN网络的记忆能力

- 从BPTT算法角度，误差函数的梯度在每次回退（随时间展开）的时候，都要乘以相应的调整系数，这个系数根据不同任务可能大于1或小于1，最终导致vanishing gradient problem，即梯度随着时间呈指数级减小/爆炸。

- Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies [J]. 2001.
- Bengio, Y., Simard, P., Frasconi, P., “Learning long-term dependencies with gradient descent is difficult”，IEEE Transactions on Neural Networks 5 (1994).



vanishing gradient problem 示意

Toward vanishing gradient problem

解决vanishing gradient problem策略

- Non-gradient based training algorithms
  - **1. Simulated annealing and discrete error propagation**
  - **2. explicitly introduced time delays or time constants**

Lang K J, Waibel A H, Hinton G E. A time-delay neural network architecture for isolated word recognition[J]. Neural networks, 1990, 3(1): 23-43.

Giles I C L, Hanson S J, Cowan J D. Holographic Recurrent Networks[J].

  - **3. Hierarchical sequence compression**

Schmidhuber J. Learning complex, extended sequences using the principle of history compression[J]. Neural Computation, 1992, 4(2): 234-242.

  - **4. Hessian-Free**

Martens, J., Sutskever, I., "Learning Recurrent Neural Networks with Hessian-Free Optimization", Proc. of the 28th Int. Conf. on Machine Learning 2011

- **LSTM, Long Short-Term Memory**

改变网络结构，使调整系统固定为1，成功避免梯度消失问题，同时保持训练算法不变

2.2.2 Long Short-Term Memory

Long Short-Term Memory（LSTM）

- Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

author1 ：Sepp Hochreiter，TUM（慕尼黑工业大学）

Author2：Jürgen Schmidhuber， IDSIA（Swiss AI Lab）

Technische Universität München

- 手写识别领域获得成功（2009 won the ICDAR handwriting competition）

Refer to read

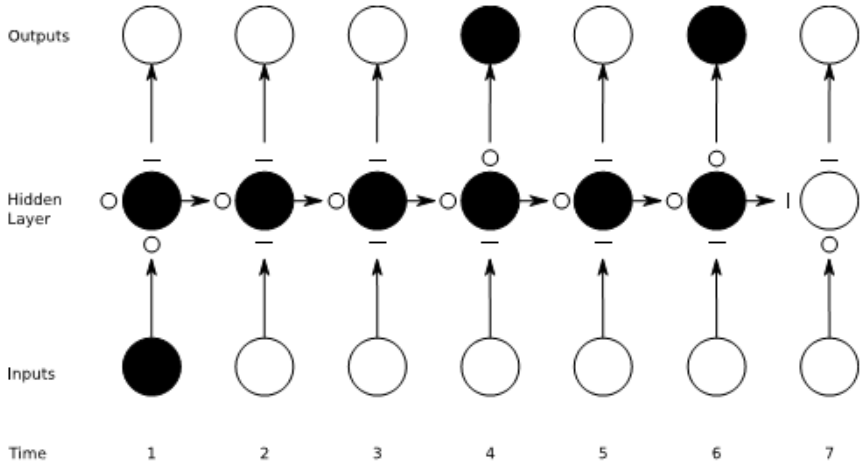
- 参阅 博士论文 Graves A. Supervised sequence labelling with recurrent neural networks[M]. Springer, 2012.
- Alex Graves. http://www.cs.toronto.edu/~graves/

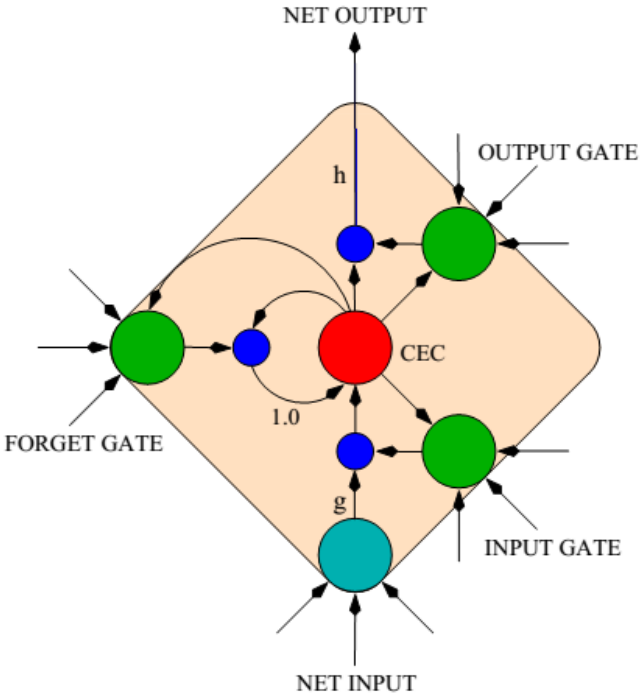
UNIVERSITY OF TORONTO

- RNNLIB开源代码（http://sourceforge.net/projects/rnnl/ ）

LSTM的原理和数学模型

- LSTM致力于解决Gradient vanish问题，提升网络的记忆能力。
- 设计：设计多门“开关”，保存并获取历史信息，该功能通过一个“memory block”实现，每个block是一个循环连接的子网“recurrently connected subnets”，整个网络由子网集合组成。
- 与之前Elman RNN不同的是，将隐层替换了





- 记忆能力说明：隐层对当前输入的记忆，可以通过关闭后续输入开关（input gate）得以保存，在需要调用记忆的时候，在某个时间点打开输出开关（output gate）即可。

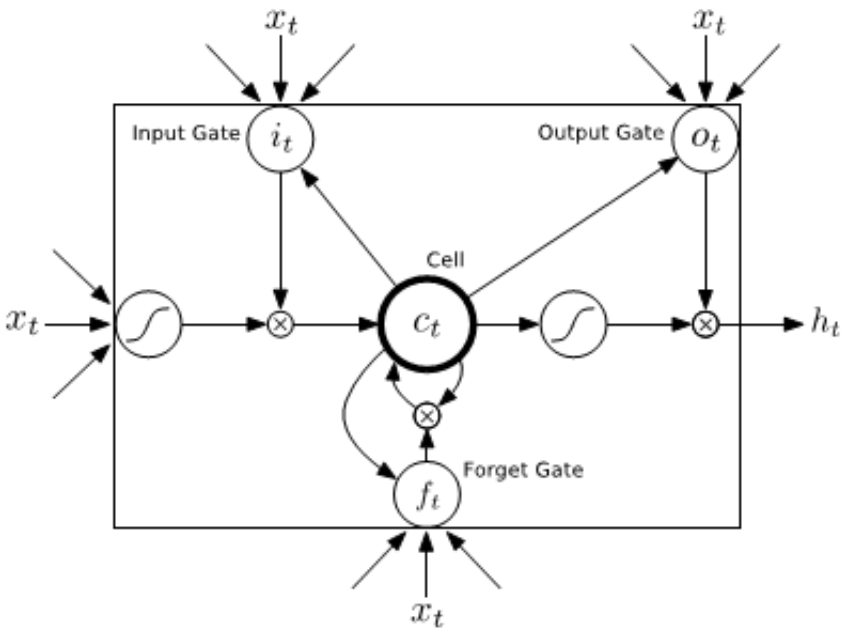
LSTM的原理和数学模型

- LSTM vs Elman RNN : 区别在于隐层节点

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$h_t = o_t \tanh(c_t)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$



更强的记忆控制能力

Output gate $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o)$

Forget gate $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$

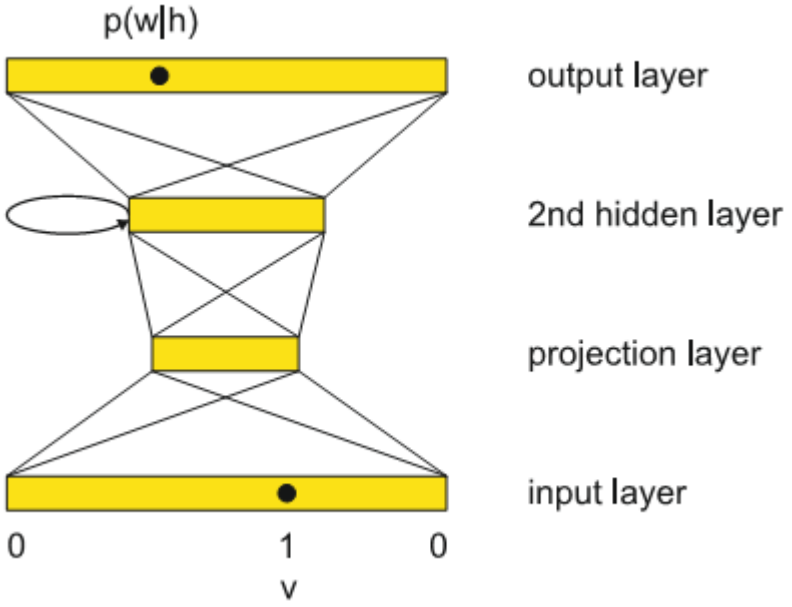
Input gate $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$

三个控制开关

LSTM应用于语言建模

LSTM NNLM实验：将LSTM cell 嵌入具有记忆功能的隐层中

Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling. INTERSPEECH. 2012.



LSTM NNLM

实验数据，KN 4-gram语言模型的训练数据100倍于此，隐层大小为300

LM	train	dev1	dev2	test
Treebank	930 K	74 K	–	82 K
Quaero French	27 M	46 K	36 K	35 K

识别任务：在基线系统的解码结果上，作n-Best rescore，n=1000

LM	dev2	test
KN 4-gram	19.7 %	17.6 %
KN 4-gram + LSTM	19.2 %	17.3 %

Table 2: Word error rate results for Quaero French.

结论：LSTM在LM建模中的应用，暂没有如Elman RNN网络被广泛研究

3. 神经网络语言模型在语音识别中的应用进展

RNNLM研究及其在语音识别中的应用

- RNN研究现状

RNN近两年研究论文数量

Year	Place	# for LM	# for Denoise / Robust ASR	# for other domain	# total
2012	ICASSP	**1**	**3**	**3**	**7**
	Interspeech	**3**	**4**	**1**	**8**
2013	ICASSP	**4**	**4**	**9**	**17**
	Interspeech	**4**	**2**	**6**	**12**
	ASRU	**3**	**1**	**1**	**5**

- RNN-LM研究重点

  - 性能层面：添加辅助的输入层特征，输出层聚类策略，多LM融合
  - 速度层面：加速训练
  - 应用于ASR：嵌入解码器

3.1 RNNLM研究——性能层面

工作一：在RNN-LM结构的输入层中，添加额外特征，拼接成新的输入再进行网络训练

Delft INTERSPEECH. 2012. Shi Y, Wiggers P, Jonker C M. Towards Recurrent Neural Networks Language Models with Linguistic and Contextual Features.

$$x(t) = [w(t)^{\mathrm{T}}, f_1(t)^{\mathrm{T}}, \ldots, f_P(t)^{\mathrm{T}}, h(t-1)^{\mathrm{T}}]^{\mathrm{T}}$$

- 三类可选附加特征（**离线提取**）

1. POS tags and lemmas
2. Topic
3. Socia-situational setting

- 实验数据：Corpus Spoken Dutch

- **论文结论**

添加POS tags and lemmas和Socia-situational setting时，性能提升最大，PPL下降27.18%。

Model	Perplexity
SRILM+KN5	228.82
SRILM+class+KN5	227.88
RNNLM	114.79
RNNLM+POS	90.56
RNNLM+lemma	102.26
RNNLM+SS	103.99
RNNLM+T10	102.45
RNNLM+T20	101.07
RNNLM+T40	104.48
RNNLM+T100	106.28
RNNLM+POS +SS	93.63
RNNLM+POS +T20	86.43
RNNLM+SS+T20	94.20
RNNLM+SS+lemma	93.35
RNNLM+POS +lemma(no word)	230.75
RNNLM+POS +lemma	90.49
RNNLM+POS +SS+T20	87.41
RNNLM+POS +SS+lemma	83.59
RNNLM+complete(300H)	85.88
RNNLM+complete(500H)	84.81

3.1 RNNLM研究——性能层面

工作二：改进输出分类层提升PPL性能

AT&T ICASSP2013. Caseiro D, Ljolje A. Multiple parallel hidden layers and other improvements to recurrent neural network language modeling.

- 传统做法为按词频将词典中词分类，本文使用Brown Clustering

不同分类策略对PPL的性能影响对比

h	classes	speed	Avg PPL	Min PPL
50	200 Freq	26938	156.34	155.19
100	200 Freq	12740	142.93	142.11
200	200 Freq	5178	136.12	135.32
300	200 Freq	2866	135.14	133.99
50	200 Brown	25599	145.33	142.62
100	200 Brown	12087	132.60	131.58
200	200 Brown	4966	127.09	126.23
300	200 Brown	2785	125.06	124.21
50	none	720	145.37	142.79
100	none	355	132.95	131.37
200	none	186	126.55	125.51
300	none	131	125.12	123.46

固定类别数量

h	classes	speed	Avg PPL	Min PPL
200	50 Freq	4904	137.35	135.58
200	100 Freq	5620	136.54	135.60
200	200 Freq	5178	136.12	135.32
200	400 Freq	3822	135.96	134.65
200	50 Brown	3402	129.21	128.26
200	100 Brown	4846	128.10	126.69
200	200 Brown	4966	127.09	126.23
200	400 Brown	3786	126.31	125.32
200	none	186	126.55	125.51

固定隐层大小

3.1 RNNLM研究——性能层面

工作三：多个RNN-LM融合(AT&T ICASSP2013 cont.**)**

多个隐层并行传播，同时进行参数更新（ Multiple parallel hidden layers，MPHL ）

$$P_{12}(w|h) = \frac{P_1(w|h)^{\lambda} P_2(w|h)^{1-\lambda}}{\sum_v P_1(v|h)^{\lambda} P_2(v|h)^{1-\lambda}}$$

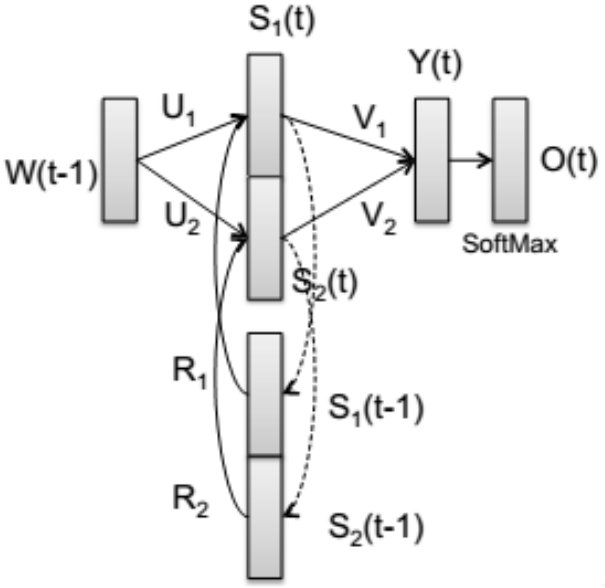
实验（on AT&T Voice Mail Task）数据（两类）

UM = corpus from internal AT&T system (2.4M tokens)

MW= poorly transcribed voice mails

结果

System	Word ACC
Baseline 4-gram	78.06
Oracle 100 best	80.40
8x UM + 6x MW RNNLM linear	79.06
8x UM + 6x MW MPHL	79.07
Oracle 1000 best	81.28
8x UM + 6x MW RNNLM linear	79.23
8x UM + 6x MW MPHL	79.36



多RNNLM融合示意图

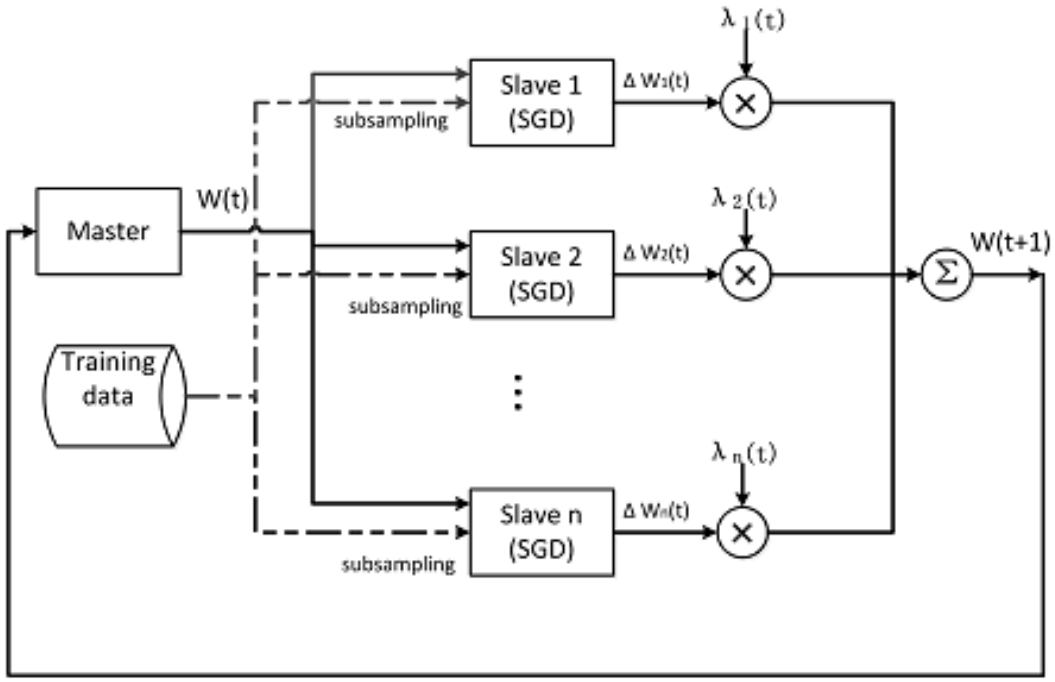
实验结果并不可观，MPHL比线性插值好一点

3.2 RNNLM研究——速度层面

Microsoft Main Work On Speed Up RNNLM Constructing

- ✓ Speed Up of Recurrent Neural Network Language Models With Sentence Independent Subsampling Stochastic Gradient Descent. Interspeech2013
- ✓ ACCELERATING RECURRENT NEURAL NETWORK TRAINING VIA TWO STAGE CLASSES AND PARALLELIZATION. ASRU2013

➢ **工作一：基于句间独立假设的并行训练**



并行策略分类：并行训练的每个处理器完成什么工作？

➢ Batch SGD： 每个处理器独立进行参数更新，主机将所有处理结果融合。

➢ **Parallel SGD**：每个处理器独立**计算更新量**，待主机将其汇总后再据此更新参数。

➢ 参数更新（n为并行数量）

$$W(t+1) = W(t) + \frac{\alpha}{n} * \sum_{k=1}^{n} \lambda_k(t) \Delta W_k(t)$$

3.2 RNNLM研究——速度层面
工作一：基于句间独立假设的并行训练

■ **并行中的参数更新**

$$W(t+1)=W(t)+\frac{\alpha}{n}*\sum_{k=1}^{n}\lambda_k(t)\Delta W_k(t)$$

α为学习速率，n为并行数量，$\lambda_k(t)$为贡献系数，$\Delta W_k(t)$为每个处理器的更新量

贡献系数确定：根据每个处理器在各自训练数据上的熵值，$E_k(t)$为第k个处理器的熵

$$\lambda_k(t)=\frac{\ln(E_k(t))}{\sum_{l=1}^{n}\ln(E_l(t))}$$

较高的熵值 ⇨ 较好的性能

■ **数据采样分配：句间独立假设**

- RNN-LM使用的BPTT算法不能有效学习更长时间的上文，如句间的影响
- 部分实际任务不需要考虑句间影响，如Voice Search

■ **采样原则**

- 保证处理器之间的数据有交叠，避免因数据切分导致各块并行结果差别大，同时起到正则化的效果
- 每次参数更新后，需要进行重新采样。

3.2 RNNLM研究——速度层面
工作一：基于句间独立假设的并行训练

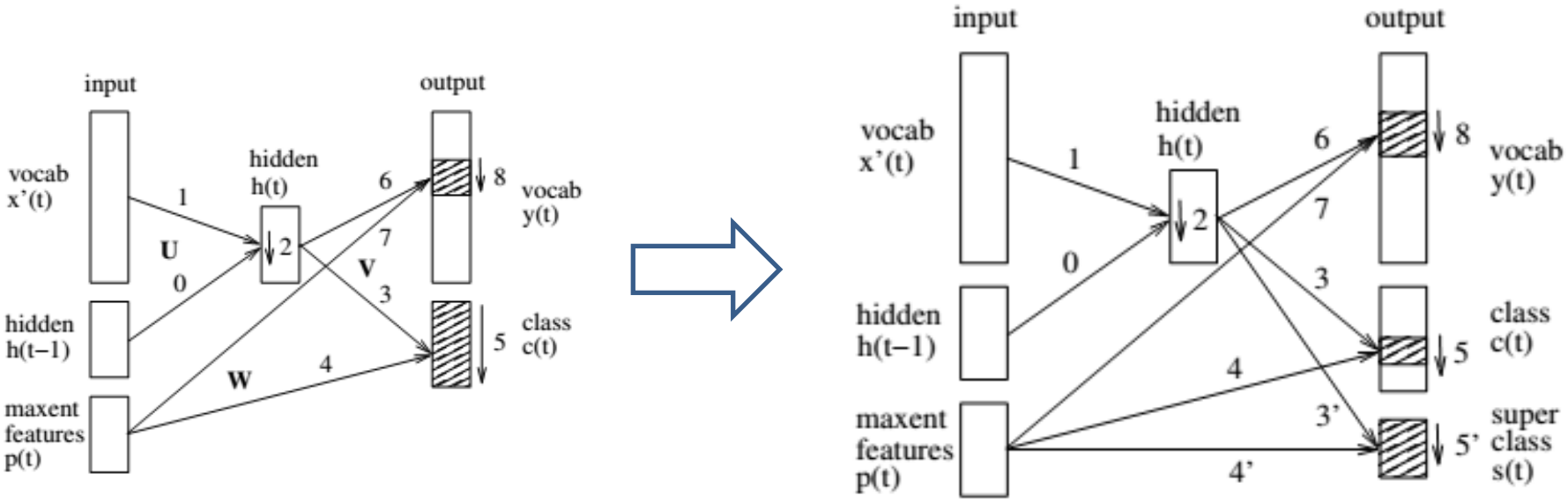
实验结果（WSJ Task）

model	size (%)	time (hours)	PPL TEST	WER DEV(%)	WER EVAL(%)
KN5-base	100	-	174.5	12.09	17.30
RNNLM-base	100	77.5	140.6	11.15	15.69
8 threads	100	78.4	140.5	11.00	16.10
8 threads	50	65.9	138.5	10.77	15.84
8 threads	40	46.1	140.4	11.01	15.71
8 threads	25	33.6	144.4	11.07	15.99
16 threads	12.5	26.1	151.0	11.14	15.66

size为训练数据量占总体训练集的比重

3.2 RNNLM研究——速度层面
工作二：Two Stage Class RNN Model

工作二：Two Stage Class RNN Model



- Super-Class选择：将class当成word处理，根据其规模分类
- 计算复杂度变化（H为隐层规模， T_b为BPTT时间，V为词典大小，C和S为分类规模）

org $H \times H \times T_b + H \times (C + \frac{V}{C})$

$C = \sqrt{V}$

new $H \times H \times T_b + H \times (S + \frac{C}{S} + \frac{V}{C})$

$C = \sqrt[3]{V^2}, S = \sqrt[3]{V}$

3.2 RNNLM研究——速度层面
工作二：Two Stage Class RNN Model

实验结果（微软短信听写任务集）

hidden size	model	setting	speed (w/s)	train time (days)	valid		test	
					PPLX	WER	PPLX	WER
N/A	baseline	N/A	N/A	N/A	79.99	26.02	83.37	24.51
25	RNN	N/A	12K	0.7	**75.63**	25.87	**78.64**	24.35
	two stage class RNN	40x1200	29K	0.3	76.54	26.02	79.49	24.38
	two stage class RNN	40x1600	28K	0.3	75.76	25.97	78.94	24.42
	parallel RNN	25x0.1	12K	0.1	76.24	**25.83**	79.25	**24.23**
	parallel RNN	50x0.05	12K	0.1	76.52	25.84	79.51	24.40
50	RNN	N/A	6.1K	1.4	**72.39**	25.50	**75.32**	24.37
	two stage class RNN	40x1200	15K	0.6	73.08	25.69	75.86	**23.95**
	two stage class RNN	40x1600	12K	0.7	72.83	25.79	75.62	24.05
	parallel RNN	25x0.1	6.1K	0.2	73.06	**25.49**	75.99	24.30
	parallel RNN	50x0.05	6.1K	0.2	72.80	25.58	75.73	24.25
100	RNN	N/A	2.7K	3.1	**70.47**	**25.34**	**73.21**	23.76
	two stage class RNN	40x1200	5.0K	1.7	71.31	25.50	73.98	**23.59**
	two stage class RNN	40x1600	5.1K	1.7	70.82	25.65	73.55	23.81
	parallel RNN	25x0.1	2.7K	0.6	71.16	25.45	73.85	23.78
	parallel RNN	50x0.05	2.7K	0.4	71.28	25.38	73.95	23.83
200	RNN	N/A	0.9K	9.4	**68.74**	**25.16**	**71.26**	23.70
	two stage class RNN	40x1200	1.5K	5.6	69.58	25.32	72.37	23.78
	two stage class RNN	40x1600	1.6K	5.3	69.24	25.42	71.89	**23.64**
	parallel RNN	25x0.1	0.9K	1.7	70.50	25.29	73.18	23.88
	parallel RNN	50x0.05	0.9K	1.2	69.95	25.25	72.68	23.76

小结微软的提速工作：

1. **两种做法虽然都带来一定的速度提升，但对性能的保持不够稳定；**
2. **并行数量远没有达到GPU的水平**

3.3 RNNLM研究——嵌入语音解码
RNNLM to WFST

工作：将RNNLM转化为WFST结构，再用于解码

Idiap Research Institute, Martigny, Switzerland. INTERSPEECH. 2012.

Lecorv é G, Motlicek P. Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition.

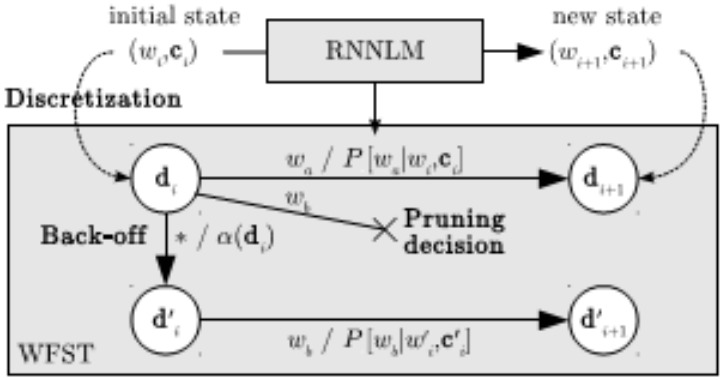


Figure 1: *Overview of the RNNLM discretization scheme.*

Data: L, a list of discrete states, i.e., of WFST nodes

$L \leftarrow f(\text{beginning of sentence})$;
while $L \neq \emptyset$ **do**
  $\mathbf{d}_{\text{src}} \leftarrow \text{pop}(L)$;
  $(w_{\text{src}}, \mathbf{c}_{\text{src}}) \leftarrow f^{-1}(\mathbf{d}_{\text{src}})$;
  $\mathbf{c}_{\text{dst}} \leftarrow \text{hidden\_layer}(w_{\text{src}}, \mathbf{c}_{\text{src}})$;
  foreach $v \in V$ **do**
    if $\mathbf{d}_{src} = \mathbf{d}_{min}$
    or not $\pi(v, \mathbf{d}_{\text{src}})$ **then**
      $p \leftarrow P(v|w_{\text{src}}, \mathbf{c}_{\text{src}})$;
      $\mathbf{d}_{\text{dst}} \leftarrow f(v, \mathbf{c}_{\text{dst}})$;
    else
      $p \leftarrow 0$;
      $v \leftarrow \epsilon$;
      $\mathbf{d}_{\text{dst}} \leftarrow \beta(\mathbf{d}_{\text{src}})$;
    if node $\mathbf{d}_{\text{dst}}$ does not exist **then**
      add_node_to_wfst($\mathbf{d}_{\text{dst}}$);
      push($L, \mathbf{d}_{\text{dst}}$);
    add_edge_to_wfst($\mathbf{d}_{\text{src}} \xrightarrow{v:v,p} \mathbf{d}_{\text{dst}}$);
compute_backoff_weights();

Algorithm 1: *Pseudo-code of the RNNLM conversion.*

3.3 RNNLM研究——嵌入语音解码
RNNLM to WFST

实验结果（NIST RT2007测试集）

Table 3: *WERs on the evaluation set of RT 2007 using n-gram LM or RNNLM-derived WFST to generate N-best lists and using n-gram LM or RNNLM to rescore them.*

Rescoring \ Decoding	2-gram LM	WFST derived from RNNLM
No rescoring	47.8%	47.3%
4-gram LM	45.2%	45.0%
RNNLM	42.9%	43.2%

实验结论

在转化后的WFST解码结果上做Rescore，性能提升没有在基线系统上的好。

4. 总结

神经网络语言（NNLM）建模分析

- 循环型NNLM**利用隐层保留历史信息**，其记录的历史并没有被深入研究或使用，抽取历史信息或基于历史信息的相关工作将是个有意义的工作。
  - LSTM的开关结构是一种过渡型的隐层节点
  - 相信将会有更为高效的隐层结构设计出来
- 前向型NNLM基于N-Gram假设，但并没有使用相应的平滑技术，使用其对概率分布的拟合能力将受限于过于复杂的计算量。

基于NNLM的工作

- 针对特定领域，设计合理的优化目标，是当前的NNLM的重点工作
- 结合LSTM在TIMIT音素识别上的工作，思考LSTM在声学和语言学的融合
- 可以多从NLP，IR等领域借鉴思路

Thank You！