Word acquisition with neural networks based on the state of a small world: Handling multiple question types

Mats U. Nystrand[1]
Kazuhiro Ueda[1]
Naoto Takahashi[2]

[1]Department of General Systems Studies
Graduate School of International and Interdisciplinary Studies
University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, JAPAN
Tel: +81-3-5454-6798 Fax: +81-3-5454-6990
Mail: mats@gogh.c.u-tokyo.ac.jp, ueda@gregorio.c.u-tokyo.ac.jp

[2] National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, 305-8568, JAPAN
Tel: +81-298-61-5168 Fax: +81-298-61-5891
Mail: naoto.takahashi@aist.go.jp

Abstract
The major part of all words in natural language is related to a state or a change of states in our environment. For a computer, it is not completely trivial to learn the relationship between a word and a state – a word can be used in many different states and a state can be expressed with various words. In this paper, we present a neural network, which can be trained to answer short questions (being input as a series of words) relating to the state of a simulated small world. We also show that dividing the network into modules increases the overall performance of the network when it is being trained on different question types.

Key words
Question & Answering Systems, Language Acquisition, Neural Networks, Modular Networks

Introduction
In general, a language and its elements are used to describe some state or change of states. In the case of natural language, most of the elements (or the words) are referring to some state, which is not related to the language itself, but rather to some object or event in our environment. For a newborn child, this relation between the words and the environment is not known, but has to be learned over time. It is also important to note that when a child learns a language, in most cases it also has access to the environment to which the language refers.

Programming a computer to learn a language has proven to be very difficult. An important reason for this is that it is very hard to define, in terms of a computer program, all the relationships that exist between the words and the environment to which they refer.

In order to partially deal with these problems, we below describe a neural network model, which makes it decisions based on two sources of information: one input for language and one input for a simulated small world (a virtual environment). Both these inputs are dynamic and their representation will be explained further below. The overall task of the system is to learn to answer questions (the language input) about the state of the small world (the small world input).

Model
We have designed two neural network models: one plain (non-modular) network and one modular network. Their structures are illustrated in *figure 1*.

The structure of the plain network was first described in a paper by Nystrand, Ueda and Takahashi (2000) and consists of an input,

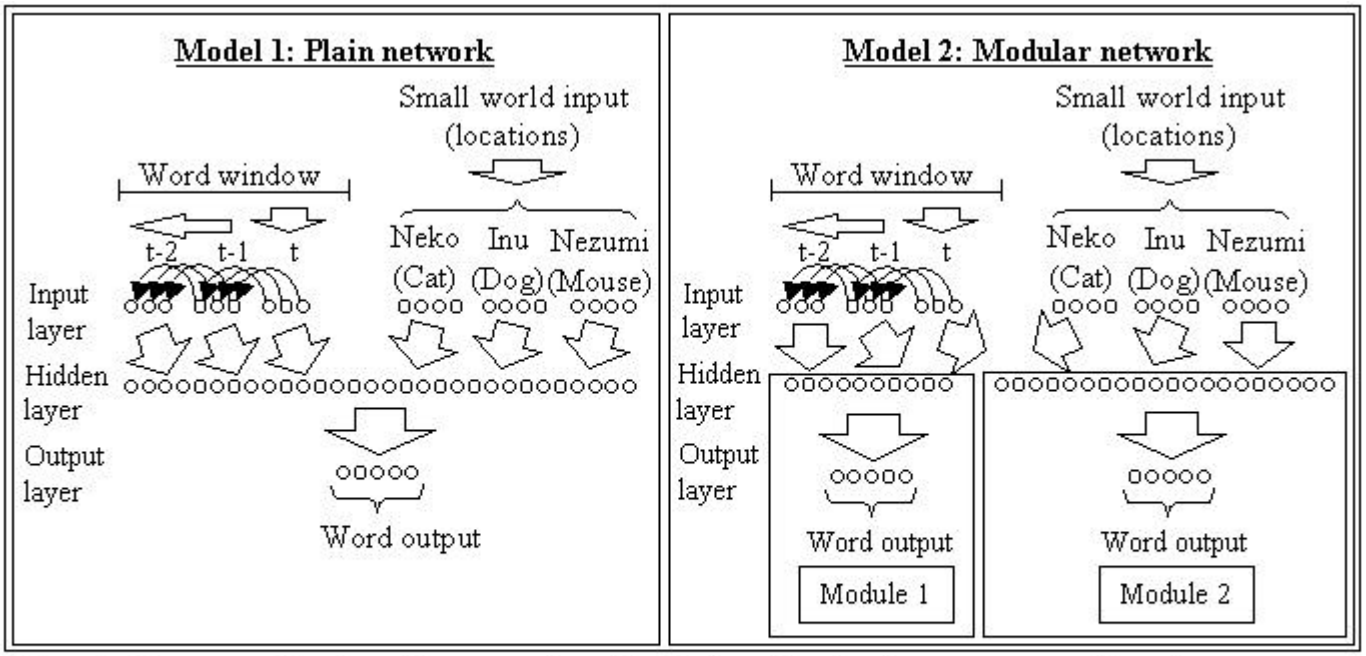

Figure 1: *Plain network and modular network models. Both models take the same input and have the same number of links between the input, hidden and output layers. However, the output from the modular network comes either from module 1 or module 2 depending on what type of question is being inputted in the word window.*

hidden and output layer with full forward connectivity between the input and the hidden layer as well as between the hidden and the output layer. The modular network is based on two modules, where each module consists of a hidden layer to which the input layer of the whole network is fully connected. The hidden and output layers within each module are also fully connected whereas there are no links between the modules. Which module is chosen for the output depends on what type of question is inputted in the word window (explained further below).

The most important difference between the plain network and the modular network is that the output in the modular network is solely determined either by module 1 or by module 2 – not both. The output from the plain network is, however, determined by the whole network. Both the plain and the modular network take the same input and have the same number of links between the input, hidden and output layers.

The representation of the input and output of words in both of the network models is realized with binary non-orthogonal (overlapping) vectors. In order to reduce the computational load, some words are grouped together with a relating particle. All simulations were run with Japanese as language and the full set of words used for input and output is presented in Table 1.

The representation that we have chosen is different from representations found in other research with neural networks, such as Elman (1990), in which the words were represented as orthogonal vectors where each word corresponded to a certain element in the vector. The advantage with Elman's representation is that the network's representational capability can be tested without being affected by intrinsic similarities of the input. However, a disadvantage is that the vectors require one more element for each word that is added, which significantly raises the computational load of the system. An alternative to the binary word representation that we are applying in this paper, could be to use graded word vectors, where each element is not restricted to one of only two possible values, but rather can take any value within a certain range (see for instance Mikkulainen and Dyer, 1991). In this way, words with similar meanings can be encoded with similar word vectors. The binary representation of words that we are using has not been adjusted to represent specific similarities between the word themselves,

Table 1 Word Representation

Word Input*			Word Output**		
Word unit (Japanese)	English translation	Binary value	Word unit (Japanese)	English translation	Binary value
neko ha	the cat	000	heya 0 – 15	room 0 – 15	00000–01111
inu ha	the dog	001	neko	cat	10000
nezumi ha	the mouse	010	inu	dog	10001
dare to	with whom	011	nezumi	mouse	10010
doko ni	where	100	Dare to mo issho ni imasen	Not together with anyone.	10011
imasu ka?	is	101	[Not used]		10100–11111
[Not used]		110–111			

** The word input is represented through three neurons, which together can be seen as a binary value corresponding to one of the word units in the left part of the table.*

*** The word output is represented through five neurons, which similarly to the input can be interpreted as a binary value corresponding to one of the word units in the right half of the table.*

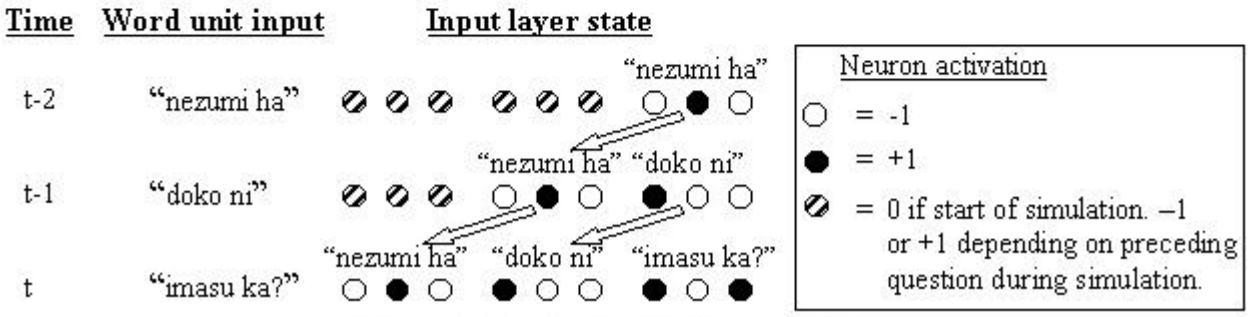

***Figure 2**: Question input in the word window*

however, this representation is similar to the one facing children when they are confronted with a language for the first time. That is, there are many words in the natural language, which look or sound similar but have very different meaning, e.g. lake, bake and sake. This also applies to the binary word representation that we are using.

Further, in order to be able to input sequences of words to the network, as when asking the network a question, we use a word window of length three. That is, for each iteration in the simulation, the words are shifted one step to the left, where the left most word is discarded, and the binary representation of a new word is applied to the right most neurons as illustrated in *figure 2*. The reason for that we have chosen this kind of representation is that we think that it is has several similarities to how we have to process (verbal) information ourselves, i.e. data is presented in a sequential stream and we have a limit for how much of the stream that we concurrently can process. In addition, with the representation above, it is possible to study what the "expected" output of the network will be as it is confronted with only the first or two first words of a question. (This is not the focus of the paper, but makes the model more general).

The second type of input comes from a small world (a simulated virtual environment), which is depicted in *figure 3*. In our experiment, the small world simply consists of 16 rooms and three objects: "neko" (cat), "inu" (dog) and "nezumi" (mouse). The world is dynamic in the sense that the object positions can change with each time step. The state of the small world is represented with 12 input neurons in the network, where four neurons are allocated to each object as a position indicator. The position is binary encoded as a number between 0 and 15 indicating the number of the room in which the object is. That is, the existence of the objects themselves is implicitly encoded through the input of the three separate position indicators as shown in *figure 3*.

The hidden layer consists of 30 neurons in both the plain and the modular network.

Table 2 Question types and questions used during the simulation

Question type 1: X ha doko ni imasu ka? (Where is X?)

Questions divided into word units			English translation
neko ha (the cat)	doko ni (where)	imasu ka? (is)	Where is the cat?
inu ha (the dog)	doko ni (where)	imasu ka? (is)	Where is the dog?
nezumi ha (the mouse)	doko ni (where)	imasu ka? (is)	Where is the mouse?

Question type 2: X ha dare to imasu ka? (Who is X together with?)

Questions divided into word units			English translation
neko ha (the cat)	dare to (with whom)	imasu ka? (is)	Who is the cat together with?
inu ha (the dog)	dare to (with whom)	imasu ka? (is)	Who is the dog together with?
nezumi ha (the mouse)	dare to (with whom)	imasu ka? (is)	Who is the mouse together with?

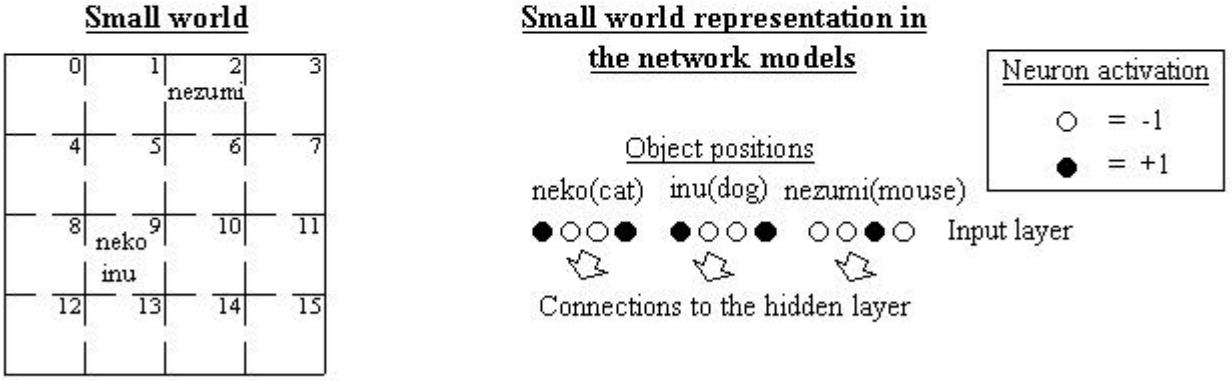

***Figure 3**: The small world and its representation in the network models. The neurons in the input layer form a binary representation of the room numbers in the small world, where "nezumi" (the mouse) is alone in room 2 and "neko" (the cat) and "inu" (the dog) are in the same location (room 9).*

The output of the network models consists of one of the words shown in Table 1. Since the output neurons can take any value between –1.0 and +1.0, the binary representation of the output is obtained by regarding each neuron as a bit in the output vector where the bit value is 0 if the activation of the neuron is negative and 1 if the activation is positive.

Thus, the network models described above have no prior knowledge about the relationship between the words that are being inputted through the word window and the objects existing in the small world.

Simulation

We trained the plain and the modular network models on the same learning task, which consisted of answering correctly to questions about the state of the small world. During the training, we used the two types of questions presented in Table 2. Each question type contains three possible variants. As mentioned above, when the simulation starts, the network models have no prior knowledge about the relationship between the words and the objects in the small world; they have to acquire this knowledge during the training. Since the small world is dynamic, it is important to note that, in order for the network models to answer the questions correctly, they must be able to recognize the relationship between the words and the small world.

We used the back propagation algorithm to train the network models on answering the questions in Table 2. For the modular network, we selectively allocated one module for each type of question. That is, if the question being inputted belonged to question type 1, we only used the output from module 1 and back propagated the errors within this module only. Similarly, if the question belonged to question type 2, we used the output from module 2 and back propagated the errors through the links connected to this module only. As a result, in the modular network, each module became specialized in answering questions of a certain type.

For each iteration in the simulation, one word was inputted in the word window as was

Table 3 Correct answer ratio and average mean square error for the plain network and for five variations of the modular network after 200,000 iterations.

	Plain network	Modular network				
		5–25*	10–20*	15–15*	20–10*	25–5*
Correct answer ratio**	35.30%	57.70%	70.40%	64.90%	61.90%	57.40%
Average MSE**	1.63	0.97	0.56	0.61	0.67	0.64

* *Indicates the number of neurons in the hidden layer for module 1 and module 2*
** *Measured over 1,000 iterations*

shown in *figure 2*, and the state of the small world was updated (details below). The weights of the network were also adjusted after each iteration through the back-propagation algorithm. That is, although only the first word of a question had been presented to the network, the network was trained on predicting the right answer for that question[1]. The desired output vector of the network was set to the word output vector (in Table 1) that corresponds to the correct answer of the question being inputted.

The state of the small world was updated after each iteration with a constrained form of dynamics, where always at least two objects (randomly chosen at each iteration) were placed in the same room (also randomly chosen). The reason for that we used this kind of constraint was that it raised the probability that the correct answer to questions of type 2 would not be the output vector "Dare to mo issho ni imasen" ("Not together with anyone"), but rather one of the word units "neko" (cat), "inu" (dog) or "nezumi" (mouse) (see Table 1). As a result of this constraint, the objects were not restricted to move only between adjacent rooms, but they could "jump" from one room to any other room between the iterations. Thus, the state of the small world changed every time a new word was inputted in the word window.

When a question had been completely inputted (after 3 iterations) another one was randomly selected from Table 2 and was posed to the networks in the same manner. The simulations were run for 200,000 iterations (i.e. each question was asked an average of 11,111 times), reaching a level where the learning curves of the network models were close to constant.

Results

We ran the simulation above for the plain network and for a number of variants of the modular network, where we changed the number of hidden neurons within module 1 and module 2 (still keeping the total number of hidden neurons constant at 30). The variations we chose for the modular network were 5-25, 10-20, 15-15, 20-10 and 25-5 (where the first figure indicates the number of hidden neurons in module 1 and the second figure specifies the number of hidden neurons in module 2).

The results are presented in Table 3. The average mean square error has been calculated as the mean of the instantaneous square errors over the last 1,000 iterations in the simulation. The correct answer ratio has been calculated as the ratio of the number of correct answers produced by the network over the same period.

As the table shows, after 200,000 iterations, the number of correct answers produced by the plain network model was only 35.3%. This can be compared to the results from training the same kind of network on only one of the question types, as was done by Nystrand, Ueda and Takahashi (2000), where the network was able to give the correct answer in 88.4% of the cases for questions of type 1 and 60.6% of the cases for questions of type 2. It is, however, important to note that when the plain network is trained on both types of questions at the same time, the theoretical optimal correct answer ratio is not 100% since when the first word unit of a sentence (i.e. "neko ha" [the cat], "inu ha" [the dog] or "nezumi ha" [the mouse]), is presented to the network it does not know whether the question will be of type 1 or type 2. But, considering that it has a 50% chance of

[1] We used this kind of training since we also wanted to study how the word output ("the expectations") changed in the network on a word-by-word basis during the input of a question. This is not, however, the focus of this paper.

guessing which question type is being used as it sees the first word, the network should at least be able to reach a level of 83.3% correct answers if it accurately realizes the relationship between the words and the environment. Therefore, we cannot say that the plain network model has been successful in learning to answer the two types of questions.

However, when the network was forced to use separate modules for each type of question, as in the case with the modular network, the performance increased considerably. The best correct answer ratio for the module configurations tested was obtained for the modular network whose module 1 contained 10 neurons and module 2 contained 20 neurons (below called the 10-20 modular network). In this case, the correct answer ratio was 70.4% and the average mean square error was 0.56.

The fact that the modular configuration with 10 hidden neurons in module 1 and 20 hidden neurons in module 2 gave the best result, reflects the increased complexity of questions of type 2 compared to questions of type 1. In order to answer the questions of type 1 ("Where is X?"), the network only had to regard four of the neurons in the small world input to find the right answer. On the other hand, in order to answer questions of type 2 ("Who is X together with?") the network had to make a comparison between the three groups of location neurons in the small world input, before it could judge whether an object was in the same room as another object. Since the questions of type 1 and type 2 were posed to the network an equal number of times (statistically), the modular network could improve its overall performance when it used more hidden neurons for questions of type 2 rather than type 1.

These results indicate that the influence of interference (or spatial cross-talk) within the plain network was considerable as it was being trained on two question types interchangeably. It is, however, important to mention that the simulations above for the modular network do not provide any mechanism for separating the question types automatically based on the input patterns. It only shows that if we can separate the two question types, the overall performance of the system can be improved by allocating each question type to separate networks. How such an automatic task allocation can be implemented has been in focus of many papers (e.g. Jacobs, Jordan, Barto 1991 and Ronco, Gollee and Gawthrop 1997). We would also like to stress that the mean square error and the correct answer ratio above show how well the different network models were able to learn to answer correctly to the specific questions defined in table 2, given the representation of the small world presented in figure 3. The measures do not represent the generalization capability of the network, since they are based on data that also was used during training.

Conclusion

The basis for this paper was that natural language is most often used for expressing some state or change of states in our environment. We therefore developed a model, which takes two types of input – language and environmental ("small world") input. The model was implemented using both a plain feed-forward neural network and variations of modular networks. Each network was trained on answering short questions (being input as a series of words) relating to the state of the small world.

We found that a plain feed-forward three-layer neural network suffers from spatial cross talk when it is interchangeably being trained on answering different types of questions, and its performance decreases as a result of this. We could show, however, that if we consistently allocate the different types of questions to separate modules, the performance of the network increases. The best result obtained, when training the network on two types of questions at the same time with a modular network, was a correct answer ratio of 70.4%, given the fixed number of links and assumptions made in the model.

References

Elman, J. L. (1990). Finding Structure in Time, *Cognitive Science*, **14**, 179-211.

Jacobs R. A., Jordan M.I. & Barto A.G. (1991). Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks. *Cognitive Science*, **15**, 219-250.

Miikkulainen R. & Dyer M.G (1991). Natural Language Processing With Modular PDP Networks and Distributed Lexicon. *Cognitive Science*, **15**, 343-399.

Nystrand M.U., Ueda K. & Takahashi N. (2000). Acquiring word meanings in a small world using neural networks. Technical Report of IEICE, TL2000-30 December. The Institute of Electronics, Information and Communication Engineering, 39-46.

Ronco E., Gollee H. & Gawthrop (1997). Modular Neural Networks and Self-Decomposition. Technical Report: CSC-96012, February 11.