Dear Members of this list-serve:

Would it be possible to build stacked neural networks℗ like the one shown below? There are some questions you may want to ask. First, what is a stacked neural network? What is the difference between stacked neural networks and the existing neural network?

Here, four kinds of neural networks are described.

The first one is the general simple neural network. It usually consists of three layers, including one hidden layer. In this neural network, there is just feedback from the output layer to the input. Because there is no feedback or training for the hidden layers, it is impossible to solve any problems directly from the hidden layer. See Figure 1.

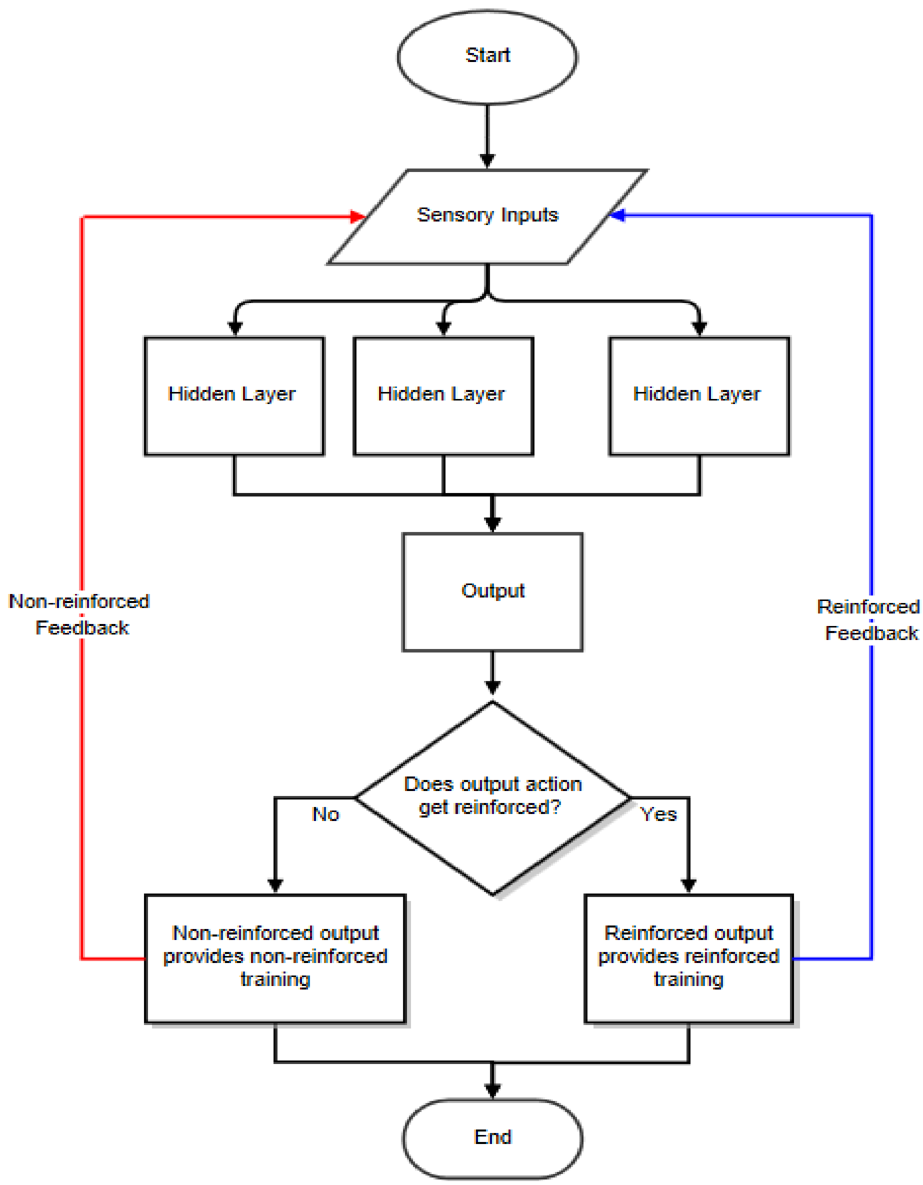


Figure 1: Simple Stacked Neural Network with Hidden Layers

The second type consists of only two stacks of neural networks. In these stacked neural networks, there is no feedback or training of the first layer of neural network. Feedback is provided only from the ultimate layer output to the input. See Figure 2.

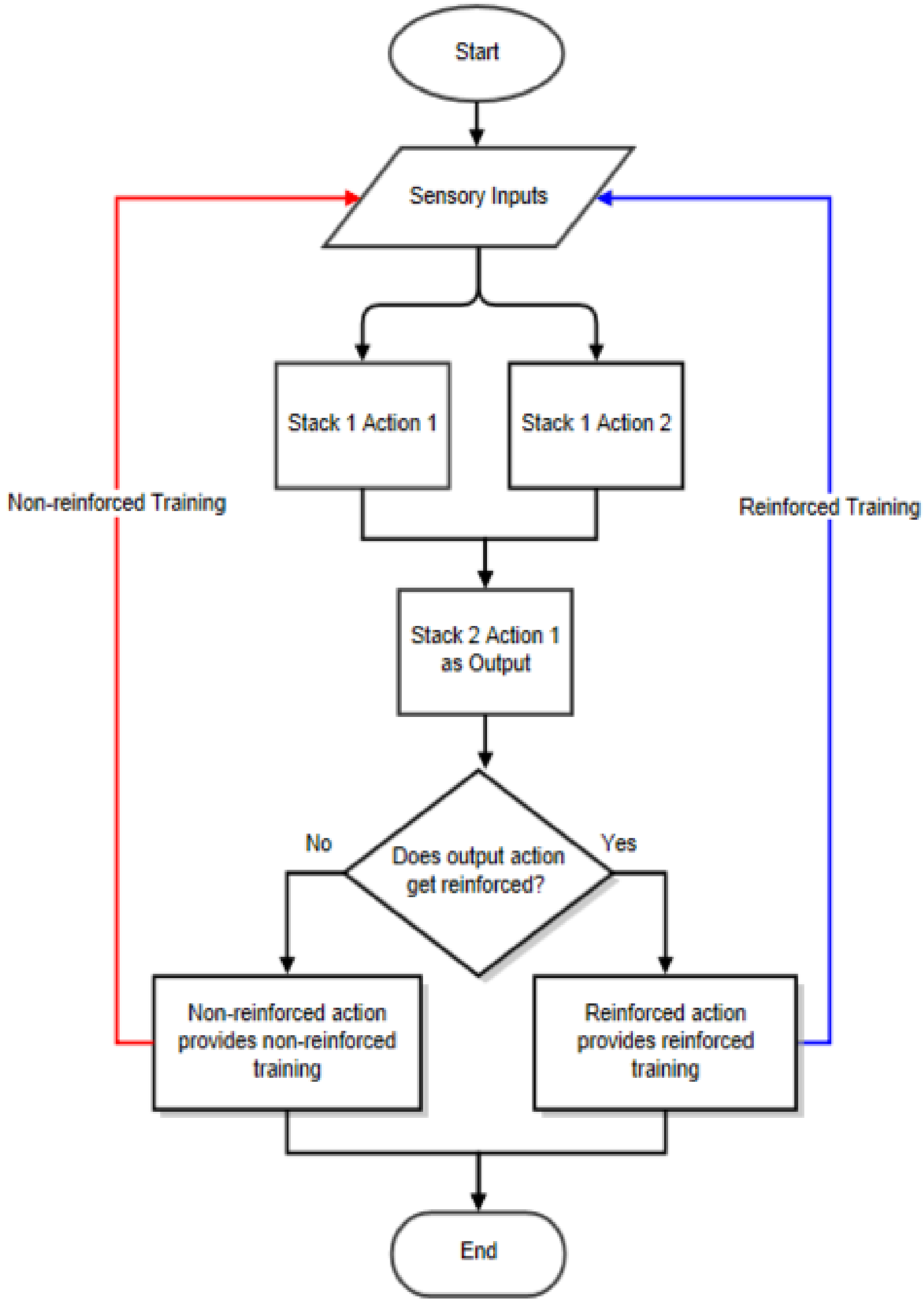


Figure 2: Two Layers of Stacked Neural Network without Intermediate Feedback

The third type is the only real stacked neural network. It consists of $n$ ($n \geq 3$) stacked neural networks. Training is carried out sequentially. The first layer is trained first, then the second layer being trained second and so on. Feedback and training is provided from each layer back to all the previous layers. See Figure 3. A description follows next. At the top of a stack of neural networks are the sensory inputs. Underneath the sensory inputs is the first layer of stacked neural networks. As can be seen, there are just 4 neural networks in the first layer in this simple diagram, producing actions 1, 2, 3 and 4. As shown by the variable number of arrows under each box, there might be differing numbers of such inputs into the first layer. The number of neural networks in the first layer may be very large. We only show four. The outputs from the first layer are fed into the second layer of stacked neural network networks in the second layer, and sometimes fed directly to higher layers in the stack. The second layer output is fed into the third layer and so on. Note, as different from traditional small stacks of neural networks, there is feedback from the real world outcomes to train at each layer and each network within a layer.

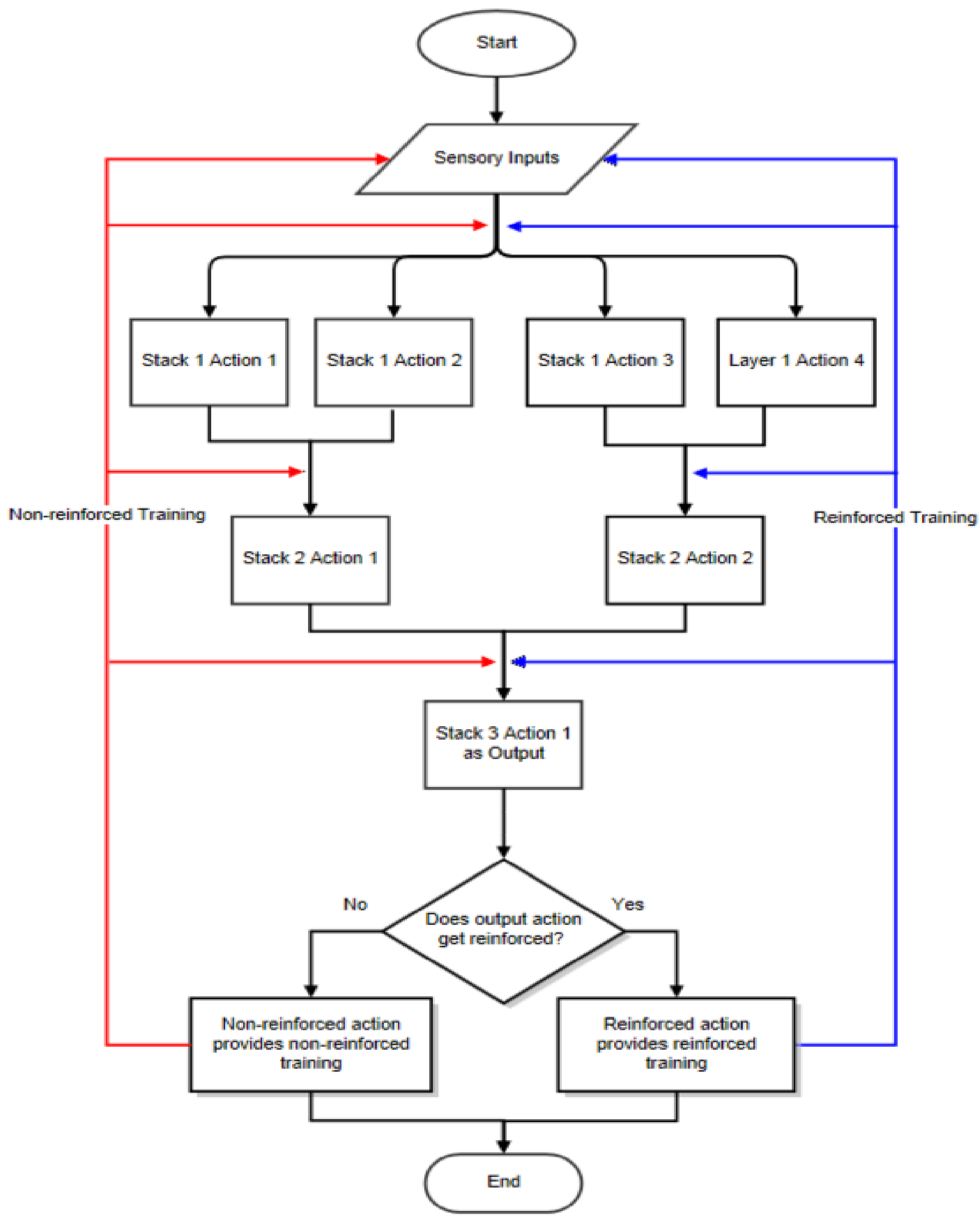


Figure 3: Real Stacked Neural Network with Intermediate Feedback

There are a number of reasons to have such stacked neural networks. One of the main reasons for this is that existing neural network models are not articulated enough to take on complex learning. The only ones that exist now have two layers, not counting the internal layers within a network. Networks that are more multilayered can address issues at each developmental stage and should therefore be able to model intelligence of a variety of animals and of humans. For example, at Order 4 Nominal (Commons & Pekker, 2008) a four layered stack could ***understand*** single words. It could respond to the ***meaning*** of such words.

Stacked neural networks use the Model of Hierarchical Complexity (Commons & 2008) to accomplish the following tasks: model human development and learning; reproduce the rich repertoire of behaviors exhibited by humans; allow computers to mimic higher order human cognitive processes and make sophisticated distinctions between stimuli; and allow computers to solve more complex problems.

The order of hierarchical complexity is measured by the number of recursions that the coordinating actions must perform on a set of primary elements. Recursion refers to the process by which the output of the lower-order actions forms the input of the higher-order actions. Each new, task-required action in the hierarchy is one order more complex than the task-required actions upon which it is built (Commons, et al., 1998).

Based on this brief description, how would one go about building such stacked neural networks cheaply and easily? Is there any software available that can do this? How much would it cost?

Please feel free to contact me if you think that it would be possible or easier to apply stacked neural network into a more practical field? Suggestions are welcome as well.

My Best,

Michael Lamport Commons, Ph.D.
Assistant Clinical Professor

Department of Psychiatry
Harvard Medical School
Beth Israel Deaconess Medical Center
234 Huron Avenue
Cambridge, MA 02138-1328

Telephone (617) 497-5270
Facsimile (617) 491-5270
Cellular (617) 320–0896
Commons@tiac.net
http://dareassociation.org/

Appendix

There are 16 orders in the model of hierarchical complexity. Only 5 orders are described here, but even constructing neural networks with these 5 would considerably increase the complexity of problems that could be addressed.

Table 1: 16 Stages and orders of hierarchical complexity

| Order | Name of Complexity |
|---|---|
| 0 | Calculatory |
| 1 | Sensory and Motor |
| 2 | Circular Sensory-Motor |
| 3 | Sensory-Motor |
| 4 | Nominal |
| 5 | Sentential |
| 6 | Preoperational |
| 7 | Primary |
| 8 | Concrete |
| 9 | Abstract |
| 10 | Formal |
| 11 | Systematic |
| 12 | Metasystematic |
| 13 | Paradigmatic |
| 14 | Crossparadigmatic |
| 15 | Meta-Crossparadigmatic |

In order 0: Machines can do simple arithmetic on 0s and 1s. There is no learning process in this order. Everything needs to be programed.

In Order 1: Salient stimuli are processed automatically. There is no coordination of stimuli or of actions with stimuli. The relationship between the detection of stimuli and the production of responses is not flexible and there is no understanding of concepts.

In Order 2: Actions are coordinated. A salient stimulus may change to a discriminative stimulus by coordinating response with sensory input. The reliably repeated discrimination is reinforced over time. The reinforcement leads to consequences and forms open-ended features of generalized stimuli. No understanding of concepts in this order.

In Order 3: Recognition of common perspective features in discriminative stimuli. Positive reinforced consequences of features lead to formation of observable concepts. Use common perceptual features in observable concepts to distinguish differences between different discriminative stimuli. Understand the difference between observable concepts but no understanding in relationship or coordination between different concepts is formed.

In Order 4: Use representation to represent observable concepts. Exp.: name, label or use arbitrary symbols for observable objects. No relationship or coordination between representations of concepts is formed.

References

Commons, M. L., Gane-McCalla, R., Barker, C. D., & Li, E. Y.  (In press).  The Model of Hierarchical Complexity as a measurement system. *Journal of Adult Development*

Commons, M. L., &  Pekker, A. (2008).  Presenting the formal theory of hierarchical complexity.  *World Futures: Journal of General Evolution  65*(1-3), 375-382.

Commons, M. L, & White, M. S. 2006. Intelligent control with hierarchical stacked neural networks. U.S. Patent 7,152,051, filed September 30, 2002, and issued December 19, 2006.

Zhang, J. (2008). Batch-to-batch optimal control of a batch polymerisation process based on stacked neural network models. *Chemical Engineering Science*, *63*(5), 1273-1281.
[I cannot read the original paper but based on its abstract, I think it is a layer neural network paper. However, it does mentioned that they use stacked neural network to build their work. ]

Zhang, J. J., Martin, E. B., Morris, A. J., & Kiparissides, C. C. (1997). Inferential estimation of polymer quality using stacked neural networks. *Computers & Chemical Engineering*

Luyue, X., Haitian, P., Mengfei, Z., Yijun, C., & Xiaofang, S. (2011). Estimation of Polypropylene Melt Index Through the Application of Stacked Neural Networks. *Advanced Materials Research, 187*411-415.

Furtuna, R., Piuleac, C., & Curteanu, S. (2012). Multi-objective Optimization of a Stacked Neural Network Used in Modeling of An electrolysis Process. *Environmental Engineering & Management Journal (EEMJ)*, *11*(9), 1657-1665.

Herrera, F., & Zhang, J. (2009). Optimal control of batch processes using particle swam optimisation with stacked neural network models. *Computers & Chemical Engineering*, *33*(10), 1593-1601.