Recurrent neural network based language model

Tomáš Mikolov

Brno University of Technology, Johns Hopkins University

20. 7. 2010

Overview

- Introduction
- Model description
- ASR Results
- Extensions
- MT Results
- Comparison and model combination
- Main outcomes
- Future work

Introduction

- Neural network based LMs outperform standard backoff n-gram models
  - Words are projected into low dimensional space, similar words are automatically clustered together.
  - Smoothing is solved implicitly.
  - Backpropagation is used for training.

Introduction

- Recurrent vs feedforward neural networks
  - In feedforward networks, history is represented by context of $N-1$ words - it is limited in the same way as in N-gram backoff models.
  - In recurrent networks, history is represented by neurons with recurrent connections - history length is unlimited.
  - Also, recurrent networks can learn to compress whole history in low dimensional space, while feedforward networks compress (project) just single word.
  - Recurrent networks have possibility to form short term memory, so they can better deal with position invariance; feedforward networks cannot do that.

Model description - feedforward NN

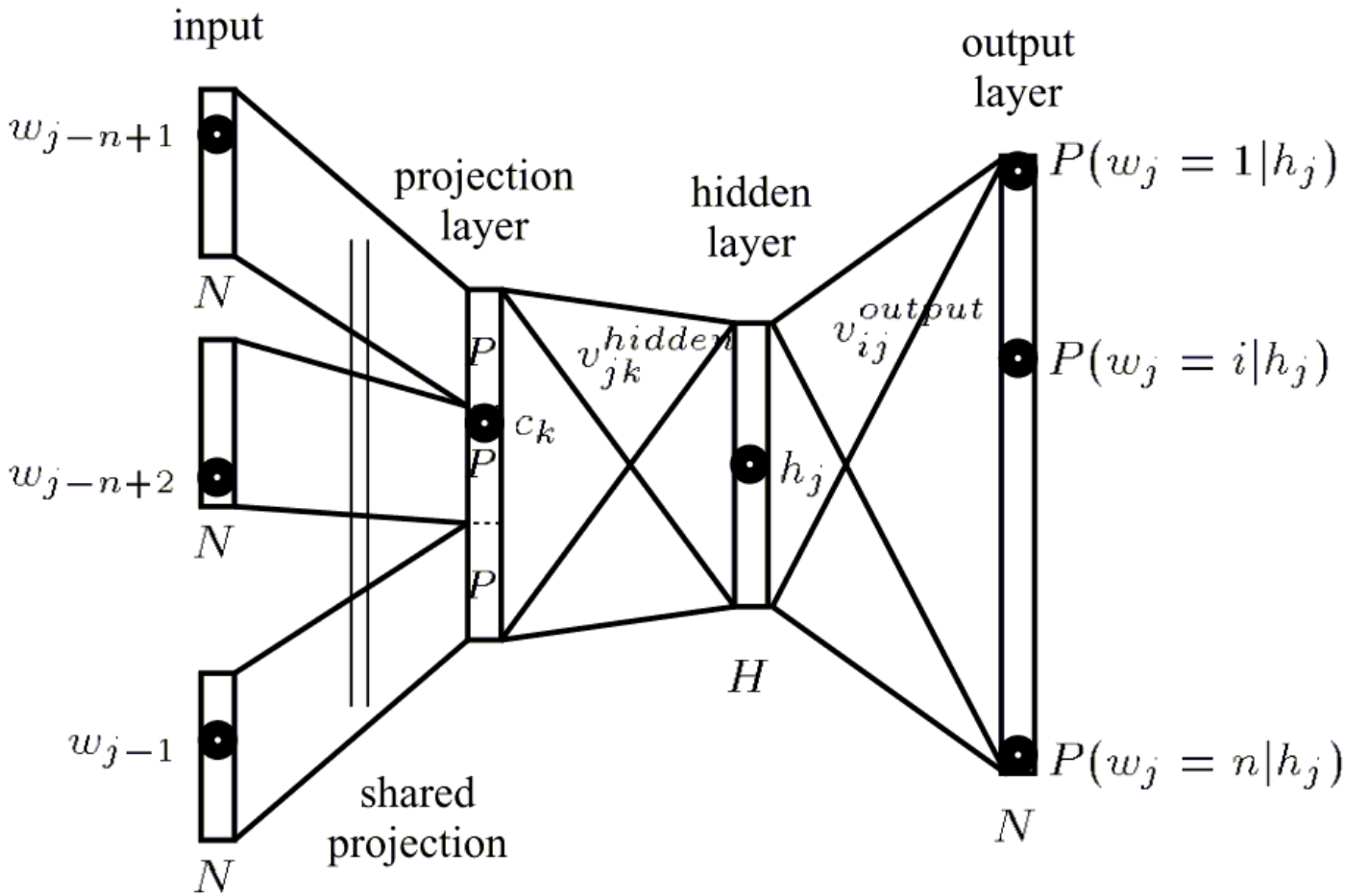

Figure: *Feedforward neural network based LM used by Y. Bengio and H. Schwenk*

Model description - recurrent NN

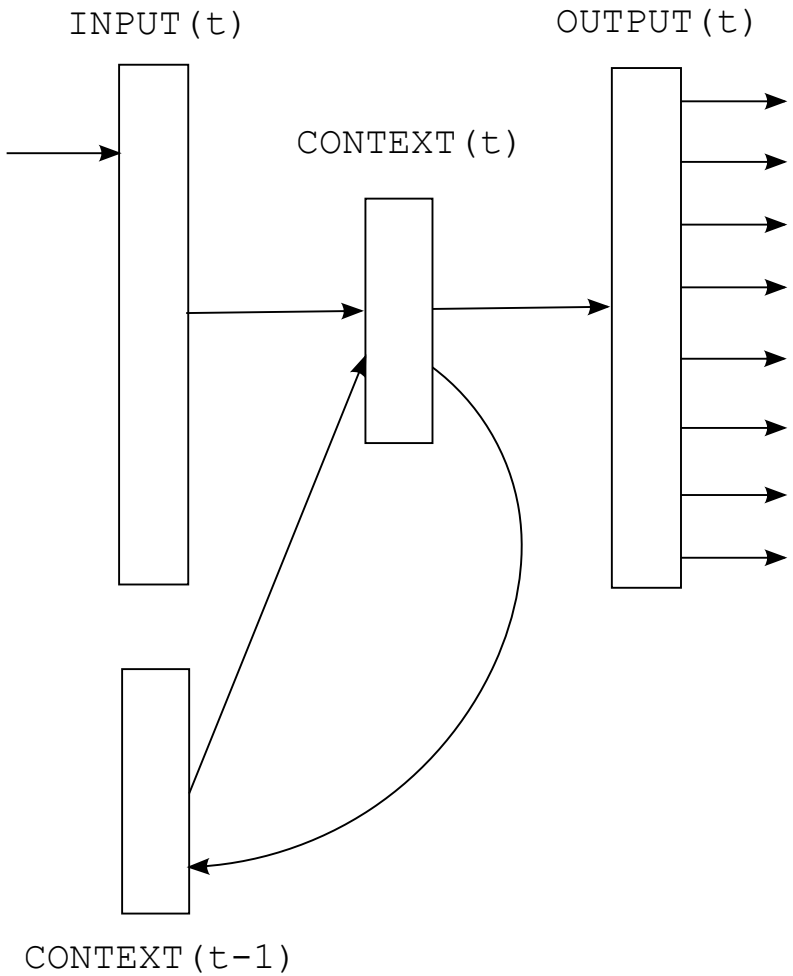

Figure: *Recurrent neural network based LM*

Model description

- The recurrent network has an input layer x, hidden layer s (also called context layer or state) and output layer y.
- Input vector $x(t)$ is formed by concatenating vector w representing current word, and output from neurons in context layer s at time $t-1$.
- To improve performance, infrequent words are usually merged into one token.

Model description - equations

$$x(t) = w(t) + s(t-1) \tag{1}$$

$$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right) \tag{2}$$

$$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right) \tag{3}$$

where $f(z)$ is sigmoid activation function:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

and $g(z)$ is softmax function:

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \tag{5}$$

Comparison of models

Model	PPL
KN 5gram	93.7
feedforward NN	85.1
recurrent NN	80.0
4xRNN + KN5	73.5

- Simple experiment: 4M words from Switchboard corpus
- Feedforward networks used here are slightly different than what Bengio & Schwenk use

Results - Wall Street Journal

	PPL		WER	
Model	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1

- RNN configuration is written as *hidden/threshold* - 90/10 means that network has 90 neurons in hidden layer and threshold for keeping words in vocabulary is 10.
- All models here are trained on 6.4M words.
- The largest networks perform the best.

Results - Wall Street Journal

Model	DEV WER	EVAL WER
Baseline - KN5	12.2	17.2
Discriminative LM	11.5	16.9
Joint LM	-	16.7
Static 3xRNN + KN5	11.0	15.5
Dynamic 3xRNN + KN5	10.7	16.3

- Discriminative LM is described in paper *Puyang Xu and Damianos Karakos and Sanjeev Khudanpur. Self-Supervised Discriminative Training of Statistical Language Models. ASRU 2009.* Models are trained on 37M words.
- Joint LM is described in paper *Denis Filimonov and Mary Harper. 2009. A joint language model with fine-grain syntactic tags. In EMNLP.* Models are trained on 70M words.
- RNNs are trained on 6.4M words and are interpolated with backoff model trained on 37M words.

Results - RT05

Model	WER static	WER dynamic
RT05 LM	24.5	-
RT09 LM - baseline	24.1	-
3xRNN + RT09 LM	23.3	22.8

- RNNs are trained only on in-domain data (5.4M words).
- Backoff models are trained on more than 1300M words.

Extensions - Dynamic models

- Language models are usually *static*. Testing data do not change models directly.
- By *dynamic* language model we denote model that updates its parameters as it processes the testing data.
- In WSJ results, we can see improvement on DEV set and degradation on EVAL set. Current explanation is that testing data need to keep natural order of sentences, which is true only for DEV data.

Character based LMs - Results

Model	Log Probability
5gram	-175 000
9gram	-153 000
basic RNN 640	-170 000
BPTT RNN 640	-150 000

- Simple recurrent neural network can learn longer context information. However, it is difficult to go beyond 5-6 grams.
- Backpropagation through time algorithm works better: resulting network is better than the best backoff model.
- Computational cost is very high as hidden layers need to be huge and network is evaluated for every character.

Results - IWSLT 2007 Chinese → English

Model	BLEU
Baseline	0.493
+4xRNN	0.510

- Machine translation from Chinese to English.
- RNNs are used to provide additional score when rescoring N-best lists.
- 400K words in training data both for baseline and for RNN models. Small vocabulary task.

Results - NIST MT05 Chinese → English

Model	BLEU	NIST
Baseline	0.330	9.03
RNN 3M	0.338	9.08
RNN 17M	0.343	9.15
RNN 17M full + c80	0.347	9.19

- NIST MT05: translation of newspaper-style text. Large vocabulary.
- RNN LMs are trained on up to 17M words, baseline backoff models on much more.
- RNN c80 denotes neural network using compression layer between hidden and output layers.

Extensions - compression layer

Model	BLEU
RNN 17M 250/5 full	0.343
RNN 17M 500/5 c10	0.337
RNN 17M 500/5 c20	0.341
RNN 17M 500/5 c40	0.341
RNN 17M 500/5 c80	0.343

- Hidden layer keeps information about the whole history, some of that might not be needed to compute probability distribution of the next word.
- By adding small *compression* layer between hidden and output layers, amount of parameters can be reduced very significantly (more than 10x).
- Networks can be trained in days instead of weeks (with a small loss of accuracy).

Comparison and model combination - UPenn

- UPenn Treebank portion of the WSJ corpus.
- 930K words in training set, 74K in dev set and 82K in test set
- Open vocabulary task, vocabulary is given and is limited to 10K words.
- Standard corpus used by many researchers to report PPL results.

Backpropagation through time - UPenn corpus

Steps	1	2	3	4	5	6	7	8
PPL	145.9	140.7	141.2	135.1	135.0	135.0	134.7	135.1

- Table shows perplexities for different amount of steps for which error is propagated back in time (1 step corresponds to basic training).
- BPTT extends training of RNNs by propagating error through recurrent connections in time.
- Results are shown on dev set of UPenn corpus (930K words in training set)
- Results are averages from 4 models to avoid noise.
- BPTT provides 7.5% improvement in PPL over basic training for this set.
- With more data, the difference should be getting bigger.

Comparison and model combination - UPenn

Model	PPL	Entropy reduction
GT3	165.2	-2.2%
KN5	147.8	0%
KN5+cache	133.1	2.1%
Structured LM (Chelba)	148.9	-0.1%
Structured LM (Roark)	137.2	1.5%
Structured LM (Filimonov)	127.2	3%
Random Forest (Peng Xu)	131.9	2.3%
PAQ8o10t	131.1	2.3%
Syntactic NN (Emami, baseline KN4 141)	107	5.5%
8xRNN static	105.4	6.8%
8xRNN dynamic	104.5	6.9%
static+dynamic	97.4	8.3%
+KN5	93.9	9.1%
+KN5(cache)	90.4	9.8%
+Random forest (Peng Xu)	87.9	10.4%
+Structured LM (Filimonov)	87.7	10.4%

UPenn: data sampling: KN5 n-gram model

coke common closed at $ N a share including modest high backed by with its proposed
for denied by equivalent to ibm the they build a <unk> used in october N republics
australia 's domestic affairs and <unk> but by private practice of the government to the technology traders say
rural business buoyed by improved <unk> so <unk> that <unk> up <unk> progress
spending went into nielsen visited were issued soaring searching for an equity giving
valued at $ N to $ N
but a modest what to do it
the effort into its <unk>
spent by <unk> in
a chance affecting price after-tax legislator board closed down N cents
sir could be sold primarily because of low over the <unk> for the study illustrates the
company one-third to executives note cross that will sell by mr. investments
which new however said
he <unk> up
mr. rosen contends that vaccine deficit nearby in benefit plans to take and william gray
but his capital-gains provision
a big engaging in other and new preferred stock was n't chevrolet bidders answered
what i as big were improvements in a until last the on the economy <unk> appearance
engineered and porter an australian dollars halted to boost sagging <unk> which
previously announced accepted a cheaper personal industries the downward its N
support the same period
the state department say is $ N

UPenn: data sampling: RNN mixture

meanwhile american brands issued a new restructuring mix to <unk> from continuing operations in the west
peter <unk> chief executive officer says the family ariz. is left get to be working with the dollar
it grew the somewhat <unk> that did n't blame any overcapacity
if the original also apparently might be able to show
it was on nov. N
the stock over the most results of this is very low because he could n't develop the
senate says rep. edward bradley a bros. vowed to suit the unit 's latest finance minister
i helps you know who did n't somehow he got a course and now arrived
that there wo n't be drawn provides ima <unk> to better information management in several months
the <unk> world-wide bay area although declining stock that were planning by that reserves continues as workers at a special level of several gold slowly <unk> and <unk> mining stocks and affiliates were n't disclosed
silver are for tax-free college details and the university of hawaii
cellular claims however that went into building manufacturing huge <unk>
we need to move up with liquidity and little as much as programs that <unk> adopted forces can necessary
stock prices recovered paid toward a second discount to even above N N
the latest 10-year interbank misstated <unk> in new york arizona peak
merrill lynch capital markets

Main outcomes

- RNN LM is probably the simplest language model today. And very likely also the most intelligent.
- It has been experimentally proven that RNN LMs can be competitive with backoff LMs that are trained on much more data.
- Results show interesting improvements both for ASR and MT.
- Simple toolkit has been developed that can be used to train RNN LMs.
- This work provides clear connection between machine learning, data compression and language modeling.

Future work

- Clustering of vocabulary to speed up training
- Parallel implementation of neural network training algorithm
- Evaluation of BPTT algorithm for a lot of training data
- Go beyond BPTT?
- Comparison against the largest possible backoff models