OINNIONN - Outward Inward Neural Network and Inward Outward Neural Network Evolution

Aviv Segev, Rituparna Datta, Ryan Benton
Department of Computer Science
University of South Alabama
Mobile, Alabama, USA
{segev,rdatta,rbenton}@southalabama.edu

Dorothy Curtis
Computer Science and Artificial Intelligence Laboratory
MIT
Cambridge, Massachusetts, USA
dcurtis@csail.mit.edu

ABSTRACT

Neural networks perform well when they are built for a specific task and the set of inputs and the set of outputs are well defined. However, these results are very limited in scope, and communication between different neural networks to share knowledge that can lead to the performance of more general tasks is still inadequate. Communication between specialized neural networks is the goal of the present work. We utilize independent sets of neural networks trained for specific tasks, while transferring knowledge among the neural networks allows them to evolve chaining the input and output information. The idea is based on computer network architecture, which is a communication system that transfers data between components inside a computer or between computers. The idea can similarly allow each neural network to specialize in its own task while transferring and receiving information from other neural networks. This can allow different neural networks to be plugged in and knowledge transfer to evolve. It can also allow additional information to be requested, when the task at hand is difficult or hard to resolve.

1 OINNIONN METHOD

Neural networks and especially the deep learning approach have displayed successful results in areas such as image processing, medicine, and the stock market. The goal of the present work is to show that knowledge specialization in one domain learned by a neural network can be implemented in multiple domains. An evolving model of chaining information input and output of neural networks working independently allows the expansion of the use of neural networks. The neural networks learning process, which is today limited to one task in a specific area, can be extended. The information learned by one neural network can be shared and transferred to other neural networks in multiple domains. Similarly the input needed to perform a task can be extracted by any network which can be accessed at that time.

Currently the neuron doctrine views the brain as a network [8]. The brain can be viewed as a set of regions and a set of nodes while a relationship between two regions is mapped to a link between corresponding nodes. Categorizing each region is an issue which corresponds to building nodes in a network. Previous work in graph theory employed a clustering coefficient as a measure of the degree to which nodes in a graph tend to cluster together. In most real-world networks, and especially in social networks, nodes tend to form tightly knit groups marked by a relatively high density of links which tends to be greater than the average probability of a link randomly established between two nodes [5]. Methods have been developed for the analysis of homogeneous networks [9]. The analysis of heterogeneous networks is not simple, for links across entities can have several types. Another important issue is how to build a network for a given complex system.

One approach is using Web Services as an interface. There are existing protocols that deal with Web services such as: Open Service Interface Definitions (OSIDs) based on Service-Oriented Architecture (SOA) [1], Web services based on Web Service Definition Language (WSDL) [3], and Simple Object Access Protocol (SOAP) [4]. However, these services are not built for automatic registration of inputs and outputs of each system. Furthermore, the inputs and outputs have to be predefined by the user as opposed to learned by the communicating system. Most important is that these solutions are based on a predefined centralized registry that stores all the information in the network. In the field of automatic annotation of Web services, Patil et al. [6] present a combined approach towards automatic semantic annotation of Web services (e.g., string matcher, structural matcher, and synonym finder), which are combined using a simple aggregation function. Chabeb et al. [2] describe a technique for performing semantic annotation on Web services and integrating the results into WSDL.

Our Outward Inward Neural Network and Inward Outward Neural Network (OINNIONN) solution is based on an evolving decentralized learning system for identifying inputs and outputs of each neural network component. The proposed method includes a multi-task machine learning system by utilizing/merging goal-specific neural networks. The method develops a neural networks grouping system for decentralized learning. Each neural network focuses on one task. Each neural network is capable of sharing its knowledge with other networks and is able to request and receive knowledge from other neural networks. We use an adaptive neural network group membership selection to overcome the rigidness of conventional deep-learning.

The work involves designing a Plug-and-Play neural network group platform which provides a unique learning functionality through neural network groups matching individual networks. We plan to independently train task specific neural networks and then group them to implement a personalized and adaptive learning system. The solution is planned to provide short-term adaptation capability by re-using previously trained neural networks in new

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326791

neural network groups. We establish and adopt a communication protocol based on the existing Internet protocol.

The OINNIONN structure (Figure 1) allows different neural networks to form a hierarchical structure of sending and receiving information between one another. For example, a sound recognition neural network can parse a section of sound and send it to another layer which can identify whether it is voice, background sound, or foreground sound. After voice recognition, the next layer can transfer speech to text or identify gender. The last layer can identify the language spoken and possibly local dialect. Meanwhile, sound can be parsed into background and foreground noises, followed by neural networks specializing in identifying animal sounds or vehicle sounds. The outermost layer can possibly classify specific outdoor or indoor locations.

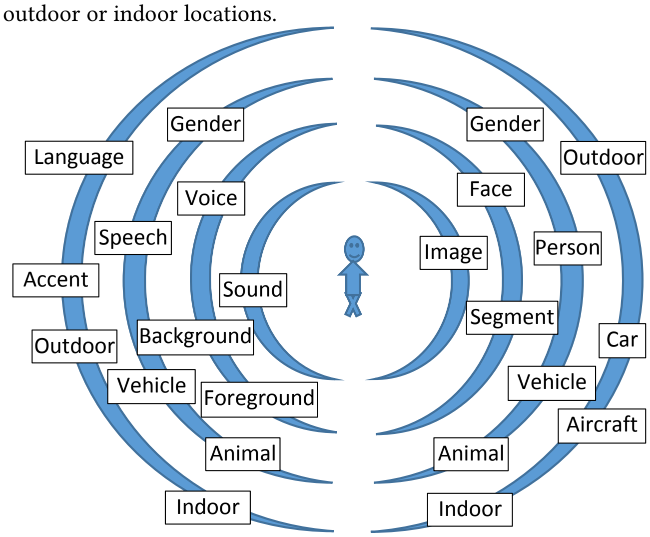

Figure 1: OINNIONN Structure of Neural Networks.

Meanwhile, the same OINNIONN structure allows a specific image to be selected from the scene. The image can be transferred to face recognition or alternatively image segmentation neural networks. If a face is recognized, the next layer can decide whether the face belongs to a person or animal. In parallel, gender recognition can be performed of facial features extracted. The gender recognition can be verified with complementary information received from voice gender recognition. Image segmentations which were not identified as faces can be sent to multiple neural networks identifying a person, vehicle, or animal in the image segment. Identification of a vehicle can be further sent to another layer identifying a specific type of car or aircraft. Simultaneously, the image can be identified as indoor or outdoor. The indoor or outdoor identification can again receive additional information from the voice processing communication between the neural network layers.

The layers of the neural networks do not need to be predefined and can be formed per scenario. Additional neural networks can be joined in each layer, additional layers can be formed, and multiple neural networks performing similar tasks, such as animal sound recognition or animal image recognition, can be integrated. For example, if two animal voice recognition networks are working in parallel, both can send the information to the outer layer without loss of information or interruption in the communication. Supporting scenario interpretations can increase the confidence of the scene viewed. Similarly, opposing scenario interpretations can lead to a search for additional neural networks which can serve as a tie breaker or supply a different perspective.

The communication network can be modeled by a graph in which each arc represents a path between two nodes that communicate. The arcs are associated with a cost C_{ij} (that represents the efforts in communications between nodes i and j-relevance metrics). If the arc does not exist, then the cost $C_{ij} = \infty$, else the cost has positive value. If the communication network has N nodes, the associated neural network has $N(N-1)$ neurons. Each neuron is identified by (r, c) where r is the row and c is the column in a neuron matrix. The relevance algorithm should find the optimal relevance between given source node S and destination node d. The output of a neuron is $Y_{rc} = 1$, if the link between nodes r and c is relevant, and 0 if not. The neural network model is to be based on Hopfield networks [7] that introduced the concept of neural network energy at a given time. A Hopfield neural network consists of a number of neurons completely connected with a dynamic functioning sequenced by a clock. The dynamic of such neural networks is described by the following equation: $\partial X_{rc}/\partial t = \sum_{i=0}^{N-1} \sum_{j=0, j\neq i}^{N-1} w_{rc,ic} Y_{ij} - X_{rc}/T + B_{rc}$ where X_{rc} represents the input for neuron (r, c), $w_{rc,ic}$ are the weights (connections) between neurons (r, c) and (i, j), T is the system clock, and B_{rc} is the bias for neuron (r, c). Considering the energy function E, the neural network dynamics can be given by: $\partial X_{rc}/\partial t = -X_{rc}/T - \partial E/\partial Y_{rc}$ The neural network energy can be computed as: $E = \frac{\alpha_1}{2} \sum_{r=0, r\neq s}^{N-1} \sum_{c=0, c\neq r, c\neq s}^{N-1} C_{rc} Y_{rc} + \frac{\alpha_2}{2} \sum_{r=0, r\neq s}^{N-1} \sum_{c=0, c\neq r, c\neq s}^{N-1} \lambda_{rc} Y_{rc} + \frac{\alpha_3}{2} \sum_{r=0, r\neq s}^{N-1} (\sum_{c=0, c\neq r, c\neq s}^{N-1} Y_{rc} - \sum_{c=0, c\neq r, c\neq s}^{N-1} Y_{cr})^2 + \frac{\alpha_4}{2} \sum_{r=0, r\neq s}^{N-1} \sum_{c=0, c\neq r, c\neq s}^{N-1} Y_{rc}(1 - Y_{rc}) + \frac{\alpha_5}{2}(1 - Y_{ds})$. Matrix λ_{rc} determines the network topology. $\lambda_{rc} = 0$ if there is a link between nodes r and c, and 1, otherwise.

The proposed solution allows each neural network to specialize in one domain such as vision or language processing. The neural network communication will make it easier to parse human activity into smaller tasks and integrate the results into more meaningful outcomes. The proposal can serve as a decentralized AI that can reuse existing solutions and enable humans to intervene at different levels of AI using a neural network that communicates via IP.

REFERENCES

[1] M. Bell. 2009. *SOA Modeling patterns for service-oriented discovery and analysis.* John Wiley & Sons.

[2] Y. Chabeb, S. Tata, and D. Belaïd. 2009. Toward an integrated ontology for web services. In *2009 Fourth International Conference on Internet and Web Applications and Services.* IEEE, 462–467.

[3] A. Chaturvedi. 2014. Subset WSDL to access subset service for analysis. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science.* 688–691.

[4] F. Hirsch, J. Kemp, and J. Ilkka. 2007. *Mobile web services: architecture and implementation.* John Wiley & Sons.

[5] P. W. Holland and S. Leinhardt. 1971. Transitivity in structural models of small groups. *Comparative Group Studies* 2, 2 (1971), 107–124.

[6] A. A Patil, A. Oundhakar, S, A. P. Sheth, and K. Verma. 2004. Meteor-s web service annotation framework. In *Proceedings of the 13th ACM International Conference on World Wide Web.* 553–562.

[7] S. Pierre, H. Said, and W.G. Probst. 2000. Routing in computer networks using artificial neural networks. *Artificial Intelligence in Engineering* 14 (2000), 295–305.

[8] M. Rubinov and O. Sporns. 2010. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* 52, 3 (2010), 1059–1069.

[9] A. Segev, D. Curtis, S. Jung, and S. Chae. 2016. Invisible brain: Knowledge in research works and neuron activity. *PLoS One* 11, 7 (2016).