NEUROGENIC DEEP LEARNING

Timothy J. Draelos, Nadine E. Miner, Jonathan A. Cox, Christopher C. Lamb, Conrad D. James, and James B. Aimone
Sandia National Laboratories
Albuquerque, NM 87185, USA
{tjdrael, nrminer, jacox, cclamb, cdjame, jbaimon}@sandia.gov

ABSTRACT

Deep neural networks (DNNs) have achieved remarkable success on complex data processing tasks. In contrast to biological neural systems, capable of learning continuously, DNNs have a limited ability to incorporate new information in a trained network. Therefore, methods for continuous learning are potentially highly impactful in enabling the application of DNNs to dynamic data sets. Inspired by adult neurogenesis in the hippocampus, we explore the potential for adding new nodes to layers of artificial neural networks to facilitate their acquisition of novel information while preserving previously trained data representations. Our results demonstrate that neurogenesis is well suited for addressing the stability-plasticity dilemma that has long challenged adaptive machine learning algorithms.

1 INTRODUCTION

Deep learning (DL) and other deep neural network (DNN) methods have proven successful in part due to their ability to utilize large volumes of data to progressively form sophisticated hierarchical abstractions of information (LeCun et al., 2015) (Schmidhuber, 2015). Training of DNNs can be very expensive, often requiring several days on large computing clusters (Le, 2013). Ideally a fully trained network will continue to prove useful for a long duration even if the application domain changes. DNNs are typically not robust to concept drift, where the data being processed changes gradually over time, nor are they well suited for transfer learning (TL), where a trained model is repurposed to operate in a different domain. Unlike the developing visual cortex, which is exposed to varying inputs over many years, the data used to train DNNs is typically from a limited set of data, diminishing the applicability of networks to encode information statistically distinct from the training set. One mechanism the brain uses in selective regions such as the hippocampus is the permissive birth of new neurons throughout one's lifetime, a process known as adult neurogenesis (Aimone et al., 2014) that provides it with a unique form of plasticity. Of particular relevance to machine learning, neurogenesis (NG) has been proposed to enable the acquisition of novel information while minimizing the potential disruption of previously stored information (Aimone et al., 2011).

1.1 NEUROGENESIS ALGORITHM AS AN EFFICIENT METHOD FOR ADAPTING DNNS

The value of a model to continuously adapt to changing data is challenging to quantify. Here, we quantify the value of a machine learning algorithm at a given time as $U = B - C_M/\tau - C_P$, where the utility, U, of an algorithm is considered as a tradeoff between the benefit, B, that the computation provides the user, the costs of the algorithm generation or the model itself, C_M, and the associated run-time costs, C_P, of that computation. C_P typically consists of the time and physical energy and space required for the computation to be performed. For machine learning applications, we must consider the lifetime, τ, of an algorithm for which it is appropriate to amortize a model's build costs. In algorithm design, it is desirable to minimize both of the cost terms; however the dominant cost will differ depending on the extent to which the real-world data changes. Consider a DNN with N neurons and on the order of N^2 synapses. In this example, the cost of building the model, C_M, will scale as $O(N^4)$ due to using N^2 training samples N^2 times during training of a well-regularized, appropriately fit model. As a result, C_M will dominate the algorithm's cost unless the lifetime of the model, τ, can offset the polynomial difference between C_M and C_P. This description illustrates

the need to extend the lifetime of a model (e.g., via neurogenesis), and to do so in an inexpensive manner that minimizes the amount of data required to adapt the model for future use.

2 NEUROGENIC DEEP LEARNING ALGORITHM

At each encode/decode pair of layers of a trained autoencoder (AE), reconstruction error (RE) is a readily available gauge of the quality of the network's ability to represent data and can therefore guide NG. We compute RE at internal layers by encoding the input, I, through the specified number of encode layers and propagating through the corresponding decode layers to get its reconstruction. An AE parameterized with weights W, biases b, and activation function s can be described from input, x, to output as N encode layers followed by N decode layers.

Encoder: $f_{\Theta_N}(f_{\Theta_{N-1}}(\cdots(f_{\Theta_2}(f_{\Theta_1}(x)))\cdots))$, where $y = f_\Theta(x) = s(Wx + b)$

Decoder: $g_{\Theta'_N}(g_{\Theta'_{N-1}}(\cdots(g_{\Theta'_2}(g_{\Theta'_1}(y)))\cdots))$, where $g_{\Theta'}(y) = s(W'y + b')$

Then, RE at layer L is $RE_L(x) = (x - g_{\Theta'_N}(\cdots(g_{\Theta'_{N-L}}(f_{\Theta_L}(\cdots(f_{\Theta_1}(x))\cdots)))\cdots))^2$.

2.1 NEUROGENESIS

Neurogenic Deep Learning (NDL) begins by using a single hidden layer (SHL) AE to reconstruct all samples of a new class at layer L in a similar way to stacked AE pretraining, regardless of how deep into the network a layer is. If enough samples have RE_L above a layer-wise threshold T_L, then a new node is added to layer L and the expanded SHL AE is trained, where only the weights connected to the newly added node are allowed to be updated at the full learning rate (η). Encoder weights connected to old nodes are not allowed to change at all and decoder weights from old nodes are allowed to change at a lower learning rate ($\eta/100$ in our experiments) to maintain stability in the old weights. This step relates to the notion of plasticity in biological NG. After training of the new node is complete, a stabilization step takes place, where the entire layer is trained using samples from all classes seen by the network. Samples from old classes are generated via hippocampus-inspired "intrinsic replay" (IR) by retrieving a high-level representation through sampling from the multivariate Normal and Cholesky decomposition of the top layer of the full encoder network and then leveraging the full decoder to reconstruct new data points from that previously trained class. After again calculating RE_L on samples from the new class, additional nodes are added until either 1) RE_L for enough samples falls below T_L, 2) a user-specified maximum number of new nodes are reached for the current layer, or 3) the number of samples with RE_L above T_L stabilizes. Once NG is complete for the first layer, weights connecting to the next layer are trained using samples from all classes, new and old. This process repeats for each of the succeeding layers of the encoding network using outputs from the previous layer. After NG, the new AE should be capable of reconstructing images from the new class in addition to the current previous classes.

3 RESULTS

We evaluate NG on a deep AE on a specific example of the MNIST handwritten digit data set (LeCun et al., 1998), whereby a network is initially trained with two digits (1, 7) that are not statistically representative of the other digits. Our results demonstrate that in such a situation, NG along with IR enables the training of new digits while minimally impairing original representations. We use an 8-layer AE inspired by Hintons network on MNIST (Hinton & Salakhutdinov, 2006), but reduced to 784-200-100-75-20-75-100-200-784 since only a subset of digits (1, 7) are used for training. We focus initial training on digits 1 and 7 as they may represent the smallest set of features in any pair of digits, and simulate learning a new task by progressively expanding the number of encountered classes through adding samples from the remaining digits in sequence one at a time.

Results of NDL experiments show that an established network trained on just digits 1 and 7 can be enlarged through NG to represent new digits as guided by RE at each encode layer of an AE. We compared a network created with NDL and IR (NG+IR) to three control networks: Control 1 (TL+IR) - an AE trained first on the subset digits 1 and 7 and then retrained with one new single digit at a time with standard TL, using IR to generate samples of previously trained classes throughout the experiment; Control 2 (NG) - TL on the original 1,7 network using NDL, but not using IR; and

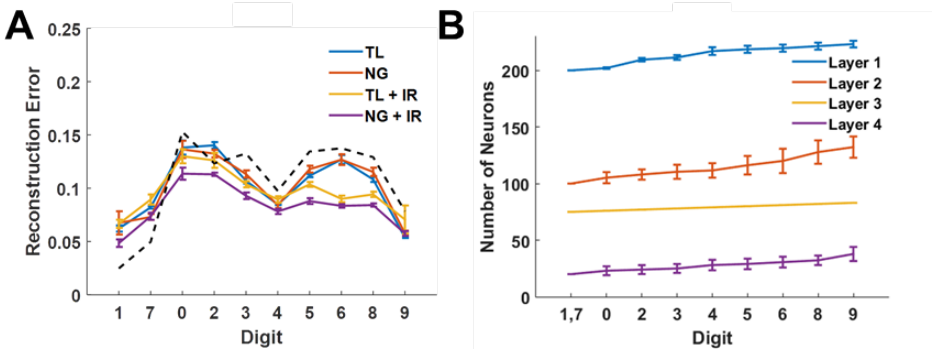

Figure 1: Reconstruction errors (scaled by the number of samples and dimension of each sample) of trained MNIST autoencoders after exposure to all 10 digits. (A) Full network reconstructions of all networks after progressive training through all digits; the dotted line is the original 1,7-trained network reconstruction of all digits. (B) Neurogenesis contribution to network size in NG-IR networks.

Control 3 (TL) - an AE the same size as the enlarged Control 2 network trained first on digits 1 and 7, and then retrained without IR on a new single digit at a time. Notably, NG+IR outperforms (has lower RE) straight TL not only overall, but in both the ability to represent the new data as well as preserving the ability to represent previously trained digits (see Figure 1A, where the TL+IR process performs well on new digits, but poorly on retaining the original old digits, whereas the NG+IR process does well on all digits). While getting a trained network to learn new information is not particularly challenging, getting it to preserve old information can be quite difficult. Note that the final DNN size of the ultimate network is unknown prior to NG. The network size is increased based on the RE magnitude when the network is exposed to new information, so there is particular possible value in using this method to determine an effective DNN size. Figure 1B shows how the DNN grows as new classes are presented during NG. The network gains more representational capacity as new classes are learned.

4 Conclusions

We presented a new adaptive algorithm using the concept of neurogenesis to extend the learning of an established DNN when presented with new classes of data. The algorithm adds new neurons at each layer when the new data samples are not accurately represented by the network as determined by a high RE. We anticipate that there are several significant advantages of a neurogenesis-like methods for adapting existing networks to incorporate new data, particularly given suitable intrinsic replay capabilities. The first relates to the costs of DL in application domains. The ability to adapt to new information can extend the useful lifetime of a model in real-world situations, possibly by substantial amounts. As a result, online adaptation can potentially make DL cost effective for domains with significant concept drift. The second advantage concerns the online learning nature of the NDL algorithm. The ability to train a large network without having to maintain a growing repository of data can be highly valuable, particularly in cases where the bulk storage of data is not permitted due to costs or other restrictions. While much of the DL community has focused extensively on cases where there is extensive unlabeled training data, this technique can provide a solution for situations where training data at any time is limited and new data is expected to arrive continuously. Furthermore, here we have considered a very stark change in the data landscape, with the network exposed exclusively to novel classes. In real-world applications, one may imagine novel information to be encountered more gradually. This slower drift would likely require neurogenesis less often, but it would be equally useful when it is needed.

Acknowledgments

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND2016-1422 C.

REFERENCES

J. B. Aimone, W. Deng, and F. H. Gage. Resolving new memories: a critical look at the dentate gyrus, adult neurogenesis, and pattern separation. *Neuron*, 70(4):589–96, 2011. ISSN 1097-4199 (Electronic) 0896-6273 (Linking). doi: 10.1016/j.neuron.2011.05.010. URL http://www.ncbi.nlm.nih.gov/pubmed/21609818.

J. B. Aimone, Y. Li, S. W. Lee, G. D. Clemenson, W. Deng, and F. H. Gage. Regulation and function of adult neurogenesis: from genes to cognition. *Physiol Rev*, 94(4):991–1026, 2014. ISSN 1522-1210 (Electronic) 0031-9333 (Linking). doi: 10.1152/physrev.00004.2014. URL http://www.ncbi.nlm.nih.gov/pubmed/25287858.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Quoc V Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8595–8598. IEEE, 2013.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Jurgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. ISSN 0893-6080.