Neural network generation for estimation of tissue optical properties

Gil, Eddie, Hokr, Brett, Bixler, Joel

Eddie Gil, Brett H. Hokr, Joel N. Bixler, "Neural network generation for estimation of tissue optical properties," Proc. SPIE 11238, Optical Interactions with Tissue and Cells XXXI, 1123809 (20 February 2020); doi: 10.1117/12.2546068

Event: SPIE BiOS, 2020, San Francisco, California, United States

Neural network generation for estimation of tissue optical properties

Eddie Gil[1,2], Brett H. Hokr[3], and Joel N. Bixler[4]

[1]Department of Biomedical Engineering, Texas A&M University, College Station, TX 77843; [2]SAIC Corporation, JBSA Fort Sam Houston, TX 78234; [3]Radiance Technologies Inc., Huntsville, AL 35805; [4] Air Force Research Laboratory, JBSA Fort Sam Houston, TX 78234

Abstract

Monte Carlo Simulations (MCSs) allow for the estimation of photon propagation through media given knowledge of the geometry and optical properties. Previous research has demonstrated that the inverse of this problem may be solved as well, where neural networks trained on photon distributions can be used to estimate refractive index, scattering and absorption coefficients. To extend this work, time-dependent MCSs are used to generate data sets of photon propagation through various media. These simulations were treated as stacks of 2D images in time and used to train convolutional networks to estimate tissue parameters. To find potential features that drive network performance on this task, networks were randomly generated. Generated networks were then trained. The networks were validated using 4-fold cross validation. The consistently performing top 10 networks typically had an emphasis on convolutional chains and convolutional chains ending in max pooling.

1. Introduction

Monte Carlo simulations can be used to model the distribution of photons in a medium, or backscattered to a detector over time, and have become a state-of-the-art technique for doing so [1] [2] [3]. Such models can be used to offer valuable insights into heating and subsequent damage of tissues [4]. By the same token, changes in the status of some medium or a tissue can result in a change in optical properties. On a large scale, by creating a map of these optical properties over a region, tissue damage can be imaged. Then, by extracting features from or by passing such images to an intelligent network, the status of the tissue can be further assessed. This would be highly valuable to medical diagnosis. To accomplish this, a model would need to predict optical properties, given a backscattering sequence. Creating such an inverse model has proven to be challenging due to run time and noise in solutions, along with biological variability found among populations [5]. The problem increases in difficulty as more material layers are modeled, and objects of varying geometry are placed within the scope of the material, i.e., biological materials with capillaries, hair follicles, etc. However, neural networks and data driven approaches may mitigate these issues. The Air Force Research Lab (AFRL) [5] has previously shown that neural networks can be trained on simulated data to predict optical properties of a slab of material, given a vector of moments taken from an aggregate of detected photons. This technique is extremely advantageous since collecting a robust data set of samples from physical material with various embedded obstacles of known properties can be a lengthy and challenging process. This work extends upon that line of thought, by training convolutional neural networks directly on the backscattering sequences to estimate a matrix containing vectors of optical parameters at varying depths. This allows for more degrees of freedom in the input space and may allow more sensitivity for samples that have many layers and some embedded obstacles.

The network architecture that would be best suited to address this problem is not currently known. Recently, neural architecture searches were performed using an RNN to generate descriptions of models and learn hyper parameters for each layer in a network, along with connectivity between operational elements [6]. It has also been shown that it is possible to make powerful networks that utilize only convolutional and summing units [7] prior to the classification stage. This past work has resulted in interesting subunits to use for networks. However, the works focus on automated proposal of networks leaving the details to the machine. Thus, there is space for insights about what structural choices to make when designing a network. This is valuable because this information can be used to constrain future architecture searches, meaning shorter search times. In this study, not only do we show that randomly generated networks can be used to address the inverse Monte Carlo problem for backscattering sequences, we show what structural features provide good results in these models.

Optical Interactions with Tissue and Cells XXXI, edited by Bennett L. Ibey, Norbert Linz, Proc. of SPIE
Vol. 11238, 1123809 · © 2020 SPIE · CCC code: 1605-7422/20/$21 · doi: 10.1117/12.2546068

2. Materials and Methods

A forward Monte Carlo simulation was used to generate sequences of backscattered photons, as they would appear at a detector. Neural network architectures were randomly generated such that the number and type of nodes used in the models followed distinct probability distributions. Connection density was also varied. Each network was trained to predict a set of optical parameters that describe a multilayered medium, given a detection sequence. Once the networks were trained, the models were ranked and features about the network structure were extracted to explain what networks perform better over others.

2.1 Monte Carlo Simulation Inverse Problem

The Monte Carlo simulation assumes that photons will be sent into a medium. The number of backscattered photons detected is dependent on the characteristics of the material, including the absorption and reduced scattering coefficients, and refractive index. Thus, the forward problem is to estimate backscattering given material characteristics. In the inverse problem, optical properties of the medium are predicted given detected backscattering. The model described in [1] simulating transport in multilayered media was used without nonlinear effects. Each layer of the sample has its own optical parameters. Material samples simulated in this study were constrained to have between two and 10 uniform layers. Each sample was placed at a given depth from the surface between 1-50 mm, a refractive index between 1.3 and 1.6, an anisotropy coefficient between 0.5 and 0.95, a scattering coefficient between 0.1 and 100, and an absorption coefficient of 0.01 and 10. All labels were then normalized so that they ranged between 0 and 1, since the final layer in the networks makes use of sigmoid activation. These constraints were used to simulate 4,500 photon detection sequences. The resulting detection sequences had the shape 64x64x64. The x, y plane spanned a square region of 100 mm^2. The 64-time bins (3rd dimension) covered 2 ns with a step size of 10 ps. A total 10^5 photons were used for each simulation. The label for each sequence was a 10x6 matrix, where each row of the matrix corresponds to a layer in the simulated sample, while the columns map to the six optical properties. The first two columns represented the number of regions, and the depth of each layer in the material. The last four columns contained the optical properties. For materials that had less than 10 layers, spots that would be empty in the matrix were replaced with zeros.

2.2 Neural Network Generation and Training

Network Generation: Networks were generated by first setting a fixed set of possible nodes to sample from. The set consisted of convolutional, max pooling, concatenation, and summing layers. All activations used rectified linear units. A stack of fully connected layers (2,048 neurons, 1,024 neurons, 10 neurons) was placed at the end of the generated networks to perform the final label prediction. The final layer used sigmoid activation. To create groups of similar networks, the node set for each group of networks was constrained using a probability distribution. A total of 10 probability distributions were used. The first five distributions were constructed by ramping the percentage of convolutional layers from 40% to 90%, and enforcing a uniform distribution on the remaining node types. All other distributions were generated by sampling numbers from a uniform distribution on (0,1], and normalizing the distribution by the sum of the samples. In this work, nine node networks were generated. To accomplish this, 9x9 graphs were used to describe connectivity between nodes. The graphs were made by thresholding 9x9 matrices of uniformly distributed random numbers on (0,1] so that the resulting matrix was binary. The threshold was ramped so that in each group the connection density went from 5% to 95%. The row labels in the matrix set the input node, while the columns set the output node. Thus, the entry {0,3} represents the connection from node zero to node three. The graph matrices were made to be upper right triangular to prevent backflows and self-connection in the networks. For summing nodes and concatenation nodes, number of input connections was set to be two or greater. Since the size of the input data was constrained to 64x64x64, the number of max pooling layers was restricted to four (so the minimum size representation was 4x4x64). A constraint was also placed so that the first two nodes were restricted to either convolutional or max pooling layers. Once the network graphs were constrained, they were used to generate models for training.

Training: The networks were trained using mean square error between the prediction matrices and the label matrices as the loss. Stochastic gradient descent with a learning rate of 0.001 was used. Each network was trained for 100 epochs. The data was split so that 75% of the 4,500 backscattering simulations were used for training and 25% were used for testing. Four-Fold cross validation was performed, using the untrained version of each of the 50 networks as starting point every time a new fold was used.

3. Results

Once the training was completed for each fold, the average training/testing loss and training/testing accuracy signals were computed for the 50 networks. The minimum achieved in the average testing loss signal was found for each network. Figure 1 summarizes these results. Each color group represents one node type distribution. The density of connections within each color group increases from 5% to 95% from left to right in the group. Most of the networks achieve near 0.1 MSE. Certain networks clearly perform worse than others with little dependence on connection density in this case. The distribution groups were ranked in ascending order (lowest MSE wins) by computing the mean of the minimal MSE for each group.

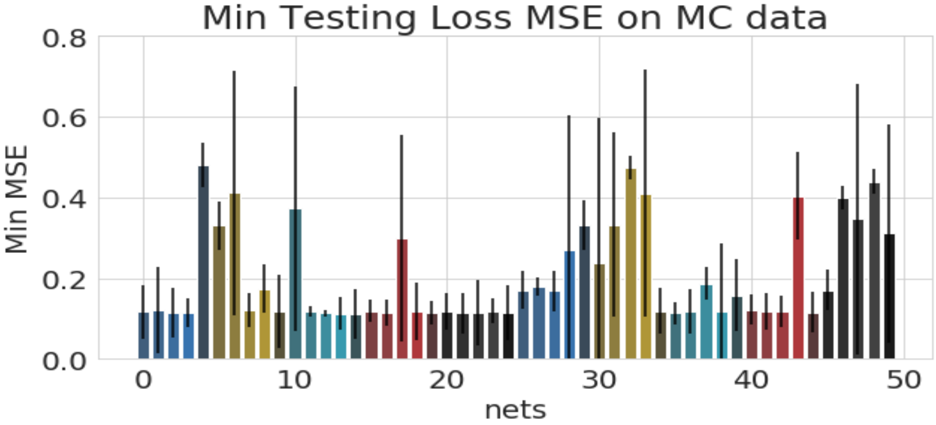

Figure 1: Color coded bar chart displaying the minimum MSE for each trained network. The connection density in each group ramps from 5% to 95% from left to right. At network 26, the colors repeat but the distributions for networks 26-50 are different from those in 0-26. The min MSE is the average for that network given the 4 training cross folds. The error bars are the standard deviation for that network given the cross folds.

Figure 2 shows the characteristic node distributions for the trained networks. The top 3 distributions are marked with darker shading. Each distribution consisted of concatenation summing, max pooling, and

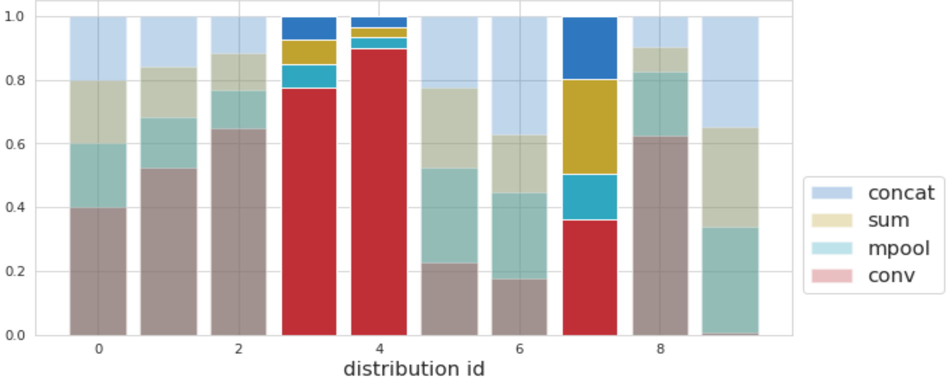

Figure 2: Characteristic distributions constraining node type generation in network graphs. The top three networks are less transparent than the other distributions. The top distributions have a high number of convolutional layers and summing layers.

convolutional units. The first five distributions were set to increase the number of convolutional layers then divide the remaining probability amongst the other node types. The final five were randomly generated following a uniform distribution.

The occurrence of each node type in a given position in networks was found independent of the generating distributions. Figure 3 shows the number of times a particular node type occurred in the top 10 networks, and the number of times a node occurred in the bottom 40 networks. The first two layers were constrained to be maxpooling or convolutional. The bottom 40 networks have a larger protion of concatenation layers than the top ten networks. The top ten networks mainly use convolutional layers, with maxpooling more common towards the end of the network.

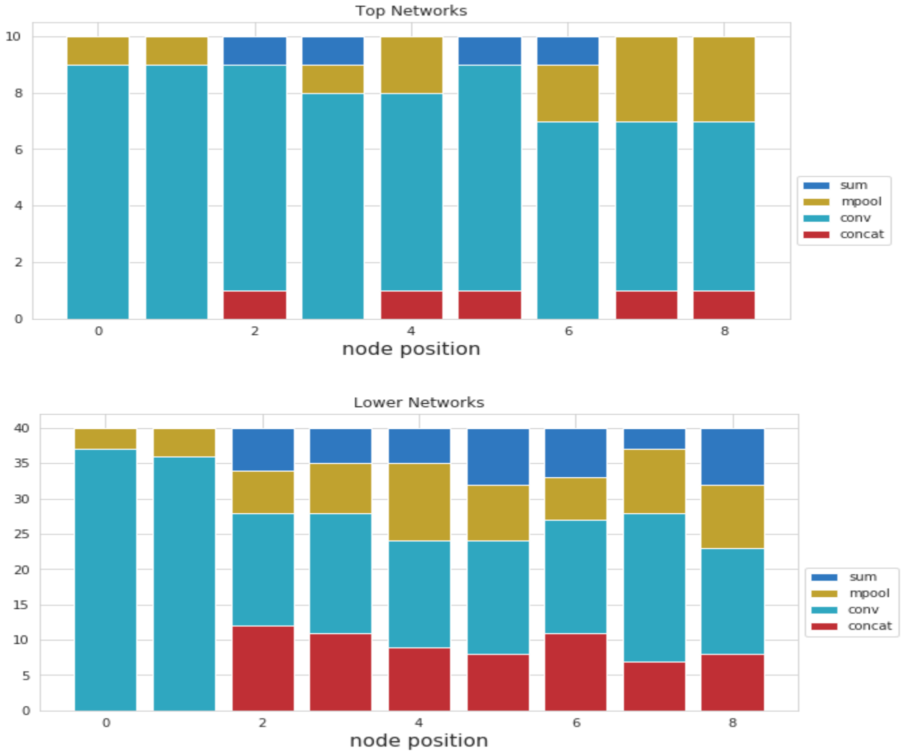

Figure 3: Occurrence of nodes in a given position of the top 10 and bottom 40 networks. The top networks use more convolutional layers than the bottom networks. Top networks use far fewer concatenation units.

Finally, the percentage of three long chains in the top and bottom models was computed and are shown in Figure 4. In the top models, 50% of the three long chains in the network were from convolutional → convolution → convolution paths. The next most important path was convolition → convolution → max pooling, accounting for 10% of all connections. The rest of the connection types were minimal. In the bottom 40 networks, three long covolution chains were also the most significant, but only account for 17.5% of all network connections. Three long convolution chains ending in maxpooling rank fifth in the bottom network connection types. The connection types in the bottom networks are less skewed than those in the top networks if the three long convolutions are left out.

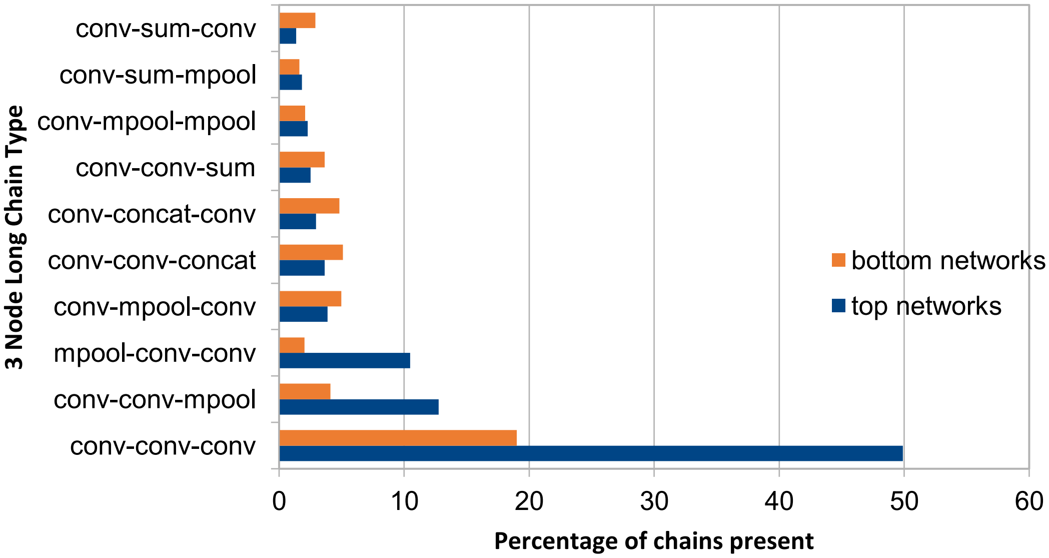

Figure 4: Percentage of three long chains in the top and bottom networks. Convolutional chains are most important followed by chains that end in max pooling.

4. Discussion

The cross-validation results in Figure 1, show that only certain randomly generated networks can achieve a consistent MSE between their predictions and the true tissue parameters output by the Monte Carlo Simulation. The best MSE these networks can achieve is about 0.1. The most successful networks were generated from distributions using a high skew towards convolutional layers. The best performing networks use little to no concatenation layers, and make use of max pooling layers more often in the end of the network. The most important three long chain in the top performing networks are convolution chains, followed by convolution chains ending in max pooling. All other connection types account for only small percent of the overall connections.

While cross validation is an adequate technique for validating the performance of the models outlined in this study, we would like to collect backscattering data from physical samples to further validate the model performances in the future. Another issue is that replacing empty cells in the labels with zeros may make the data easier to learn due to data biasing. Randomly generating networks and training each one requires a significant amount of training time. To mitigate this, we intend to examine features that describe neural output in order to predict which architectures will outperform others before even training the models. Additionally, it is unclear whether these structural features are generalizable across other sets of data, so testing on benchmarks is needed. Lastly, we plan to explore simulating multilayer materials with various embedded obstacles to scattering as input to these networks.

References

[1] B. H. Hokr, V. V. Yakovlev and M. O. Scully, "Efficient Time-Dependent Monte Carlo Simulations of Stimulated Raman Scattering in a Turbid Medium," *ACS Photonics,* vol. 1, no. 12, pp. 1322-1329, 17 12 2014.

[2] L. Wang, S. L. Jacquesa and L. Zhengb, "CML-Monte Carlo modeling of light transport in multi-la.yered. tissues," 1995.

[3] B. H. Hokr, J. N. Bixler, G. Elpers, B. Zollars, R. J. Thomas, V. V. Yakovlev and M. O. Scully, "Modeling focusing Gaussian beams in a turbid medium with Monte Carlo simulations," *Optics Express,* vol. 23, no. 7, p. 8699, 6 4 2015.

[4] A. Majdabadi and M. Abazari, "Study of Interaction of Laser with Tissue Using Monte Carlo Method for 1064nm Neodymium-Doped Yttrium Aluminium Garnet (Nd:YAG) Laser," 2015.

[5] B.H. Hokr ,. J.N. Bixler, "Machine Learning Estimations of Tissue Optical Properties", Scientific Reports (under review).

[6] B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," 4 11 2016.

[7] S. Xie, A. Kirillov, R. Girshick and K. He, "Exploring Randomly Wired Neural Networks for Image Recognition," 2 4 2019.