

University of East London Institutional Repository: <http://roar.uel.ac.uk>

This paper is made available online in accordance with publisher policies. Please scroll down to view the document itself. Please refer to the repository record for this item and our policy information available from the repository home page for further information.

To see the final version of this paper please visit the publisher's website. Access to the published version may require a subscription.

Author(s): Palmer-Brown, Dominic; Draganova, Chrisina; Lee, Sin Wee.

Article title: Snap-Drift Neural Network for Selecting Student Feedback

Year of publication: 2009

Citation: Palmer-Brown, D; Draganova, C; Lee, S.W. (2009) 'Snap-Drift Neural Network for Selecting Student Feedback' International Joint Conference on Neural Networks (IJCNN 09) pp.391-398

Link to published version: <http://dx.doi.org/10.1109/IJCNN.2009.5178859>

DOI: 10.1109/IJCNN.2009.5178859

Snap-Drift Neural Network for Selecting Student Feedback

Dominic-Palmer Brown, Chrisina Draganova, Sin Wee Lee

Abstract—This paper investigates the application of the snap-drift neural network (SDNN) to the provision of guided student learning in formative assessments. SDNN is able to adapt rapidly by performing a combination of fast, convergent, minimal intersection learning (snap) and Learning Vector Quantization (drift) to capture both precise sub-features in the data and more general holistic features. Snap and drift are combined within a modal learning system that toggles its learning style between the two modes. In this particular application the SDNN is trained with responses from past students to Multiple Choice Questions (MCQs). The neural network is able to categorise the learner's responses as having a significant level of similarity with a subset of the students it has previously categorised. Each category is associated with feedback composed by the lecturer on the basis of the level of understanding and prevalent misconceptions of that category-group of students. The feedback addresses the level of knowledge of the individual and guides them towards a greater understanding of particular concepts. The trained snap-drift neural network is integrated into an on-line Multiple Choice Questions (MCQs) system. This approach has been implemented and trialled with two cohorts of students using data sets of student answers related to a topic from an Introduction to Computer System module. Results indicate that significant learning support is provided for the students.

I. INTRODUCTION

THE Snap-Drift Neural Network (SDNN) is able to adapt rapidly in a non-stationary environment where new patterns are introduced over time. The unsupervised algorithm has proved invaluable for continuous learning in many applications, such as in the analysis and interpretation of data representing interactions between trainee network managers and a simulated network management system [1], where new patterns of the user behaviour were revealed. The classifier form of Snap-drift [2] has been used to recognize phrases extracted from Lancaster Parsed Corpus (LPC) [3]. Comparisons between snap-drift and MLP with back-propagation show that the former is faster and just as effective. Whenever running snap-drift it is easy to compare with LVQ by disabling the snap, since the drift by itself is

LVQ. Snap-drift has also been used for phonetic feature discovery in speech [4] where SDNN groups the speech input waveforms and extracts properties which are common to stammering and non-stammering speech, thus supporting classification.

This paper describes a novel application of SDNN, in unsupervised form, as an automatic diagnostic tool in a virtual learning environment, in which the students are given feedback based on their level of knowledge. This is achieved by developing a method that integrates an easy to use on-line system and the SDNN based analysis and grouping of student responses to multiple choice questions. The result is feedback based on the individual responses of the student and not tied to any particular question, so the learner is encouraged to retake the same test, receiving different feedbacks depending on their evolving state of knowledge.

In this application it is essential to have a system capable of forming groups that correspond to one or more identical question responses out of 5. Snap-drift is suitable because it is an unsupervised, easy-to-apply, quick and effective means of discovering groupings, and is capable of discovering both clearly separable clusters (drift) and groups that are characterized by precise features that may represent only a fraction of the structure of patterns (snap).

II. SNAP-DRIFT NEURAL NETWORK (SDNN)

Snap-drift is a modal learning approach. Snap is pattern intersection learning which was first used in a different way in combination with other methods as part of Adaptive Resonance Theory (ART) [5]. Drift is LVQ. Snap and drift learning modes are combined within a system that toggles its learning between the two modes. On presentation of input data patterns at the input layer $F1_1$, the distributed SDNN (dSDNN) will learn to group them according to their features using snap-drift, finding the highest D node matches [6]. Weight vectors are normalised so that in effect only the angle of the weight vector is adapted. The output winning neurons from dSDNN act as inputs to the selection SDNN (sSDNN) module for the purpose of feature grouping and this layer is also subject to snap-drift learning, and has only a single winning neuron for each pattern, thus enabling grouping.

In summary, SDNN toggles its learning mode to find a rich set of features in the data and uses them to group the data into categories. Each weight vector is bounded by snap and drift: snapping gives the angle of the minimum values (on all dimensions) and drifting gives the average angle of the patterns grouped under the neuron. Snapping essentially provides an anchor vector pointing at the 'bottom left hand corner' of the pattern group for which the neuron wins. This

Manuscript received January 5, 2009.

D. Palmer-Brown is with London Metropolitan University, Faculty of Computing, 166-220 Holloway Road, London N17 8DB, UK (phone: +44 (0) 207 133 4003; fax +44 (0) 207 133 4741; e-mail: d.palmer-brown@londonmet.ac.uk).

C. Draganova is with University of East London, School of Computing, Information Technology and Engineering, Docklands Campus, University Way, London E16 2RD (e-mail: c.draganova@uel.ac.uk).

S. Wee Lee is with University of East London, School of Computing, Information Technology and Engineering, Docklands Campus, University Way, London E16 2RD (e-mail: s.w.lee@uel.ac.uk).

represents a feature common to all the patterns in the group and gives a high probability of rapid (in terms of epochs) convergence (both snap and drift are convergent, but snap is faster). Drifting, which uses Learning Vector Quantization (LVQ) [7], tilts the vector towards the centroid angle of the group and ensures that an average, generalised feature is included in the final vector. The angular range of the pattern-group membership depends on the proximity of neighbouring groups (natural competition). The output winning neurons from dSDNN act as input data to the selection SDNN (sSDNN) module. This layer is also subject to snap-drift learning and has an activation threshold to ensure if a good match to an existing neuron is not found a new neuron is recruited, so the progress of learning determines the number of output groups.

III. E- LEARNING SNAP-DRIFT NEURAL NETWORK (ESDNN)

A. The ESDNN Architecture

In this application of snap-drift, ESDNN, the unsupervised version of the snap-drift algorithm is deployed [4], as shown in Fig. 1.

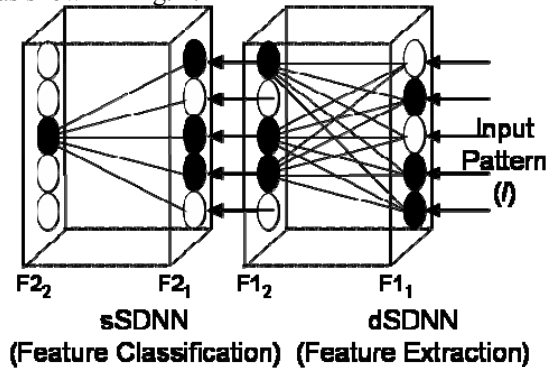


Fig. 1. ESDNN Architecture

During training, on presentation of an input pattern at the input layer $F1_1$, the distributed dSDNN will learn to group the input patterns according to their general features. In this case, the five $F1_2$ nodes, whose weight vectors best match the current input pattern (highest net input) are used as the input data to the selection sSDNN module for feature classification.

In the sSDNN module, a quality assurance threshold is introduced. If the net input to an sSDNN node is above the threshold, the node is accepted as the winner, otherwise a new uncommitted output node will be selected as the new winner and initialised with the current pattern from dSDNN.

B. The ESDNN Learning Algorithm

The learning algorithm combines logical intersection learning (snap) and Learning Vector Quantisation (drift) [7]. In general terms, the snap-drift algorithm can be stated as:

$$\text{Snap-drift} = \alpha(\text{pattern intersection}) + \sigma(LVQ), \quad (1)$$

where (α, σ) are equal to $(1,0)$ or $(0,1)$ depending on the learning mode: snap and drift respectively.

The learning of both of the modules dSDNN and sSDNN in the neural system is as follows:

$$w_{ji}^{(new)} = \alpha(I \cap w_{ji}^{(old)}) + \sigma(w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)})), \quad (2)$$

where w_{ji} are the weight vectors; I are the binary input vectors; β is the drift speed constant which must be less than 1, and in this case is equal to 0.25 for sSDNN and to 0.5 for dSDNN, and the operation \cap means logical AND intersection (or the fuzzy AND min in the case of real values as opposed to binary).

On each successive learning epoch, the learning is toggled between the two modes of learning. When $\alpha = 1$, fast, minimalist (snap) learning is invoked, causing (2) to become:

$$w_{ji}^{(new)} = I \cap w_{ji}^{(old)}, \quad (3)$$

This learns sub-features of patterns. In contrast, when $\sigma = 1$, (2) simplifies to:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)}), \quad (4)$$

which causes a simple form of clustering at a speed determined by β . Finally, the weights are normalised:

$$w_{ij}^{(new)} = w_{ji}^{(new)} / |w_{ji}^{(new)}|, \quad (5)$$

where $|w_{ji}^{(new)}|$ denotes the Euclidean norm of the vector $w_{ji}^{(new)}$.

The following is a summary of the steps that occur in ESDNN:

Step 1:

- Initialise parameters: $(\alpha = 1, \sigma = 0)$
- Set number of hidden nodes
- Set number of hidden winning nodes D

Step 2:

For each epoch (t)

Toggle the values (α, σ) between $(0, 1)$ and $(1, 0)$ after every successive epoch.

For each input pattern

Step 2.1: Find the D winning nodes at $F1_2$ with the largest net input

Step 2.2: Weights of dSDNN adapted according to the alternative learning procedure

Step 2.3: Process the output pattern of $F1_2$ as input pattern of $F2_1$

Step 2.4: Find the node at $F2_2$ with the largest net input

Step 2.5: Test the threshold condition:

IF (the net input of the nodes is greater than the threshold) **THEN**

Weights of the sSDNN output node adapted according to the alternative learning procedure

ELSE

An uncommitted sSDNN output node is selected and its weights are adapted according to the alternative learning procedure

IV. E-LEARNING SYSTEM

A. Motivation

Formative type of assessment provides students with feedback that highlights the areas for further study and indicates the degree of progress [8]. When students attempt on-line formative assessments they generate data that is invaluable for understanding their learning. That data is generally lost. We propose to capture the data and use it to provide immediate feedback to the students and to provide lecturers and tutors with a detailed picture of the learning of their students. Providing timely and frequent feedback that highlights the areas students need to study more and giving them an indication of their progress are vital in formative assessment.

There are many studies investigating the role of different types of feedback in web-based assessments that report positive results from the use of MCQs in online tests for formative assessments (e.g. [9]–[12]). In these studies it is assumed that all the possible errors for a question can be predicted and a generic and focused feedback can be written for that question. However, this kind feedback relates to a specific question rather than a combination of questions. The *diagnostic* feedback proposed here differs in that it does not reveal which questions were wrong; instead, the students are encouraged by the feedback to reflect on misunderstood concepts (that relate to their combined errors on all the questions), and then to attempt the test again.

Predicting all possible mistakes and writing generic and focused feedback for a combination of questions would be a daunting task and would not be feasible for large test banks (2 questions with 5 possible answers creates 25 possible answer combinations; 5 questions creates 3125 combinations, and so on). As the question bank grows the number of possible answer combinations increases exponentially, so that automation is essential for at least part of the process. The neural network approach proposed here cuts through these problems by providing an efficient means of discovering a relatively small numbers of groups of similar answers so that responses can be targeted to the answers given by a very wide range of students with different states of knowledge.

B. E-Learning System description

This version of ESDNN is the unsupervised version of the snap-drift algorithm, as shown in figure 1. The working of ESDNN can be divided into two phases, training and deployment. Ultimately, these two phases will be

mechanised and integrated into the MCQs online system, which we call E-learning system.

Before integration and deployment of the E-learning system, ESDNN is first trained with the students' responses for 5 questions on a particular topic in a module/subject. In this case, the responses are obtained from the previous cohorts of students on the topic 1 of the module, Introduction to Computer Systems. Before training, each of the response from the students is encoded into binary form, in preparation to be presented as input patterns for ESDNN. The format is as follows:

a – 00001; b – 00010 ; c – 00100; d – 01000 ; e – 10000

For example, a response such as [d, d, c, b, a] will be encoded into [0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1]. During training, on presentation of each input pattern, the ESDNN will learn to group the input patterns according to their general features. The groups are recorded. After training, appropriate feedback text is written by academics for each of the group of students' responses that address the conceptual errors implicit in combinations of incorrect answers.

The ESDNN is incorporated in an online self-assessment system. This system is designed and built using the JavaServer Faces Technology (JSF), which is a component-based web application framework that enables rapid development. The JSF follows the Model-View-Controller (MVC) design pattern and its architecture defines clear separation of the user interface from the application data and logic [13]. The ESDNN is integrated within the web application as part of the model layer. The ESDNN is trained for each set of questions offline with data available from previous cohorts of students, and the respective weight text files are stored on the application server. The feedback for each set of questions and each possible set of answers is grouped according to the classification from the ESDNN and written in an XML file stored on the application server.

During trials, students login into the system with their student id numbers. The student responses, time and student id are recorded in the database after each student's submission of answers. The students are prompted to select a module and a topic and this leads to the screen with a specific set of multiple choice questions. On submission of the answers the system converts these into a binary vector which is fed into the ESDNN. The ESDNN produces a group number; the system retrieves the corresponding feedback for this group from the XML feedback file and sends it to the student's browser. The student is prompted to go back and try the same questions again or select a different topic.

The student responses, recorded in the database can be used for monitoring the progress of the students and for identifying misunderstood concepts that can be addressed in following face-to-face sessions. The collected data can be also used to analyse how the feedback influences the learning of individual students and for retraining the neural network. Subsequently the content of the feedback can be improved. Once designed, MCQs and feedbacks can be reused for subsequent cohorts of students.

V. TRIALS AND RESULTS

A. Trials

Two different trials were conducted with 70 and 22 students. Students were allowed to make as many attempts at the questions as they liked. On average they gave 7 sets of answers over 20 minutes in the first trial and 12 sets of answers over 14 minutes in the second trial.

For the first trial [14] the ESDNN was trained with the responses for 5 questions on a particular topic of the module Introduction to Computer System obtained from previous cohort of students. After training, appropriate feedback text was written by academics for each of the group of students' responses that address the conceptual errors implicit in combinations of incorrect answers. During the trial, a current cohort of students was asked to provide responses on the same questions, they were given the feedback on the combination of incorrect answers and their responses recorded in the database.

The feedback texts are composed around the pattern groupings and are aimed at misconceptions that may have caused the incorrect answers common within the pattern group. An example of a typical response to the questions below is (d, d, b, c, c):

1. A common characteristic of all computer systems is that they

- lower the operating costs of companies that use them
- destroy jobs
- increase the efficiency of the companies that use them
- process inputs in order to produce outputs
- are used to violate our personal freedoms

2. A digital computer system generates, stores, and processes data in

- a hexadecimal form
- a decimal form
- an octal form
- a binary form
- none of the above forms

3. All modern, general purpose computer systems, require

- at least one CPU and memory to hold programs and data
- at least one CPU, memory to hold programs and data and I/O devices
- at least one CPU, memory to hold programs and data and long-term storage
- at least one CPU, I/O devices and long term storage
- at least one CPU, memory to hold programs and data, I/O devices and long-term storage

4. Babbage's 19th Century Analytical Engine is significant in the context of computing because it

- was the first digital computer
- was the first device which could be used to perform calculations

- contained all the essential elements of today's computers
- could process data in binary form
- was the first electronic computer

5. According to Von Neumann's stored program concept

- program instructions can be fetched from a storage device directly into the CPU
- data can be fetched from a storage device directly into the CPU
- memory locations are addressed by reference to their contents
- memory can hold programs but not data
- both program instructions and data are stored in memory while being processed

This is classified into Group 14, which generates the following feedback:

Group 14 Feedback

John Von Neumann, whose architecture forms the basis of modern computing, identified a number of major shortcomings in the ENIAC design. Chief amongst these was the difficulty of rewiring ENIAC's control panels every time a program or its data needed changing. To overcome this problem, Von Neumann proposed his stored program concept. This concept allows programs and their associated data to be changed easily.

Memory acts as a temporary storage location for both program instructions and data. Data, including program instructions, are copied from storage devices to memory and vice versa. This architecture was first proposed by John von Neumann.

Much of the flexibility of modern computer systems derives from the fact that memory is addressed by its location number without any regard for the data contained within. This is a crucial element of the Von Neumann architecture.

Prompted by the group 14 feedback the student is able, either immediately or after some reflection, to improve their answer to the question 5 to "both program instructions and data are stored in memory while being processed". This gives rise to the group 9 feedback below and after perhaps another couple of attempted answers they correct their answer to question 3, to achieve the correct answers to all questions.

Group 9 Feedback

The work of a modern computer system can be described in terms of an input-process-output model (IPO). To implement this model, a computer needs at least one means of both input and output and a means of processing the input. The design of Charles Babbage's Analytical Engine, which preceded the first digital computers by more than 100 years, also included a means of input (punched cards), a means of output (a printer) and a means of processing the input (a device which Babbage called the 'mill'). Babbage

was a genuine visionary.

For the second trial the responses generated during the first trial were used to train the ESDNN and new relevant feedback was written corresponding to the misconceptions identified in the SDNN learnt groups. The trial was conducted with different students than the first trial but from the same course level and from the same programme of study.

The data generated in the second trial was used to retrain the ESDNN in order to investigate how the states would change or stabilise.

B. Results

We call the groups which correspond to the output neurons, learning states or knowledge states. These are triggered by patterns of question topic responses. In other words, the winning neuron represents a state of learning because it captures some commonality in a set of question responses. For example, if there are several students who give the same answer (correct or incorrect) to two or more of the questions, snap-drift will form a group associated with one particular output neuron to include all such cases. That is an over simplification, because some of those cases may be pulled in to other ‘stronger’ states, but that would also be characterized by a common feature amongst the group (state) of responses.

Figures 2 and 3 illustrate the behaviour of students in terms of the learning states in the first and second trials, respectively, and show the knowledge state transitions. Each time a student gives a new set of answers, having received some feedback associated with their previous state, which in turn is based on their last answers, they are reclassified into a new (or the same) state, and thereby receive new (or the same) feedback. The tendency is to undergo a state transition either immediately, or after one additional attempt. A state is triggered (a neuron wins) if a particular combination of answers that has been learned by snap-drift is given for two or more questions. All groups are based on 2, 3 or 4 answers, with the other answers being irrelevant to the group. Because the groups correspond to more than 1 (and typically 3) identical answers within them, they also, upon inspection, correspond to knowledge groups.

A justification for calling the states ‘states of knowledge’ is to be found in their self-organization into the layers of Figure 2 and 3. For example, Figure 2 shows that a student on state 14 has to go via one of the states in the next layer such as state 9 before reaching the ‘state of perfect knowledge’ (state 25) which represents correct answers to all questions. On average, and unsurprisingly, the state-layer projecting onto state 25 (states 20, 1, 9 and 4) are associated with more correct answers than the states in the previous layer. Students often circulate within layers before proceeding to the next layer. They may also return to previous layer, but that is uncommon. The commonest finishing scores are 3, 4 and 5 out of 5 correct answers; the commonest starting scores are 0, 1, 2, and 3. The average time spent on the questions was about 17 minutes, and the

average increase in score was about 25%.

In the second trial all students (except one) arrive at (d, d, e) correct answer on the first three questions which corresponds to states 8 (d/c, d, e/a, c/d, e) or 25 (d, d, e, m, e), where ‘m’ means mixed. This result has been achieved after 3 attempts on average. After that all but one student arrive at the correct answer to the 5 questions. The average increase in score was about 40%, compared to 25% in the first trial. The time spent on the questions in the second was less than the time in the first trial and the average speed of transition in the second trial was higher than in the first trial. This result gives a clear indication that the feedback in the second trial is more effective than the one in the first trial.

After retraining the ESDNN for the third time with the data generated from the second trial it was observed that the new states formed by network are similar to those states used in the second trial. We define similarity between states as 2 or more of the same answers of the five questions, which is based on the fact that most learnt states have within them 2 or 3 consistent answers. There is strong sign of convergence of the states despite the different feedbacks and different students. This means that the previously written feedback is still relevant. However, it was observed that while only 2 states (states 8 and 25) correspond to answer (d, d, e, x, x) in the second trial, after retraining the ESDNN 6 new states similar to those 2 states were formed. Associated with this phenomenon, in the second trial most of the students arrive at state 25 and many circulate at least once on that state, resulting in the 6 states described above, so it is clear that the new states there are needed and that they need more specific feedback than state 25. The remaining new states are similar to the existing ones, and are catered for well by the existing feedbacks.

VI. CONCLUSION

This paper presents a method for using snap-drift in a diagnostic tool to provide feedback addressing the level of knowledge of individuals, guiding them towards a greater understanding of particular concepts.

It is possible to have some influence on the number of groups formed by snap-drift by modifying the learning constants and quality threshold, but no group can be formed without the similarities between patterns which are captured by snap and drift. In this application, a similarity in one of the five questions would be the minimum, but in fact almost all patterns have similarities of at least two with some other patterns and therefore making the network sensitive beyond the 2/5ths (40%) similarity makes no difference to the final groupings.

There is a clear evidence of convergence and stability across trials of most of the states formed by the neural network. This means that the feedback written for the initial two trials remains relevant. However there is also an indication of formation of substates in one case, which indicates that a mechanism for refining the feedback is needed for subsequent trials after retraining of the neural network with new data, in order to optimise the educational

effectiveness of the system. This is not surprising given that learning about learners presents a moving target problem if the feedbacks are changed, with new feedbacks producing new responses, and therefore new states which require new associated feedbacks. The good news is that this moving target problem appears to be quite small and convergent, suggesting that two or three trainings of the neural network will stabilise the states and therefore allow optimal feedbacks.

Whilst still at the preliminary stages of research into this complex educational application, the results from the two trials with students have provided very beneficial results as well as insights into the knowledge state transitions of the students. The next stage will be applying ESDNN for larger cohorts of students and using tests from other subject modules to move towards a generic approach and a system implementation that will be sufficiently lecturer and student friendly to ensure wide spread adoption.

REFERENCES

- [1] H. Donelan, C. Pattinson, D. Palmer-Brown, and S. W. Lee, "The Analysis of Network Manager's Behaviour using a Self-Organising Neural Networks", in *2004 Proc. 18th European Simulations Multiconference*, pp. 111 – 116.
- [2] S. W. Lee and D. Palmer-Brown, "Phrase Recognition using Snap-Drift Learning Algorithm", in *2005 Proc. The International Joint Conference on Neural Networks*, vol. 1, pp. 588-592.
- [3] R. Garside, G. Leech, and T. Varadi, "Manual of Information to Accompany the Lancaster Parsed Corpus", Department of English, University of Oslo, 1987.
- [4] S. W. Lee and D. Palmer-Brown, "Phonetic Feature Discovery in Speech using Snap-Drift", in *2006 Proc. International Conference on Artificial Neural Network*, S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 952 – 962.
- [5] S. Grossberg, "Adaptive Pattern Classification and Universal Recoding. I. Parallel Development and Coding of Neural Feature Detectors", *Biol. Cybern.*, vol. 23, pp 121-134, 1976.
- [6] S. W. Lee, D. Palmer-Brown, D., and C. M. Roadknight, "Performance-guided Neural Network for Rapidly Self-Organising Active Network Management", *Neurocomputing*, vol. 61, pp. 5 – 20, Oct. 2004.
- [7] T. Kohonen T., "Improved versions of learning vector quantization". In *1990 Proc. Int. Joint Conf. Neural Networks*, vol.1, pp. 545-550.
- [8] G. Brown, J. Bull., and M. Pendlebury, "Assessing Students Learning in Higher Education", London:Routledge, 1997.
- [9] E. Higgins and L. Tatham, "Exploring the potential of Multiple Choice Questions in Assessment", *Learning & Teaching in Action*. 2(1), 2003.
- [10] A. Payne, W.-P. Brinkman, and F. Wilson., "Towards Effective Feedback in e-Learning Packages: The Design of a Package to Support Literature Searching, Referencing and Avoiding Plagiarism", in *2007 Proc. of HCI2007 workshop: Design, use and experience of e-learning systems*, pp. 71-75.
- [11] G. A. Dafoulas, "The role of feedback in online learning communities", in *Proc. Fifth IEEE International Conf. on Advanced Learning Technologies*, July 2005, pp. 827 – 831.
- [12] N. Iahad, G.A. Dafoulas, E. Kalaitzakis, and L. A. Macaulay., "Evaluation of online assessment: the role of feedback in learner-centered e-learning", in *2004 Proc. 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 1 – vol. 1*, pp. 10006.1
- [13] Java Server Faces Technology, Available: <http://java.sun.com/javasee/javaserverfaces/>
- [14] S. W. Lee, D. Palmer-Brown and C. Draganova , "Diagnostic Feedback by Snap-drift Question Response Grouping", in *2008 Proc. 9th WSEAS International Conf. on Neural Networks (NN'08)*, pp. 208-214.

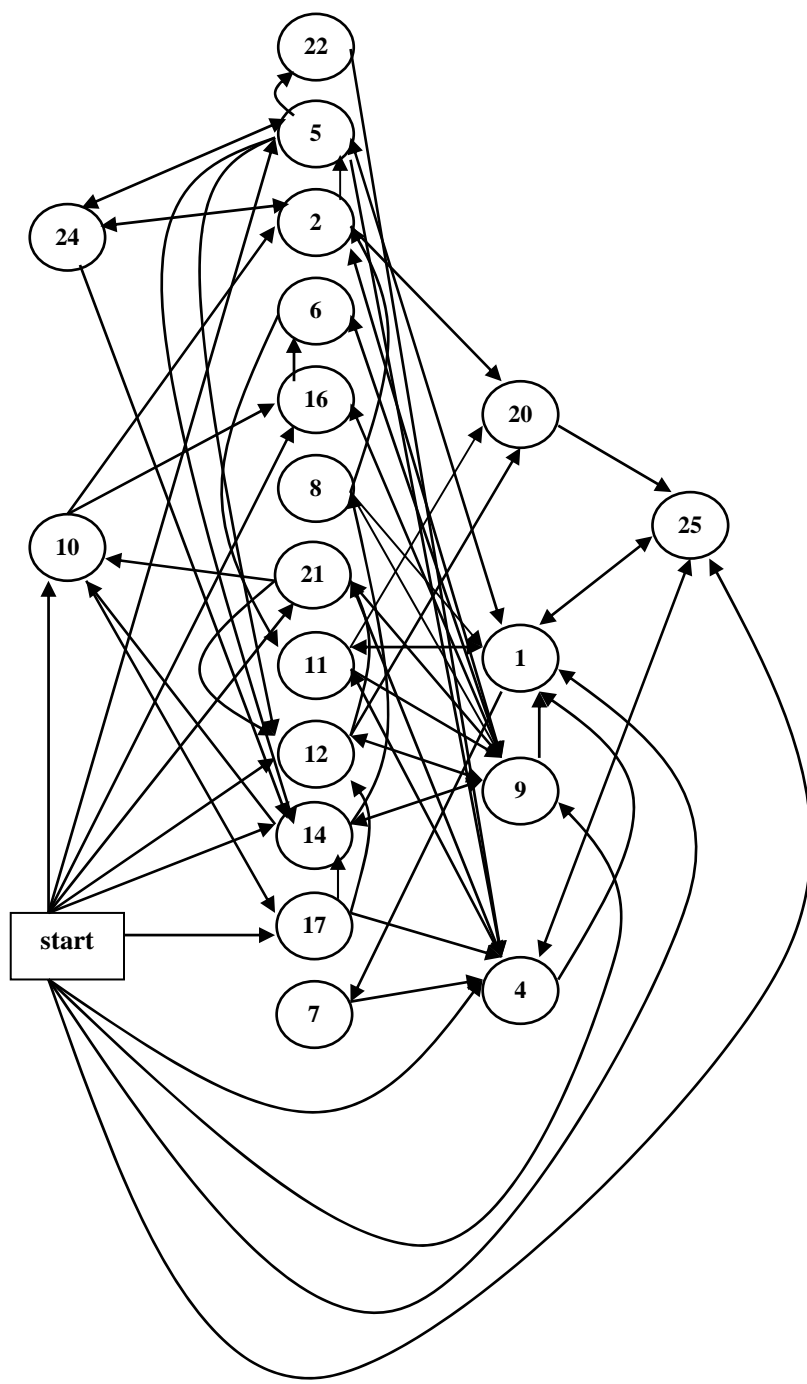


Fig. 2 Knowledge state transitions – trial 1

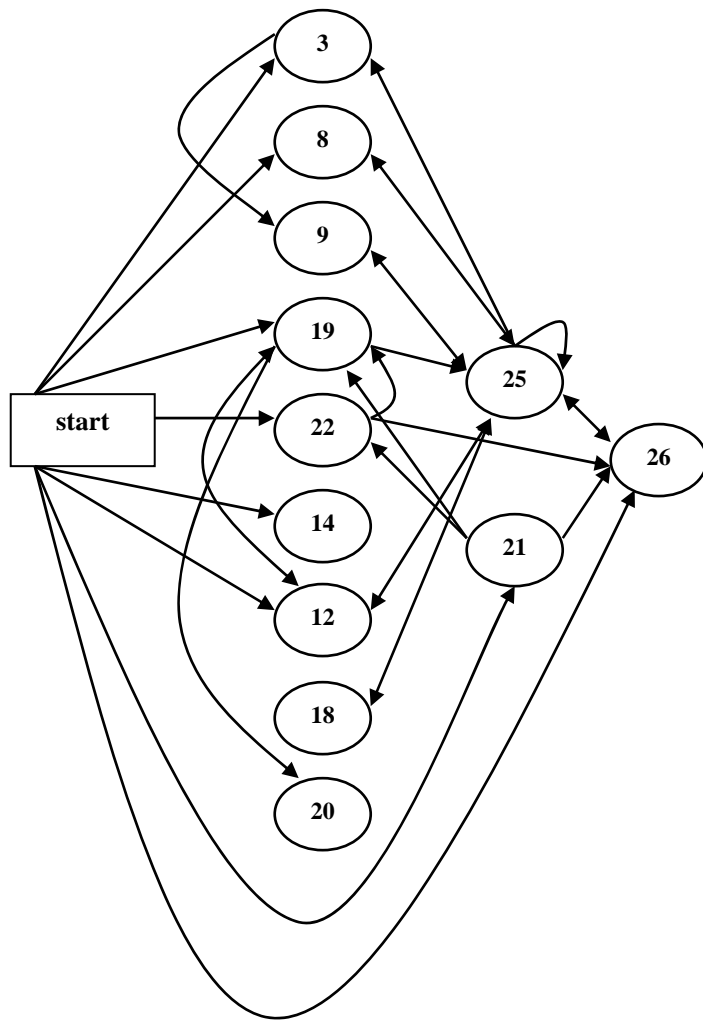


Fig. 3 Knowledge state transitions – trial 2