Text Window Denoising Autoencoder: Building Deep Architecture for Chinese Word Segmentation

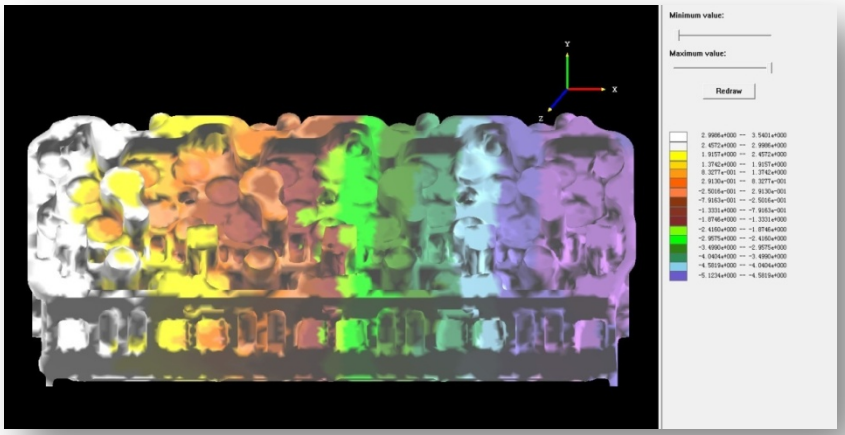
Wu Ke，Gao Zhiqiang，Peng Cheng，Wen Xiao

School of Computer Science & Engineering, Southeast University

Motivation 1

AI as Replacement for Numerical Simulation

Manufacturability analysis of car engine (General Motors Corp., USA)



Displacement analysis

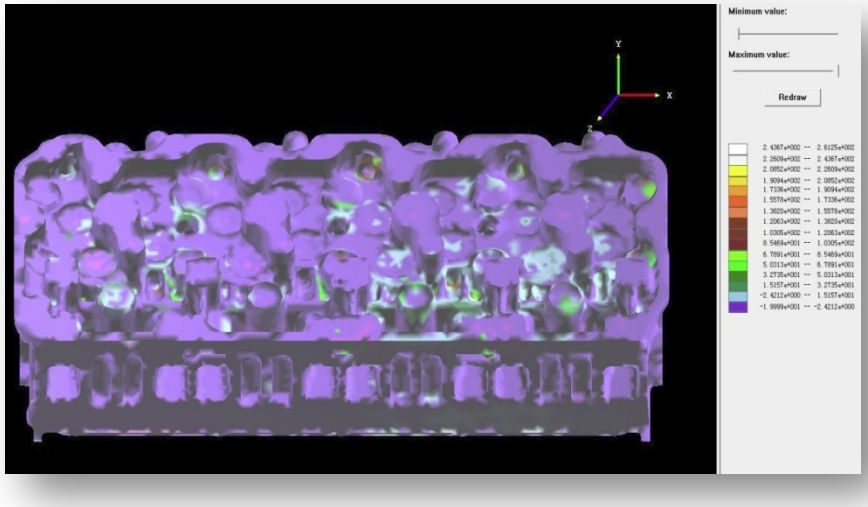
a) Finite Element Methods:
   40 servers for 1 week
b) ANN (10,000 input neurons):
   3 minutes

Average training error: 0.37%
Average test error: 0.46%

Motivation 1

AI as Replacement for Numerical Simulation

Manufacturability analysis of car engine (General Motors Corp., USA)



However, for stress analysis

Shallow ANN failed to produce better results even with very large neural network:

Average training error: 5.96%
Average test error: 16.13%

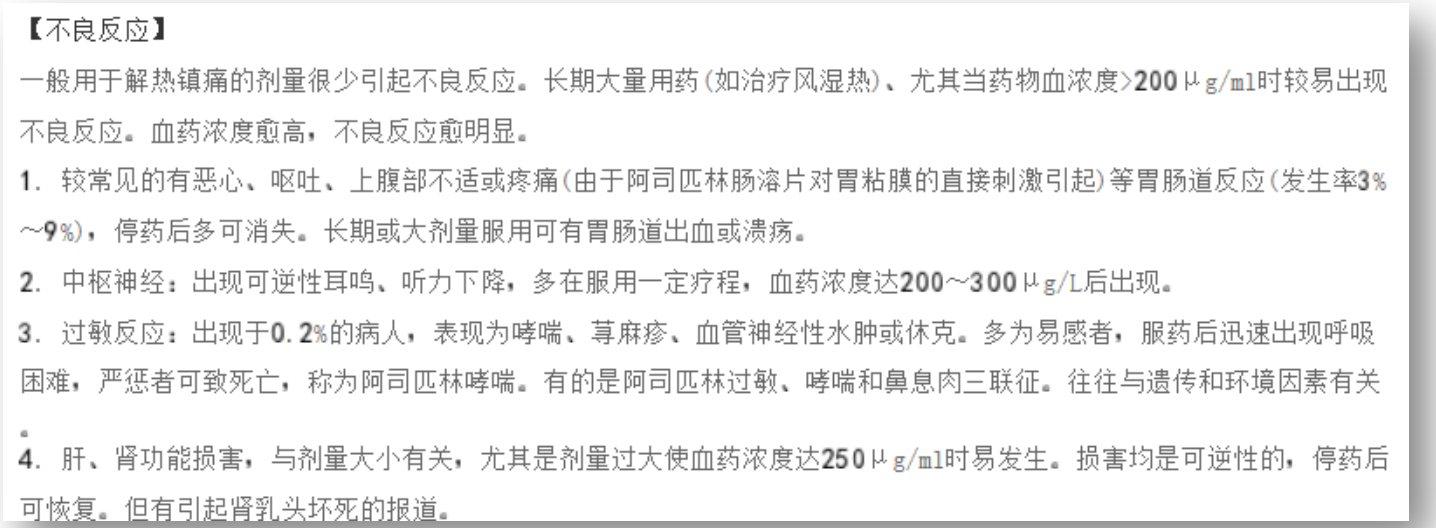
Motivation 2

NLP in Healthcare, Bioinformatics, Medical Informatics

Current NLP systems does not meet the demands of our NLP in Healthcare applications

Applying semantic analysis to Adverse Drug Reaction (ADR) from package insert, pharmacological actions, contraindications.

Automatic real-time ADR signal detection.



【不良反应】

一般用于解热镇痛的剂量很少引起不良反应。长期大量用药(如治疗风湿热)、尤其当药物血浓度>200μg/ml时较易出现不良反应。血药浓度愈高，不良反应愈明显。

1. 较常见的有恶心、呕吐、上腹部不适或疼痛(由于阿司匹林肠溶片对胃粘膜的直接刺激引起)等胃肠道反应(发生率3%～9%)，停药后多可消失。长期或大剂量服用可有胃肠道出血或溃疡。

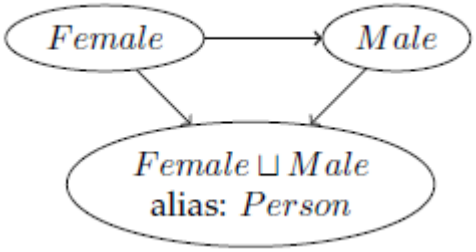
2. 中枢神经：出现可逆性耳鸣、听力下降，多在服用一定疗程，血药浓度达200～300μg/L后出现。

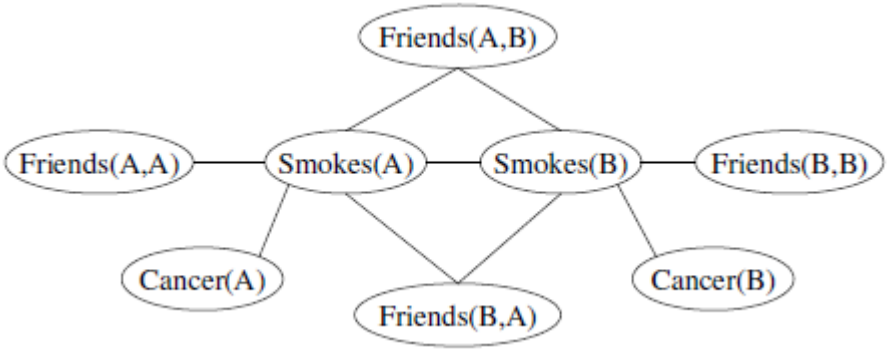
3. 过敏反应：出现于0.2%的病人，表现为哮喘、荨麻疹、血管神经性水肿或休克。多为易感者，服药后迅速出现呼吸困难，严惩者可致死亡，称为阿司匹林哮喘。有的是阿司匹林过敏、哮喘和鼻息肉三联征。往往与遗传和环境因素有关。

4. 肝、肾功能损害，与剂量大小有关，尤其是剂量过大使血药浓度达250μg/ml时易发生。损害均是可逆性的，停药后可恢复。但有引起肾乳头坏死的报道。

Motivation 3

Probabilistic Graphical Model





Entity correspondence

We are using Bayesian Description Logic (BelNet) for ontology learning (TBox learning) and Markov Logic for entity correspondence or linking.

Parameter estimation is not scalable

The reasonable number of nodes is less than 10,000 for parameter estimation.

The performance of structure learning is bad

The are too many local optima.

Contents

Building Deep Architecture for Chinese Word Segmentation

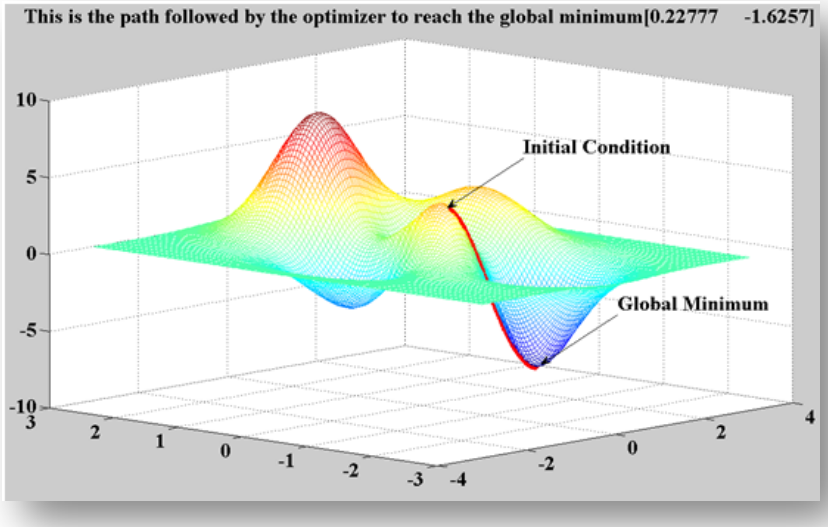
1. Introduction
2. Deep Neural Network Framework for Chinese Word Segmentation
3. Pre-train Chinese Character Embeddings
4. Pre-train Hidden Layers: (Stacked) Text Window Denoising Autoencoder
5. Experiments and Analysis
6. Future Works

1. Introduction

How to Do Deep Learning

How to effectively train a deep model (Deep Neural Network)?

Layer-wise "pre-training" before classical back propagation.



Pre-training puts the model in a near optimal position before the gradient based searching starts

Restricted Boltzmann Machines, Autoencoder, etc.

1. Introduction

How to Do Deep Learning

Denoising autoencoders have already been shown very useful at constructing deep architectures.

A denoising autoencoder consists of two parts:

a) Encoder processes noised data and produces real-valued vector as an "encode" (features) of the data.

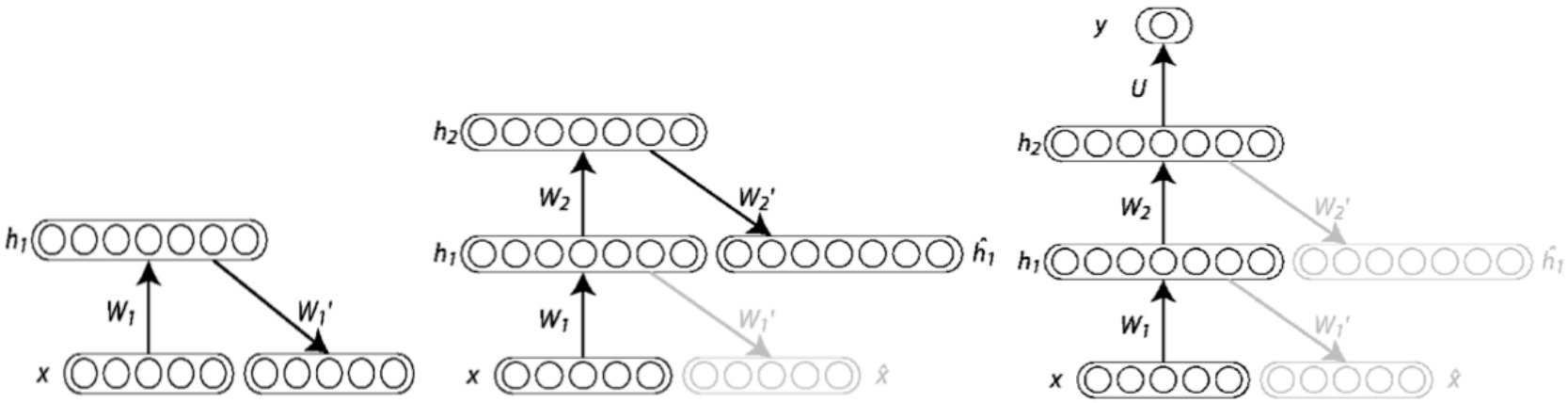
b) Decoder processes the "encode" and tries to reconstruct the clean data.

The optimization target of training denoising autoencoders is minimizing reconstruction error.

1. Introduction

How to Do Deep Learning

Deep architecture can be built by stacking Denoising Autoencoders:



1. Introduction

Related Works

Language Model

Yoshua Bengio [University de Montreal] et al. Probabilitic neural language model.

Tomas Mikolov [Google] et al. Recurrent neural network.

Parsing, Relation Classification, Sentiment Analysis, Paraphrase Detection

Richard Socher et al. [Stanford]

Chucking, Named Entity Recognition, POS Tagging

Ronan Collobert et al. [IDIAP Research Institute]

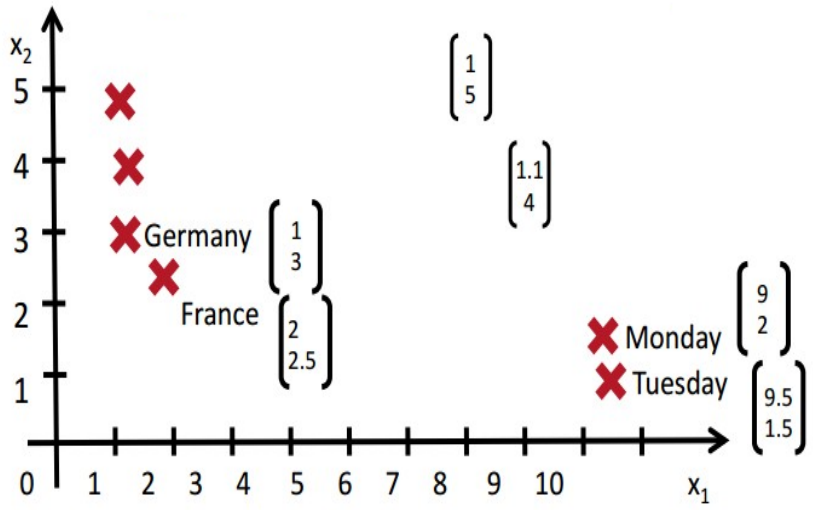
1. Introduction

Deep Learning in English NLP

Deep learning has led to many recent breakthroughs in English natural language processing [4].

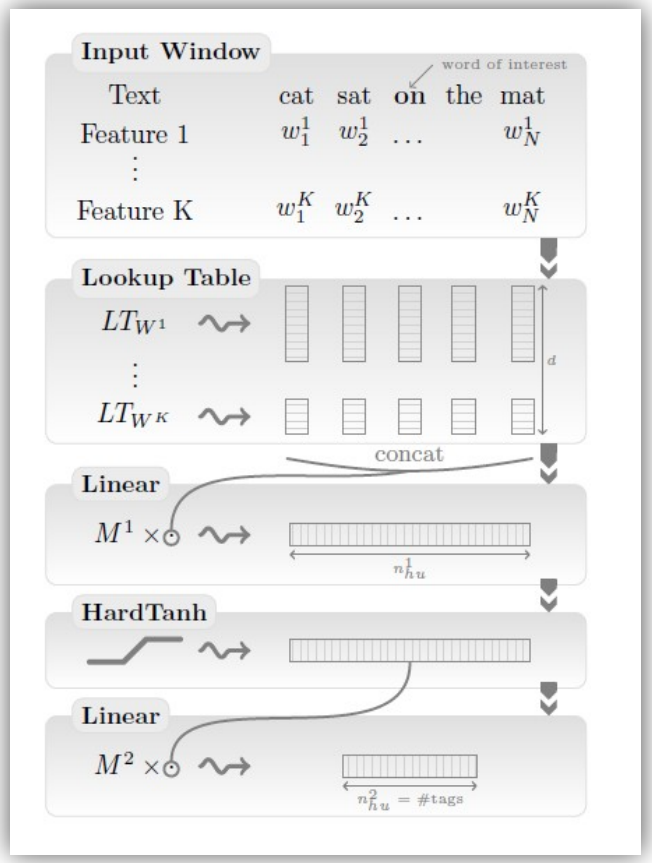
The basic idea is [1] :

a) firstly construct real-valued vectors for common English words



1. Introduction

Deep Learning in English NLP



b) then use a multi-layer neural network to process these vectors.

Only word embeddings are pre-trained. Hidden layers are not.

Good results have been achieved in English NLP tasks: Chunking, Named Entity Recognition, etc.

1. Introduction

Differences between Chinese and English

The Chinese language is composed of characters, while English of words.

a) 5,000 Chinese characters, cover 99% of Chinese Wikipedia

b) > 120,000 English words, cover 99% of English Wikipedia

Meanings in Chinese are conveyed by complex relationships between characters.

Chinese characters have meanings themselves, they can still form words that have completely different meanings from the meanings represented by the characters alone.

1. Introduction

Due to These Differences

Complete pre-training is more beneficial for deep models for Chinese NLP.

a) Current deep learning approaches for English NLP lack a pre-training solution for the hidden layers, they only pre-train the embedding layer.

b) This may be due to that vocabulary in English is so large that the embedding layer dominates in the model.

There is no explanation on why training a neural language model is a good way to pre-train the embedding layer,

or its relationship with other commonly used pre-training methods.

1. Introduction

In this paper

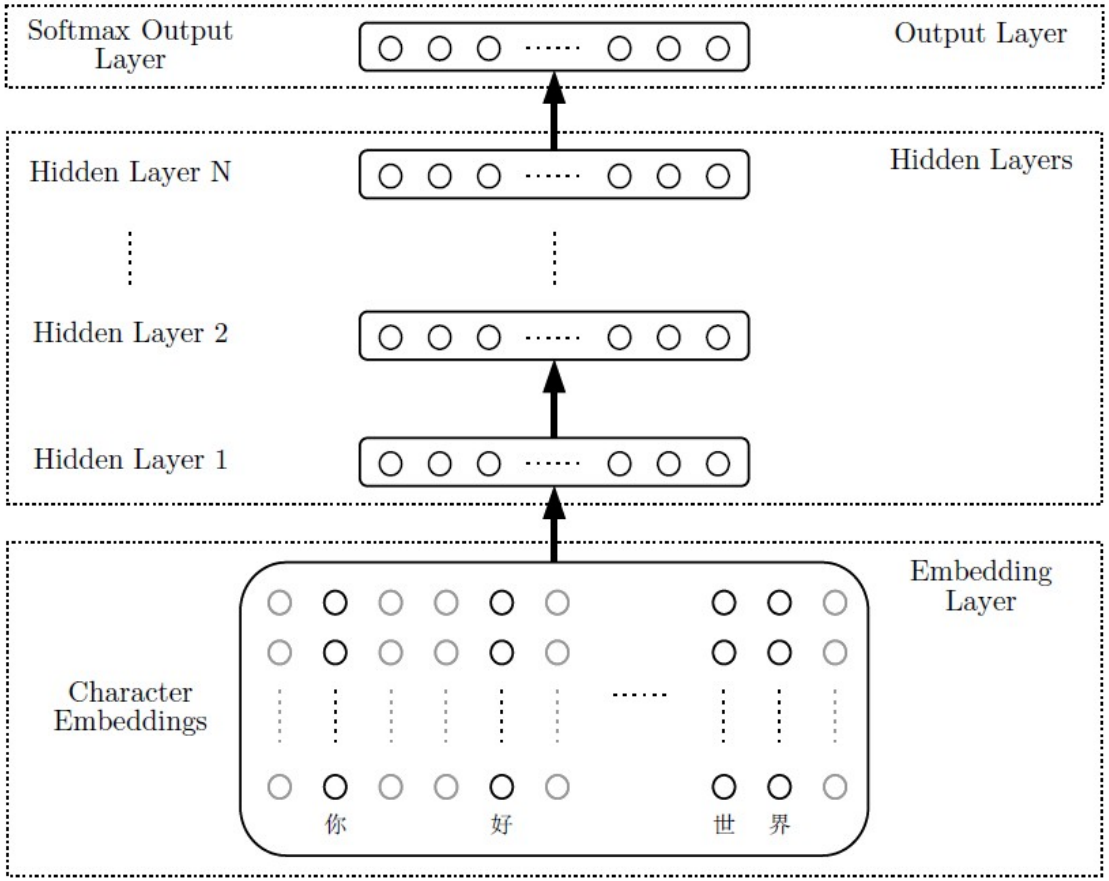
We propose a deep neural network model for sequence tagging tasks in Chinese NLP, as well as a complete pre-training solution.

a) We use a different criterion to build Chinese neural language model.

b) We explain that the training process of our neural language model is essentially the same as training a special denoising autoencoder on text window, which we call ***t****ext* ***w****indow de****n****oising* ***a****utoencoder* (TINA).

c) We describe the method to stack TINA as a way to pre-train deep neural networks for Chinese word segmentation.

2. Deep Neural Network Framework for Chinese Word Segmentation



We view Chinese word segmentation as sequence tagging, which means to assign a tag to each Chinese character ("BIU" tag schema)

3. Pre-train Chinese Character Embeddings

Pre-training by building (a slightly unconventional) Chinese neural language model: a neural network to predict the center character in a text window given its context.

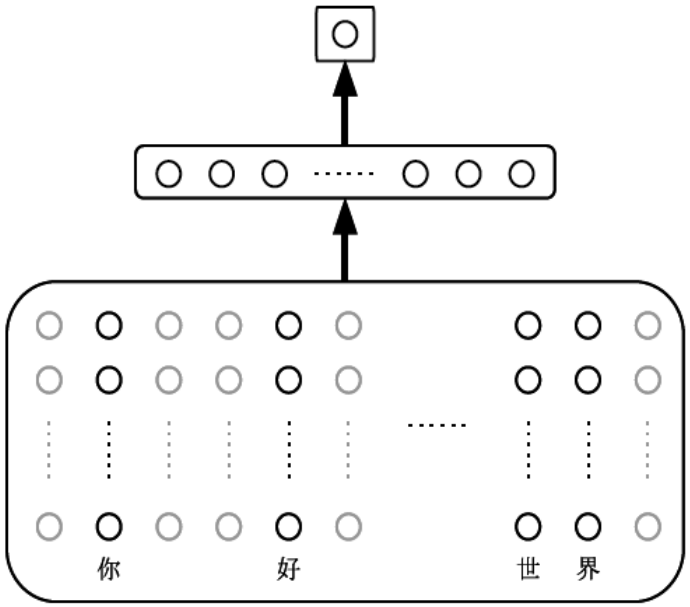
For example, given the text window:

你好_世界

the model should predict the probability distribution of the Chinese characters appear in the position of _.

3. Pre-train Chinese Character Embeddings

The model is given the context as well as a random character, and then estimate the probability that the given character is the correct one with the context.



3. Pre-train Chinese Character Embeddings

We want to build a neural network to predict the central character in a text window given its context:

Or more formally:

$$P(w|w_{-s/2}^{-1}, w_1^{s/2}) = \frac{P_c(w = w_0|w_{-s/2}^{-1}, w_1^{s/2}, w)}{\sum_{c_i \in \mathcal{D}} P_c(c_i = w_0|w_{-s/2}^{-1}, w_1^{s/2}, c_i)}$$

3. Pre-train Chinese Character Embeddings

Positive examples are text windows extracted directly from a given Chinese corpus.

Negative examples are generated by replacing the central character in a positive example with a random character.

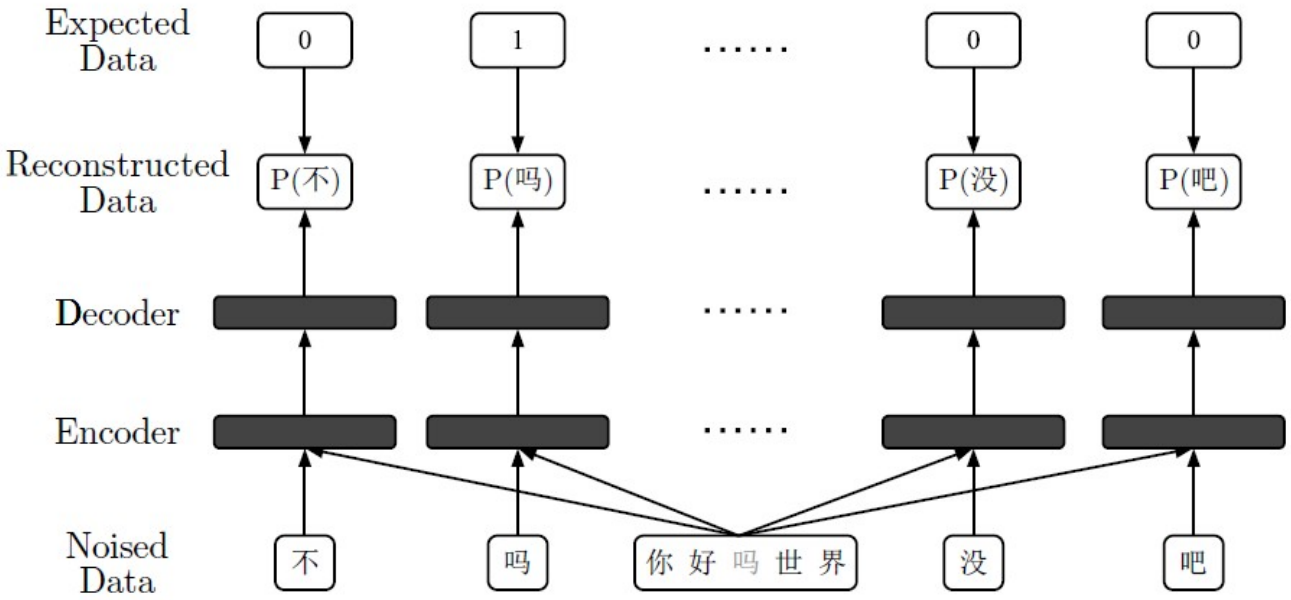
Train the model by maximizing the following log-likelihood criterion:

$$\ell(\theta : \mathcal{T}) = \sum_{\forall c_0 = w_0} \log P_c(c_0 = w_0 | \theta, w_{-s/2}^{-1}, w_1^{s/2}, c_0) - \sum_{\forall c_0' \neq w_0} \log P_c(c_0' = w_0 | \theta, w_{-s/2}^{-1}, w_1^{s/2}, c_0)$$

4. Pre-train Hidden Layer: (Stacked) Text Window Denoising Autoencoder

4.1 Neural Language Model as Text Window Denoising Autoencoder

The neural language model is essentially a special denoising autoencoder, which we call ***t**ext **w**indow de**n**oising **a**utoencoder* (TINA).



4. Pre-train Hidden Layer: (Stacked) Text Window Denoising Autoencoder

4.1 Neural Language Model as Text Window Denoising Autoencoder

Formally, given our neural language model with a single hidden layer:

$$
\begin{aligned}
L_1(\mathbf{c_1}, \ldots, \mathbf{c_s}) &= L_{hidden}(L_{input}(\mathbf{c_1}, \ldots, \mathbf{c_s})) \\
&= \tanh(\mathbf{w} \cdot (\mathbf{W} \cdot \mathbf{c_1}, \ldots, \mathbf{W} \cdot \mathbf{c_s}) + \mathbf{b}) \\
L_2(\mathbf{c_1}, \ldots, \mathbf{c_s}) &= L_{output}(L_1(\mathbf{c_1}, \ldots, \mathbf{c_s})) \\
&= \text{sigmoid}(\mathbf{w} \cdot (L_1(\mathbf{c_1}, \ldots, \mathbf{c_s}) + \mathbf{b}),
\end{aligned}
$$

4. Pre-train Hidden Layer: (Stacked) Text Window Denoising Autoencoder

4.1 Neural Language Model as Text Window Denoising Autoencoder

The encoder of ***t**ext **w**indow de**n**oising **a**utoencoder* (TINA):

$$encoder(\mathbf{x}) = (L_1(\mathbf{x}, \mathbf{c_1}), \ldots, L_1(\mathbf{x}, \mathbf{c_s})),$$

The feature of ***t**ext **w**indow de**n**oising **a**utoencoder* (TINA):

$$feature(\mathbf{x}) = (\mathbf{y_1}, \ldots, \mathbf{y_n}),$$

The decoder of ***t**ext **w**indow de**n**oising **a**utoencoder* (TINA):

$$decoder(\mathbf{y}) = (L_2(\mathbf{y_1}), \ldots, L_2(\mathbf{y_n})).$$

4. Pre-train Hidden Layer: (Stacked) Text Window Denoising Autoencoder

4.1 Neural Language Model as Text Window Denoising Autoencoder

The square reconstruction error that this ***t**ext **w**indow de**n**oising **a**utoencoder* (TINA) optimizes is:

$$E(\theta, w_{-s/2}^{-1}, w_0, w_1^{s/2}) = \sum_{\forall c_i \in \mathcal{D}} (r_i - 1_{\{c_i = w_0\}})^2,$$

Note that since:

$$r_i = L_2(\mathbf{y_i}) = P_c(c_i = w_0 | \theta, w_{-s/2}^{-1}, c_i, w_1^{s/2}),$$

Minimizing the reconstruction loss function of TINA is exactly the same as maximizing the log-likelihood criterion we proposed for the Chinese neural language model.

4. Pre-train Hidden Layer: (Stacked) Text Window Denoising Autoencoder

4.2 Building Deep Architecture

Because our neural language model can be seen as a special kind of denoising autoencoder.

We can then follow the stacking strategy of standard denoising autoencoders [12] and build a deep neural network for Chinese word segmentation.

5. Experiments and Analysis

5.1 Text Window Denoising Autoencoder as a Language Model

The performance of text window denoising autoencoder as a language model.

Language Model	Log Rank Score
3-Gram (Katz backoff)	2.54
5-Gram (Katz backoff)	2.53
NLM with Margin Loss	2.48
TINA with 1 Hidden Layer	**2.44**

(PKU dataset of Chinese word segmentation bakeoff 2005

TINA model: embedding dimension = 100, hidden layer size = 300, text window size = 11)

5. Experiments and Analysis

5.2 Stacking

The log rank score of stacked TINA models:

Number of Hidden Layers	Log Rank Score
1 Hidden Layers	2.61
2 Hidden Layers	2.52
3 Hidden Layers	**2.45**

(PKU dataset of Chinese word segmentation bakeoff 2005

TINA model: embedding dimension = 50, hidden layer size = 300, text window size = 5)

5. Experiments and Analysis

5.3 Chinese Word Segmentation

Word segmentation performance of deep neural networks pre-trained by stacking TINA models (Chinese word segmentation bakeoff 2005 dataset):

Dataset	Model	Precision	$\text{Recall}_{\text{OOV}}$	$\text{Recall}_{\text{IV}}$	F1
PKU	Baseline	83.6%	5.9%	95.6%	86.9%
	50CE(r) + 1L * 300U(r)	93.5%	75.0%	92.7%	92.6%
	50CE(p) + 1L * 300U(r)	93.7%	75.9%	93.7%	93.2%
	50CE(p) + 3L * 300U(r)	93.7%	76.0%	93.9%	93.3%
	50CE(p) + 3L * 300U(f)	93.7%	76.3%	94.6%	93.6%
	50CE(p) + 3L * 300U(TINA)	94.4%	77.9%	94.8%	94.1%
	50CE(p) + 4L * 300U(TINA)	94.6%	76.6%	95.0%	**94.3%**
MSR	Baseline	91.2%	0%	98.1%	93.3%
	50CE(r) + 1L * 300U(r)	94.5%	64.0%	95.1%	94.4%
	50CE(p) + 1L * 300U(r)	95.1%	63.6%	96.1%	95.2%
	50CE(p) + 3L * 300U(r)	95.0%	63.9%	96.0%	95.1%
	50CE(p) + 3L * 300U(f)	95.2%	64.4%	96.0%	95.2%
	50CE(p) + 3L * 300U(TINA)	95.7%	65.0%	96.4%	95.6%
	50CE(p) + 4L * 300U(TINA)	95.6%	64.9%	96.4%	**95.6%**

6. Future Works

Although our best model still under perform the state of the art Chinese word segmentation model [13] by a small margin. Our work has demonstrated that deep learning can be applied to Chinese NLP tasks, especially in sequence tagging. Also the model for Chinese NLP is different from that for English NLP.

We think our method shows great potential:

a) We've only tested a few possible model configurations. Better performance is likely to be achieved by simply using larger embedding dimensions or more hidden units.

b) There also exists tricks that can significantly boost the performance of deep neural network, for example, dropout training [6].

6. Future Works

Try other deep learning models

a) Combine prior knowledge or hand-crafted features with Deep Belief Network.

b) Probability nodes versus computational neurons.

c) Chinese syntactic parsing and semantic analysis (such as PCFG and CCG) by combining deep learning and Markov Logic.

There is still much more work to be done…

References

[1] Arisoy, E., Sainath, T. N., Kingsbury, B., Ramabhadran, B.: Deep neural network language models. In Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT, pp. 20-28. Association for Computational Linguistics (2012)

[2] Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Advances in Neural Information Processing Systems, 932-938 (2001)

[3] Bengio, Y.: Learning deep architectures for AI. Foundations and Trends ® in Machine Learning, 2(1), 1-127 (2009)

[4] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. The Journal of Machine LearningResearch, 12, 2493-2537 (2011)

[5] Emerson, T.: The second international chinese word segmentation bakeo. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, vol. 133 (2005)

[6] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)

References

[7] Jiang, W. B., Sun, M., Lv, Y. J., Yang, Y. T., Liu, Q.: Discriminative Learning with Natural Annotations: Word Segmentation as a Case Study. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)

[8] Katz, S. M.: Estimation of probabilities from sparse data for the language model component of a speech recogniser. IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3), 400{401 (1987)

[9] Low, J. K., Ng, H. T., Guo, W.: A maximum entropy approach to Chinese word segmentation. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, vol. 1612164 (2005)

[10] Salakhutdinov, R., Hinton, G. E.: Deep boltzmann machines. In Proceedings of the international conference on articial intelligence and statistics vol. 5, no. 2, pp. 448-455. Cambridge, MA: MIT Press (2009)

[11] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., Manning, C. D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 151-161. Association for Computational Linguistics (2011)

References

[12] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 11, 3371-3408 (2010)

[13] Wang, K., Zong, C., Su, K. Y.: Integrating Generative and Discriminative Character-Based Models for Chinese Word Segmentation. ACM Transactions on Asian Language Information Processing, 11(2), 7 (2012)

[14] Wang, Z. G., Zong, C. Q., Xue, N. W.: A Lattice-based Framework for Joint Chinese Word Segmentation, POS Tagging and Parsing. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)

[15] Yang, N., Liu, S. J., Li, M., Zhou, M., Yu, N. H.: Word Alignment Modeling with Context Dependent Deep Neural Network. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)

[16] Zeng, X. D., Wong, F. D., Chao, S. L., Trancoso, I.: Co-regularizing character-based and word-based models for semi-supervised Chinese word segmentation. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)

References

[17] Zhang, M., Zhang, Y., Che, W. X., Liu, T.: Chinese Parsing Exploiting Characters. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)

[18] Zhao, H., Huang, C. N., Li, M.: An improved Chinese word segmentation system with conditional random eld. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, vol. 1082117. Sydney (2006)

Thank you