Complex SOM network as a Language Model for Large Vocabulary Continuous Speech Recognition

Leszek Gajecki[1], Ryszard Tadeusiewicz[2]

[1]Wyższa Szkoła Informatyki i Zarządzania, ul Sucharskiego 2, Rzeszów
[2]Akademia Górniczo Hutnicza, ul Mickiewicza 30, Kraków
[1]lgajecki@wsiz.rzeszow.pl; [2]rtad@agh.edu.pl

Abstract

Language model module applied in typical Large Vocabulary Continuous Speech Recognition (LVCSR) helps to choose right hypothesis of recognized sequence of words. Trigram model and derivative models, which put accent on words order, can be sufficient for English language. However mostly in Slavonic languages words order is less important so we need modeling which respect such property.

We present application of complex neural network, which consist of multiple Self-Organized Maps.

Keywords: Language Modeling, Speech Recognition, Neural Networks

1. Large Vocabulary Continuous Speech Recognition System and Language Model.

To understand the function of Language Model module we need to describe architecture of typical LVCSR system (fig. 1), which can be also described as a system based on Hidden Markov Model (HMM) framework. Such systems are widely represented in literature: HTK (Young and others 2006), ESAT Speech Recognition System (Duchateau 1998, ESAT-PSI 2006), or LVCSR systems for Polish Language (Brocki, Korzinek 2007, Brocki, Korzinek 2008, Hnatkowska, Sas 2008, Szymański and others 2008).

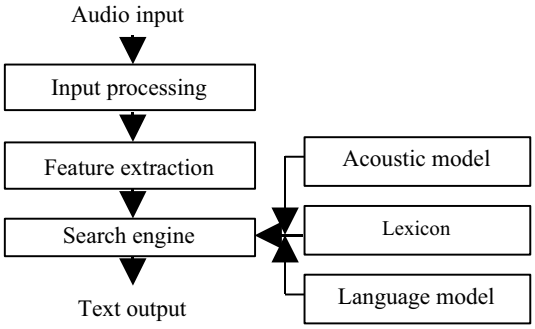

Figure 1. Architecture of LVCSR system based on HMM framework.

Acoustic waves are processed into electric signal using microphone. *Input processing* module normalize signal by amplifying, band filtering and digitizing by A/D converter. Next the spectrum of digitalized signal can be normalized. Then we have *Feature Extraction* module, which is responsible for obtaining vector of relevant acoustic properties of speech signal, which is *vector of acoustic features*: $X = [x_1, x_2, ..., x_n]$ (1.1)

Before next step we also need following modules, which are responsible for following levels of modeling (fig. 2):

a) *Acoustic model* – lowest level - represents basic speech units (like phonemes, triphones, syllables, etc.) in such way that we can compare them with vector of acoustic features.

b) *Lexicon* – describes mapping between pronunciation of each word and written form of each word. It is necessary to find which word was recognized. Pronunciation defines here the sequence of units from acoustic models that make each word. Written form is needed to create output text representing recognized word. Otherwise we would obtain only sequence of symbols of recognized phonemes.

c) *Language model* - describes how probable is given sequence of words, how much correct is this sequence according to language rules. This module should allow also simplification which we use and understand in daily speech, even thought they are linguistically incorrect.

We should also notice that lexicon and language model helps in finding right word sequence in case of much speech variation. It happens when the sequence of acoustic feature vectors doesn't match exactly to units form acoustic model, or when recognized phonemes cannot give any known word. Obtained words also can be also such, that don't occur in spoken language. However keeping all three levels of modeling together we have better chances to find right result. There are also some works concerning *Semantic model,* like publication (Erdogan 2005) which presents two layer semantic – lexical model. However we don' use it in our LVCSR system.

Having feature vector (1.1) and its past values *Search Engine* module works in two steps:

a) Building search space for second step – there are hypothesis built using knowledge obtained from Acoustic Model, Lexicon and Language Model. Hypotheses are word lattices connected with Language Model. Word lattices consist on model of each word – these are references to units from acoustic model that build each word respectively.

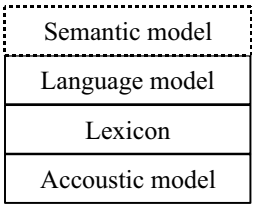

Figure 2. Levels of modeling in LVCSR systems. The highest level: Semantic model often not present in LVCSR is mentioned here for betted understanding.

b) Searching for optimal hypothesis W_{opt}, this is word sequence, which is closest to acoustic feature vector .

$$W_{opt} = \arg\max_{w_i \subset W_1^K} P(w_i \mid X) \quad (1.2)$$

where $W_1^K = w_1,...,w_K$ is vector of all possible word sequences.

Cost for each hypothesis (1.3) is computed having probability given by acoustic module $P(X \mid W_1^K)$ (it describes how similar is train of acoustic feature vector to word sequence), and using probability given by Language Model module $P(W_1^K)$:

$$f(X, W_1^K) = \log P(X \mid W_1^K) + CA \log P(W_1^K) + CC \quad (1.3)$$

where:

CA -weight for probability obtained from Language Model (applied in case of models, which can give cost after each word, like in case of trigram model)

CC -its negative value is cost for starting new word

X - acoustic feature vector

W_1^K - word sequence

Detailed information about LVCSR system can be found in (Young and others 2006, Duchateau 1998, Markovitz 1996, Benesty, Sondhi,Huang 2008), and introductory description of speech signal properties presents (Tadeusiewicz 1988).

2.Kohonen Self-Organizing Maps (SOM)

This network consist N neurons in one layer. The inputs have to be normalized that they norms are equal 1:

$$\|x\| = 1 \quad (2.1)$$

where norm: $\|x\| = \sqrt{\sum_{i=1}^{M} x_i^2}$

The outputs of neurons are computed:

$$y_j = \sum_{i=1}^{M} x_i w_{ij} \quad (2.2)$$

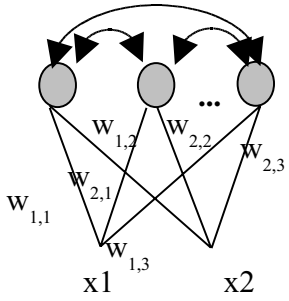

Figure 3. Architecture of Kohonen Maps (SOM)

Next the competition between neurons can be performed. The winner is neuron, which output is the highest, so in case of above assumption about norm of input, the winner's weights are template, which is closest to point represented by inputs (in M-dimensional space). Neurons can be labeled with identifiers of classes, so we get information to which class given input is closest.

SOM network is trained using competitive learning. At the beginning weights of neurons are initialized by small random numbers. After each step of learning we compute outputs of neurons and such neuron which has the highest value of output (winner) will be learned in next step – this is Winner Take All rule. Decision rule can be also soft (Winner Take Most) – then also neighbor neurons will be learned in next step – but there will be less impact, when neuron is far from winner. To establish neighborhood we put neurons on surface (it can be also space: 3-dimensional or more, but it is rare applied), and we choose the lattice (rectagonal, hexagonal,…) and we put network on nodes of this lattice. Next we decide which neurons can be neighbors. Further information about neural network can be found in (Wu Chou, Biing Hwang Juang 2003, Kohonen 2006, Duch and others 2006), while (Tadeusiewicz and others 2007) is good introductory tutorial.

3. Language modeling using neural network

Neural networks was used for speech recognition for acoustic model (Tadeusiewicz 1994, Robinson 1994), however our idea is to learn network rules present in speech for application in Language Model module. First step is rules that describe relation between words depend of their category (noun, verb, etc., but also grammatical information about number, case, person ...). As we can notice it is not perfect solution since word specific dependencies can be more important (like valence of verbs). The next step in creation of Language Model would be lexicalized grammar – grammar, which covers also rules according to specific words.

Below we present our network, which can learn relations between categories of words.

We need to code 6 properties, each one coded by "1 from N" code. Together we have string of 52 values (0/1). When network need information about category of two words at input, we concatenate their coding, this gives 104 elements.

Fleksem	Number	Gender	Case	Person	Degree
0 1 0 ...	0 1	0 1 0 0	0 0 1 ... 0	0 0 0	0 0 0

Figure 4. Coding of word's category

Basic unit of our network is Kohonen SOM network with WTM learning (Oja rule).

$$\Delta w_{ij} = y_j\left(x_i - w_{ij}\right)\eta\, \alpha\,(t) h(j, win)$$

where:

w- weights
x- input
y- output
η - learning rate

$\alpha(t) = \frac{\alpha_0 t}{C + t}$ - function depended on time of learning (t-number of learning step), to make learning stable

$h(j, win) = e^{-|j - win|}$ neighbor function (win is number of winner neuron)

The structure of network is linear, so each neuron has direct neighbors. The result should be classification – decision if given input satisfies the rules or not. To do this after learning we check how often each network wins. Having such information we decide, that neurons which wins few times, or 0 (here we chose 0) represent negative rules – data, which is classified to them represent incorrect structure of utterance. Neurons that win more times than given threshold represent positive rules and data, which have correct structure.
The output of one SOM network tell us not only which neuron won, but also how far vector of given input is from vector defined by weights of the closest neuron. In case when we want to join such network to next layer we can use outputs of neurons as information which rule is fulfilled by input and if it is positive or negative rule.
A next question is architecture of the whole network. Network has to cover all relations between words (category of them).

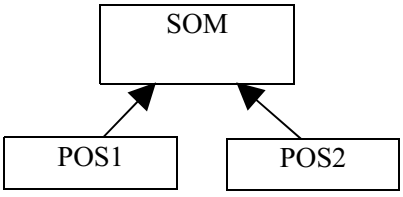

Figure 5. Simple SOM network for language modeling

First idea is to build one network (fig.5). The inputs is grammar categories (part of speech: POS) of two neighbor words. We put each two neighbor words in order to input of such network (fig.6). This simple network can represent such structure, where binary rules are applied only to neighbors, but they cannot be applied in cascade.

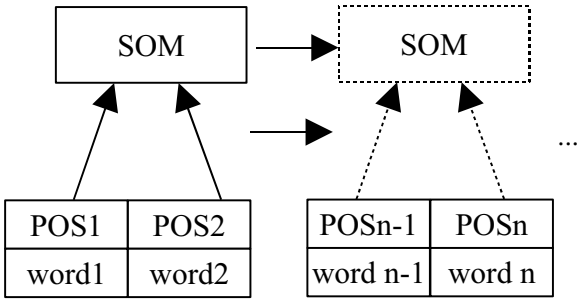

Figure 6. Application of simple SOM network.

To be able to represent rules, that can be applied according to Context-Free Grammar (CFG) we can build network that works similar to Coke-Young -Kasami (CKY) algorithm. We call such network as CKY Network.

4.CKY algorithm.

This algorithm is used for parsing according to CFG rules. Suppose, we have set of rules. In cell i,j we will write results of back application the rules, that produces symbols which we found respectively in cells: k,j and i-k-1,j+k+1 , where k=0,1,...,i. We can say also that cell i,j stores the result of parsing input elements from i to j, while top cell N-1,0 represents results of parsing whole input train.

i	j	0	1	2	3
4					
3		S,S			
2			VP		
1		VP		NP	
0		N	V	Adj	N
		John	has	white	Cat

Figure 7. CKY Algorithm

5.CKY network.

In this section we are describing the structure of CKY Network (fig. 8.)
Each CKY cell (which we can see on above figure – as areas limited by dashed lines) consist on k networks

(SOM). Further by term "network" we mean one SOM network. The whole network we call CKY Network. Each network has inputs -outputs from respective cells. Cells 0,j are direct inputs to whole network and represents categories of words. Each basic SOM network is active only when at least one SOM network is active in both cells (which are inputs to this network), and winner neurons represent positive rule on both cell. Such requirement represents application of rule to such cells, where there are results of previously applied rules. Cells 0,j are direct inputs and has no network, so we say that they are always active when we put information about category of word to such input. On the fig.8 we see that this cell substitute words by their POS (part of speech). In our model we simply generate all possible POS sequences respectively to given words. At this level possibly better solution would be application of any parser that gives on its output possible POS of given words, so number of such hypotheses can be limited to those given by parser.

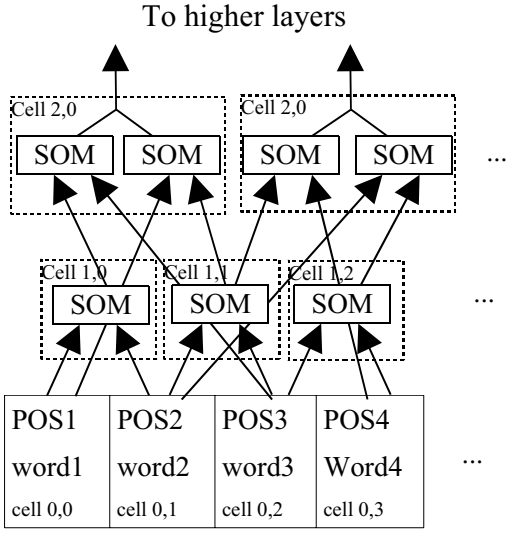

Figure 8. CKY Network

Having categories of N words on input, all cells: 0,0 , 0,1 ,..., N-1,0 are active, so we perform CKY algorithm with size N: we don't consider cells N-k,j, where j>k.

To learn such network we have to learn SOM networks firstly in cells 1,k (k=0..N-1) because they have inputs only directly from input. Next we learn networks in cells 2,k (k=0..N-2). Giving category of words on input we make that some winner neurons (one winner per one SOM network) may represent positive rules and they may activate respective SOM networks. Now we have some SOM networks in cells 2,k active so we will learn these networks in next step. Learning data for networks in cells 2,k are their inputs (outputs form respective cells). We learn only active networks. We repeat this procedure for each row in CKY table.

We decide to limit size of network to maximum 9 words, so we divide sentence to phrases (between comas). We throw sentences with longer phrases. To apply network to Language Model we need to find grammatical categories of words. Here we generate all possible sequences of categories for those words which have more than one interpretation. This makes huge amount of hypothesis and we plan to reduce getting partial information form Language Model. The result of parsing one hypothesis is:

$$P\left(W_1^K\right) = \frac{max\_level}{N+1} \quad (5.1)$$

Where: *max_level* – maximal level - row on CKY table where sequence is parsed (network is active)
N- number of words in phrase

Result is used in equation (1.2) but only after finishing whole sentence. Because we decide to limit size of network to maximum 9 words – we apply it only to first 9 recognized words of sentence. Better solution can be detection of phrases using network – finding the longest word sequence, that gives the best result with preference to longer phrases (while for shorter is easy to be accepted like noun and adjective satisfied agreement rule). Another solution can be parallel application of at least two networks to neighbor parts of sentence.

6. Words order and acceleration of learning.

We can easily notice that such presented network is not prepared for application of the same rules when the order of words will change. As we know their order is often not strict in Slavic languages. The solution is to learn each SOM network with given data and next the data of network, which inputs are this network mirror in the same cell. For example first network in cell 2,0 has input from cell 1,0 and POS3. Second network in cell 2,0 has input form POS1 and cell 1,1. The similarity is that one input is coming from level 0 (POS) and next from layer 1 (cells 1,0, 1,1). In case of learning first network by inputs of its "mirrored" network we need only to change the order of inputs: first will be POS3 and next output from cell 1,1. Next the weight if first network in cell 2,0 is copied to "mirrored" network, but now we have to change back its inputs.

Above idea will be correct when we introduce the idea of acceleration of learning, which also make weights of some networks the same. Because the inputs bring similar kind of information (POS) we can learn network in cell 1,0 giving inputs of cells 1,1, 1,2, etc. in respective order. Next we copy weights of network from cell 1,0 to each network in layer 1. Next we can notice that inputs of cell 2,0 brings the same kind information as inputs of each cell in layer 2, so we repeat above learning but according not to networks, but to the cells as a whole (finally copying the weights of respective networks from cell 2,0 to other cell in layer 2). Above procedure we repeat to layer before the highest layer we use for given length of input sentence/part of sentence.

7. Experiments.

For speech recognition task we create software that simulates real LVCSR system. Instead of recognition of speech we recognized string of letters, that was concatenation of letters representing words, where we add "noise" – we randomly change some letters (with probability 0.02) to simulate spoken phonemes not exactly matching to patterns.

We trained network on IPI PAN Corpus of Polish Language (Przepiórkowski 2004). Training data consist 270 000 words, 14 000 sentences, 38 000 phrases, we throw such sentences, which phrases have more than 10 words, so we used 8 200 sentences, 28 000 phrases. Validation set has 492 phrases (111 sentences).

Recognition task consist of 12 sentences (233 words) from IPI PAN Corpus and give Word Error Rate (WER) 68,1 % without Language Module and 67,6 % with CKY network in Language Module.

Such high WER is consequence of simplified simulation of LVCSR system, which doesn't cover some important effects, so experiments performed on real system are necessary. In case when we don't add "noise" as above we get WER=0%.

8. Conclussions.

We presented new solution of language modeling task. Our model remains further development for full paring of longer sentences and efficient generating of word category hypothesis. Results are encouraging, but we prepare for recognition task performed on real LVCSR system.

References

Benesty, J. Sondhi, M. M., Huang, Y. (2008), *Springer Handbook of Speech Processing,* Springer-Verlag, Berlin, Heidelberg.

Brocki, Ł,Korzinek, D. (2007), *Grammar Based Automatic Speech Recognition System for the Polish Language*, in: R.Jaboński, M.Turkowski, R.Szewczyk (eds.) Recent Advances in Mechatronics, pp. 87-91.

Brocki, Ł,Korzinek, D. (2008), Marasek, K.*„Telephony Based Voice Portal for a University*, Speech and Language Technology, Vol. 11,pp 55-58.

Duch, W., Korbicz, J.,Rutkowski, L., Tadeusiewicz, R. (2006), (eds.) Tom 6: Sieci Neuronowe, in: Nałęcz, M. (eds.) Biocybernetyka i inżynieria biomedyczna 2000; Exit,PTSN

Duchateau, J., (1998) *HMM based acoustic modeling in large vocabulary speech recognition*, PhD thesis, Katholieke Universiteit Leuven,Belgium.

Erdogan, H., Sarikaya, R., Chen, S. F., Gao, Y., Picheny, M. (2005) Using semantic analysis to improve speech recognition performance, Computer Speech and Language 19.

ESAT-PSI (2006) *Description of the ESAT speech recognition system January 2006*, PSI - Speech Group, Katholieke Universiteit Leuven, Belgium. http://www.esat.kuleuven.be/psi/spraak/

Hnatkowska, B., Sas, J.,(2008) A*pplication of Automatic Speech Recognition to medical reports.,* Journal of Medical Informatics and Technologies, Vol. 12.

Kohonen, T. (2001). *Self-Organizing Maps.* Third, extended edition. Springer.

Markowitz, J. A. (1996), *Using speech recognition*, Prentice Hall PTR.

Przepiórkowski A. (2004) Korpus IPI PAN. *Wersja wstępna.* Instytut Podstaw Informatyki PAN, Warszawa, http://korpus.pl.

Robinson, A., J. (1994), *An Application of Recurrent Nets to Phone Probability Estimation,* IEE Transaction on Neural Networks, Vol. 5, No. 2,March, pp. 298-304

Szymański , M. ,Ogórkiewicz, J., Lange, M., Klessa, K., Grocholewski, S., Demenko, G. (2008), *First evaluation of Polish LVCSR acoustic models obtained fom the JURISDIC database*, Speech and Language Technology, Vol. 11,

Tadeusiewicz, R., (1988) *Sygnał mowy,* Warszawa : Wydawnictwa Komunikacji i Łączności.

Tadeusiewicz R. (1994) *Zastosowanie sieci neuronowych do rozpoznawania mowy,* in Richter L. (eds.): *Analiza, Synteza i Rozpoznawanie Sygnału Mowy dla Celów Automatyki, Informatyki, Lingwistyki i Medycyny, Polska Akademia Nauk,* IPPT, Warszawa 1994, pp. 137-151

Tadeusiewicz, R., Gąciarz, T., Borowik, B., Leper, B. (2007) *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Wydawnictwo Polskiej Akademii Umiejętności, Kraków

Wu Chou, Biing Hwang Juang(eds.) (2003), *Pattern Recognition in speech and language processing,* Boca Raton : CRC Press, 2003.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., (2006) *HTK Book,* Cambridge University Engineering Department.