Intelligent Imaging: Anatomy of Machine Learning and Deep Learning

Geoff Currie

School of Dentistry and Health Sciences, Charles Sturt University, Wagga Wagga, Australia

The emergence of artificial intelligence (AI) in nuclear medicine and radiology has been accompanied by AI commentators and experts predicting that AI would make radiologists, in particular, extinct. More realistic perspectives suggest significant changes will occur in medical practice. There is no escaping the disruptive technology associated with AI, neural networks, and deep learning, the most significant perhaps since the early days of Roentgen, Becquerel, and Curie. AI is an omen, but it need not be foreshadowing a negative event; rather, it is heralding great opportunity. The key to sustainability lies not in resisting AI but in having a deep understanding and exploiting the capabilities of AI in nuclear medicine while mastering those capabilities unique to the human resources.

Key Words: nuclear medicine; artificial neural network; deep learning; convolutional neural network; artificial intelligence

Artificial intelligence (AI) is a general term used to describe algorithms designed for problem solving and reasoning. Applications in nuclear medicine and radiology have been widely documented. A subset of AI is associated with neural networks. In medical imaging, a neural network is an image analysis algorithm composed of layers of connected nodes (*1*). The nodes can be in the order of hundreds to millions and simulate the neuronal connections of the human brain (*2*). Nodes receive information from other nodes or patterns of nodes. Communication from one node to other nodes occurs when a threshold is exceeded, and the outputs from those nodes are weighted (Fig. 1). The basic principle is

to maximize the number of correct answers by comparing artificial neural network (ANN) estimates with a reference (grounded truth) and then adjusting the weightings on each node based on the error (*2,3*). There may be hundreds or thousands of iterations required to make the adjustments during the training phase of developing an ANN. Clearly, the more data that are used to train the ANN, the greater the accuracy of the inference phase. Through each iteration and subsequent adjustment of the nodes, a mathematical solution converges on a more accurate solution in a similar manner to what we might think about iterative reconstruction of SPECT and PET data.

An ANN typically has 3 phases: the training phase, in which the ANN learns; the validation phase, in which the learning of the ANN is evaluated against a second dataset; and the inference or application phase, in which the ANN is applied to actual cases. The training phase follows a diminishing-return principle, eventually reaching a point at which additional iterations do not improve the results or the improvement is negligible (Fig. 1) (*2*). The training phase can be supervised (grounded truth, or human-interpreted training data) or unsupervised (no grounded truth, or learning based on pattern recognition) (*4*). After the training phase, a second dataset can be used to test the accuracy of inferences of the ANN to validate the algorithm before it is used in clinical and research applications (Fig. 1). The role of big data in medical imaging is to provide a reliable and large training database for machine-learning (ML), representation-learning, and deep-learning (DL) algorithms to produce accurate outcomes (*1*). There are, however, potential clinical and research roles for ANNs in parallel to conventional statistical analysis in small data to identify key inputs (features) or combinations of inputs not gleaned from multivariate analysis.

In the testing and validation phase, a second smaller database of features or images is used for the ANN to evaluate, and those inferences are compared with a grounded truth (Fig. 2). This phase predicts the accuracy of the ANN when used clinically or in research (*5,6*). That degree of accuracy
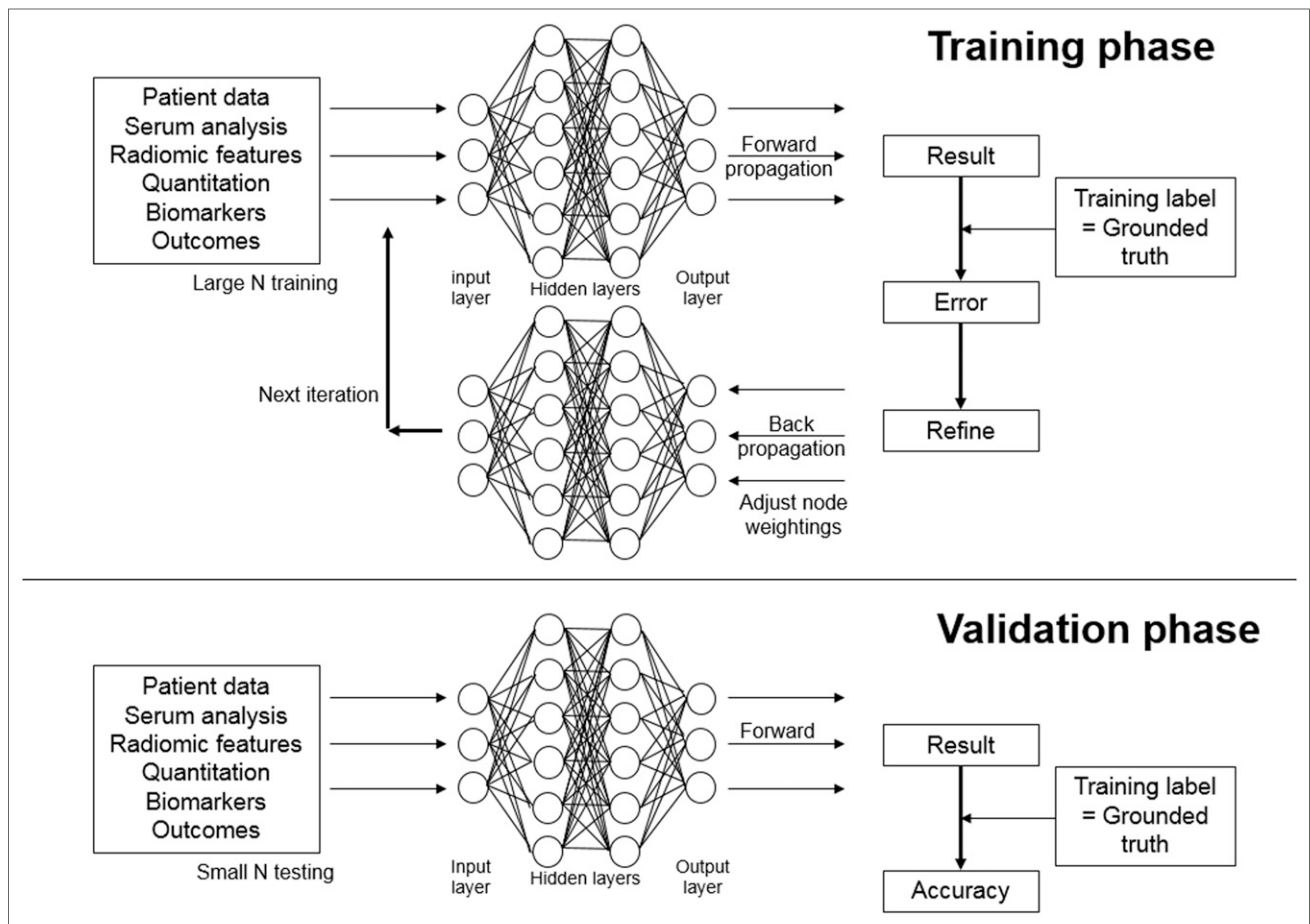
FIGURE 1. The training phase of a neural network using extracted features as inputs. The grounded truth defines the ANN as a supervised ANN. This ANN structure is also what might be used as an analysis tool in parallel with traditional statistical analysis—importing data from a spreadsheet, for example. This example, which depicts all nodes as being connected to all others in adjacent layers, represents "fully connected layer," which is more typical of CNNs. The validation phase evaluates trained ANN against a new database of known cases to determine accuracy. (Adapted from (1).)

can then be expected in the application phase, in which the neural network makes inferences about images without a grounded truth (Supplemental Fig. 1; supplemental materials are available at http://jnmt.snmjournals.org). An ANN would have data or features entered into the input layer of the algorithm as depicted in Figure 1.

DL associated with convolutional neural networks (CNNs) has a higher-order functionality in which the neural network itself is trained to identify and extract features from images (Fig. 2) (7). The term *convolution* means the mathematical combination of 2 functions to generate a third function. As depicted in Figure 2, the input has several image dimensions (*x, y,* and *z*) and several images (e.g., SPECT slices). The image itself has specific features identified and extracted into a convolution feature map (7). A kernel or rectified linear unit is an activation filter through which convolution data are pooled (7). Multiple convolution, kernel, and pooling iterations may occur before the pooled features are flattened for entry into the input layer of the fully connected neural network (7). The depth of the CNN gives rise to the expression *deep learning.*

ANATOMY OF ML

ML algorithms, including ANNs, have 3 key components (6,8). The first is the mathematical model that is used to describe or explain the relationships within the data; specifically, the relationships between inputs (features) and outputs (outcomes). The second component is the cost function, which is an evaluation of the accuracy of the mathematical model, or how well the model predicts an outcome. The error between the predicted and expected outcomes (grounded truth) is the loss function. The third component, the data, is necessary but varies among the training phase, the validation phase, and then the inference phase. Big data from multicenter trials may be used for the training phase, and a smaller number of cases with known outcomes can be used for the validation phase. Typically, the same database is used and randomly split (e.g., 80:20) to produce a large training set and a smaller but statistically significant validation set. A separate population of patients can then be used as the inference phase for further research (deeper validation with external validity) or for clinical decision making.
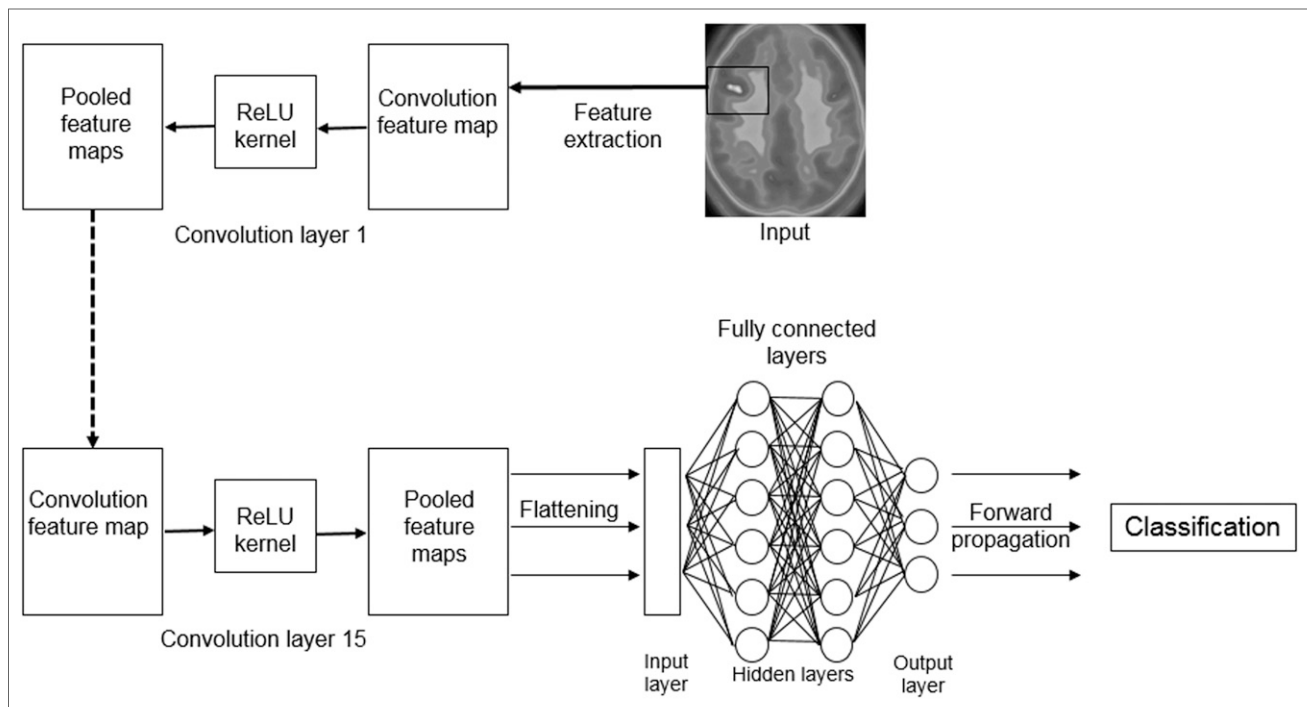
FIGURE 2. Basic structure of a CNN, with the network extracting radiomic features, producing convolution function, pooling data through rectified linear unit (ReLU) kernel, and flattening pooled feature map for input into fully connected hidden layers of the neural network. (Reprinted from (*1*).)

There are a variety of ML algorithms available, and the preferred approach (e.g., CNN vs. ML) will depend on the type of data and the purpose. For simplicity, the following discussion will assume a binary output (e.g., cardiac event or no cardiac event) and rich input data of extracted features in a model that resembles Figure 1. One should keep in mind that this is a model meant for aiding the understanding of nomenclature and processes rather than being a fit for all ANNs, just as human anatomy has normal variants and differs among mammals despite having some commonality.

Consider several potential input features (e.g., 4) in 1,000 patients in a database. A single binary output might be a cardiac event during the follow-up period or no cardiac event in the follow-up period. The ANN architecture would include 4 scaling-layer inputs and several hidden (perception) layers (let us assume 4) with multiple nodes in each hidden layer (perhaps 4, 8, 8, and 3) (Fig. 3). The scaling layer is to ensure all inputs are within the prescribed range and contain input statistics (e.g., mean, SD, minimum, and maximum). Each node (perceptron) in the perception layers receives numeric inputs, which have weightings and are combined with a bias to produce a single net input value (*8,9*):

$$\text{Combination} = \text{bias} + \sum \text{weights} \cdot \text{inputs}.$$

An activation function defines the output of the perceptron (liner, logistic, rectified linear, hyperbolic tangent) (*5,7,9,10*). In the case of a linear activation function, the activation is

equal to the net input value (*5,7,9,10*). The more common logistic activation function is a sigmoid function:

$$\text{Activation} = \frac{1}{1 + e^{-\text{combination}}}.$$

The ANN works toward a probabilistic layer (e.g., binary, continuous, competitive, or softmax) or probabilistic output function. Between the last perception layer and the probabilistic layer, an unscaling layer is needed to convert outputs to the original units (*8–10*).

The architecture needs to be trained and optimized. The loss index is a tool used to measure the error associated with the algorithm executing its task (error term) and to measure the quality of the data the ANN is learning (regularization term) (*5,7,8,10,11*). The error term can be measured in numerous ways, including mean squared error, normalized squared error, weighted squared error, or Minkowski error (*9*). The weighted squared error could be used to determine the loss index, especially when there is an imbalance between positive and negative outputs (e.g., a ratio of 1.2:1 against grounded truth). Regularization relates to the size of changes in outputs in response to changes in inputs, with small changes producing small changes being considered regular. The regularization term is summed with the error term, which will reduce weights and biases to produce a smoother output (*9,10*).

Optimization is an adjustment to the weightings on individual nodes (perceptrons) in order to minimize (optimize) the loss index (*5,8–10*). This optimization is achieved
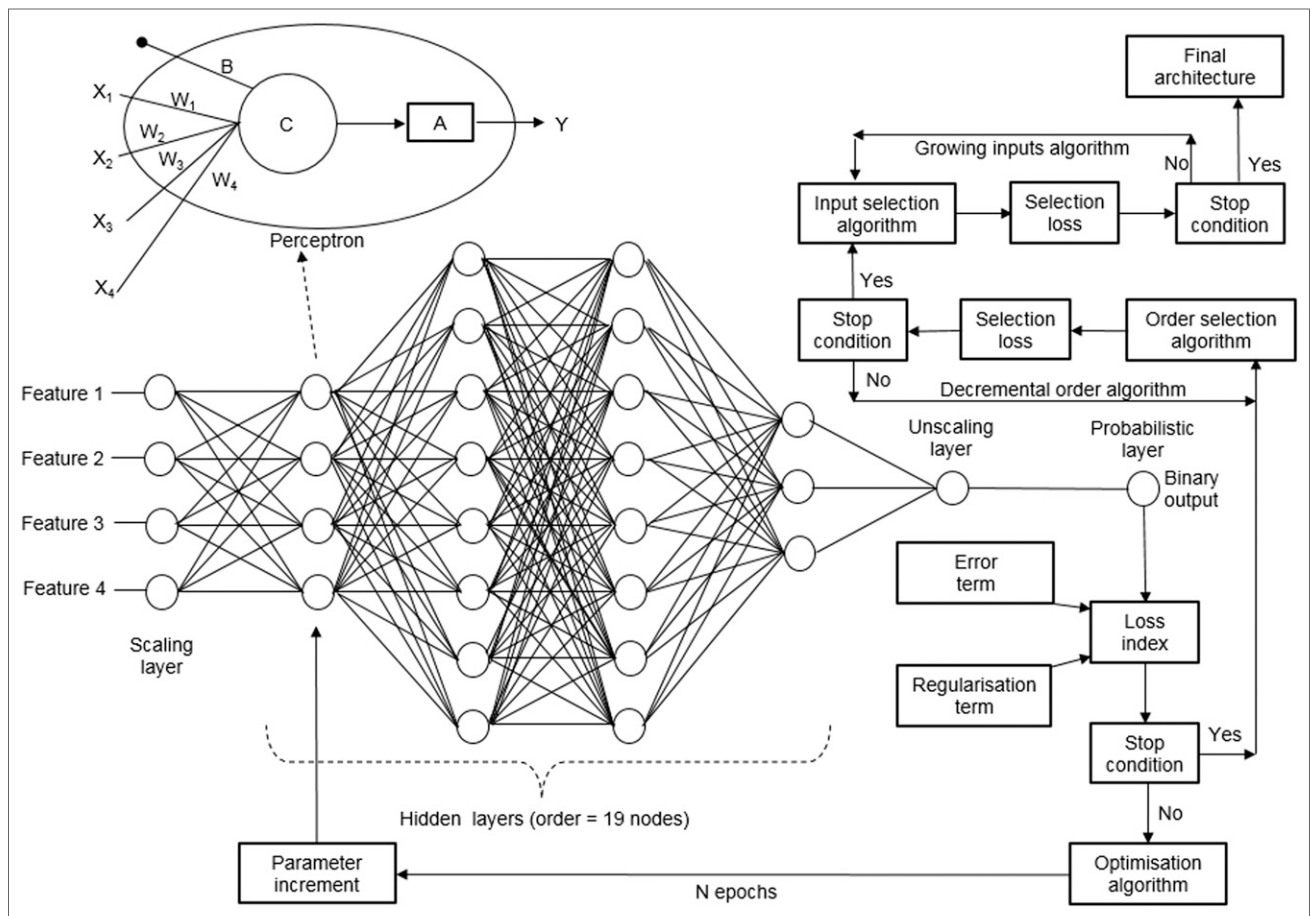
FIGURE 3. Anatomy of an ANN. Single node (C) can have multiple inputs (X) with different weighting factors (W) and bias (B) but single output (Y) via activation function (A). Multiple lines exiting each node are same output being delivered to multiple next-layer nodes.

using an iterative process of successive adjustments to the weightings. Gradient is the rate of inclination or declination (slope) and represents the learning rate. Gradient descent is an optimization method that evaluates a progressive diminishing rate of learning with each iteration (*5,9–11*). That is, the cost function is decreasing, which means the loss is decreasing and the minimum point could be used to terminate the cycle (before loss starts to increase again) (*5,7,8,10,11*). Large datasets may not be able to be processed concurrently, requiring division of the data. An epoch refers to the entire dataset being passed forward (forward propagation) and backward (back propagation) through the ANN once. This process is often referred to as an iteration, and for small datasets an iteration and an epoch are the same. In larger datasets, the data may need to be broken into batches of smaller units. Each forward propagation and back propagation of a batch through the ANN is an iteration. Passage of all batches through once is an epoch. For the dataset of 1,000 patients, the data may need to be broken into batches of 200, resulting in 5 batches requiring 5 iterations to complete 1 epoch. The optimization algorithm, therefore, changes parameters between successive epochs (parameter increment) to minimize the loss index until a specified con-

dition is met (e.g., minimum value is reached, margin loss improvement equals a set value, gradient equals a preset value, maximum number of epochs is reached, maximum time is reached) (*5,10*). The optimization algorithm itself defines how parameters are optimized (*9,10*). The Newtonian method is computationally demanding but more accurate, using the Hessian of the loss function (second-derivative matrix) (*9*). A quasi-Newtonian method may be a preferred option, and this approach uses gradient information to estimate the inverse Hessian (mathematical function using a square matrix of second-order partial derivatives) for each iteration of the algorithm, ignoring second derivatives and reducing computational demand. Other approaches include gradient descent, conjugate gradient, Levenberg–Marquardt algorithm, stochastic gradient descent, and adaptive linear momentum. The loss function associated with the training phase estimates the error associated with the prediction and the grounded truth for the dataset (*5,9,10*). The selection loss is an error measure of the ANN's generalizability to new data, or agility. These loss functions can be used to optimize the number of hidden layers or iterations in the final architecture.

The final architecture of the ANN or model selection needs to consider selection loss, or minimize the error associated
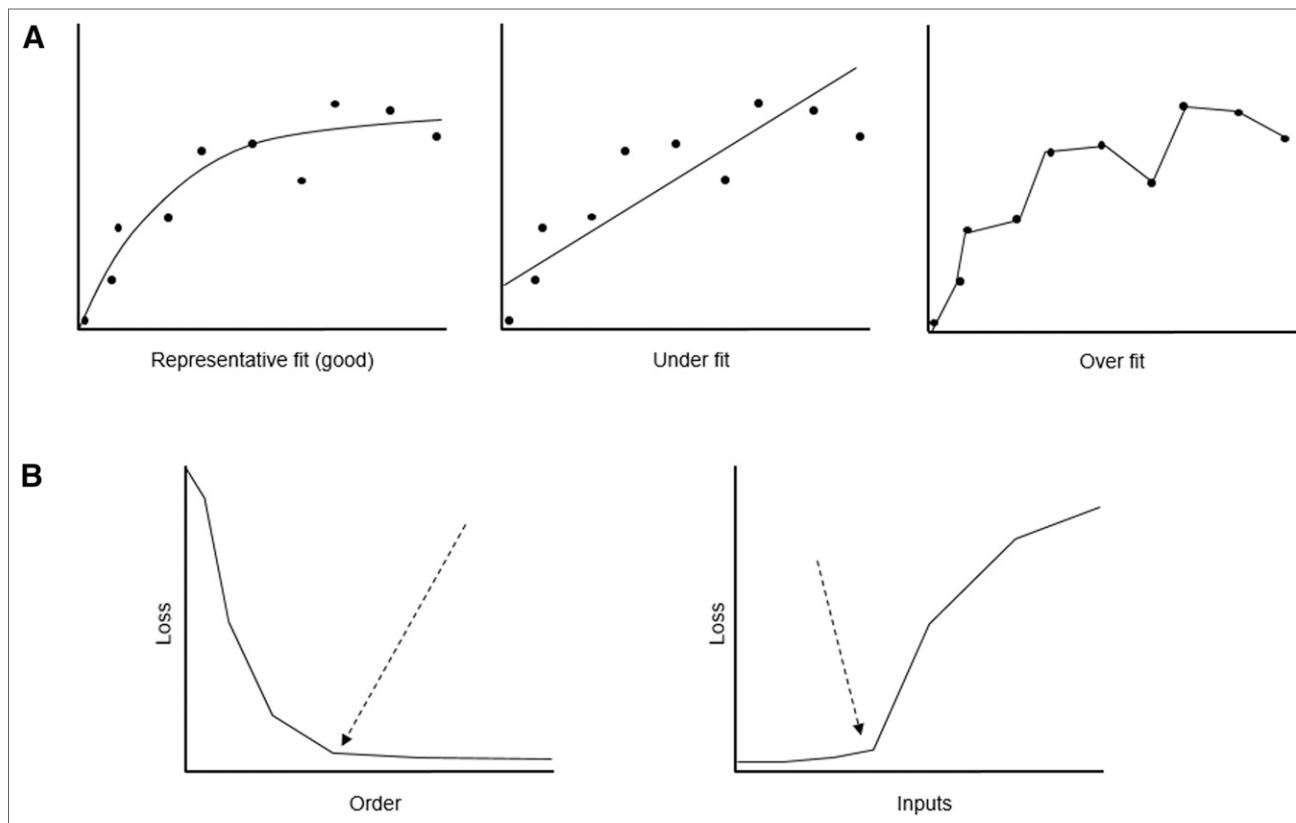
FIGURE 4. (A) Schematic representation of good fit vs. under- and overfitting associated with selection loss. (B, left) Optimization of selection loss to determine ANN complexity and node number (order) using decremental order algorithm, with arrow indicating reasonable cutoff for total node number. (B, right) Optimization of selection loss to determine inputs (features) to be included using growing-input algorithm, with arrow indicating reasonable cutoff for inputs.

with the order and range of inputs (*5,7,9,10*). Order selection relates to the depth of the ANN, or its influence on the output and its accuracy, by defining the number of nodes in hidden layers (*5,9,10*). It is important to balance the order selection with the complexity of the data to avoid under- or overfitting (Fig. 4A) (*4*). Similarly to the training error, the selection error measures the accuracy of the ANN applied to new data (generalizability) (*5,9,10*). An incremental order-selection algorithm starts by measuring selection loss for a small number of nodes and incrementally adds nodes until the selection loss is optimized (meets a predetermined value). Conversely, a decremental order algorithm starts by measuring selection loss for a large number of nodes and incrementally removes nodes until the selection loss is optimized (meets a predetermined value). In this case, knowing the low complexity of data, the user has elected to begin with a more complex ANN than necessary, which will see the decremental order algorithm reduce the complexity in the ANN.

Input selection (Fig. 4B) defines which specific features should be included in the ANN inputs. The input selection algorithm determines which input features produce the smallest selection error and, thus, provide the best generalizability for the ANN to new data (*5,9,10*). There are several algorithms that can be used. The pruning method

starts with all inputs and incrementally removes inputs with the lowest correlation until the selection loss starts to rise. A growing-input method can also be used to calculate the correlation for every input against each output in the dataset. Beginning with the most highly correlated inputs, inputs are incrementally added to the network until the selection loss increases. The final architecture of the neural network reflects the optimized subset of inputs and order with the lowest selection loss (Supplemental Fig. 2).

Several metrics can be used to test the errors in the neural network. The final architecture can then be evaluated using several tests for robust validation using a second set of data (or a validation partition of the original dataset) (*9,10*). The loss index for the final ANN can be calculated by comparing the prediction output with the grounded truth (*7*). Several tools are used in combination for validation, such as sum squared error, mean squared error, root mean squared error, normalized squared error, Minkowski error, cross entropy error, hinge error, and linear regression analysis. Receiver-operating-characteristic analysis produces an area under the curve that correlates with a sensitivity and specificity (*9,10*). This correlation is further reflected in the confusion matrix (true positives, true negatives, false negatives, and false positives). ANN performance may also be expressed or displayed as cumulative gain (benefit of

using the ANN over a random guess), lift chart (ratio of positive events using the ANN to those without the ANN), conversion rates (percentage of predicted cases with and without the ANN), and profit chart (ANN gain over random guess). Much of the literature on ML applications in nuclear medicine and radiology are in some way the validation phase of the ANN. This may include statistical analysis of the ANN capability against human interpretation and a gold standard. It may also include an evaluation of the predicted gain in economic or health outcome terms with and without the ML model. After validation, the ML algorithm can be implemented by exporting and applying the mathematical model. For simple ML and ANN models, this may represent an export of the mathematical expression in simple code language such as Python for incorporation in mobile device apps or websites.

An example of this application is previous work with ^{123}I-metaiodobenzylguanidine radionuclide imaging in heart failure. Traditional analysis with multivariate approaches demonstrated that regional washout associated with territories adjacent to infarcted myocardium was superior to traditional planar approaches to uptake and washout in predicting cardiac events. Subsequently, the same data were evaluated using an ANN in the method described above using 84 input variables and a single binary output (cardiac event or no cardiac event in the follow-up period). Training and validation phases optimized the number of inputs at just 2: a decrease in left ventricular ejection fraction ($>10\%$) and ^{123}I-metaiodobenzylguanidine planar global washout ($>30\%$) (*12,13*). The ANN in this case revealed predictive capability not illuminated by traditional regression methods, highlighting the value of ANN and ML in parallel to conventional statistical analysis.

ANATOMY OF A CNN

With the general principles of an ANN outlined above, scaffolding a deeper insight into the CNN process might be of value. As outlined in Figure 2, a CNN comprises convolution and pooling layers and the fully connected layers of a neural network. The CNN differs from the ANN described in Figure 3 in that the features are extracted from the images and the output is some form of classification (*7*). As described below, the CNN transforms 2-dimensional image data through forward propagation but can also be applied to 3-dimensional datasets such as SPECT and PET (*7,11*).

Convolution is the extraction of image features using a linear operation that applies a kernel (typically 3×3) to a subset array of image elements (pixels) or input tensor (Fig. 5) (*5,7–9,11,14*). This process is not dissimilar to the application of a 9-point smoothing filter to planar images in nuclear medicine. The kernel is positioned over elements in the input tensor, with the distance between each successive position representing the stride (*5,7–9,11,14*). A stride of 1 means that the kernel is centered over each element of the input tensor, whereas a stride of 2 indicates centering over every second element of the input tensor. This down-sampling of feature maps with strides greater than 1 can be better achieved in the pooling function (*5,7–9,14*). The product of the individual elements of the input tensor and the kernel are summed to produce a single numeric value (and position ordinates) into the feature map (output tensor) (*5,7–9,14*). A variety of kernels can be applied in a stepwise manner producing several convolution layers (Fig. 2). Of importance in convolution is that although the *x* and *y* dimensions of the input tensor are compressed, the *z* dimension does not change. The postconvolution feature map is then passed through a nonlinear activation function that, as previously described, is typically the rectified linear unit, before entering the pooling layer (*5,7–9,14*).

Pooling reduces the in-plane (*x, y*) dimensionality of feature maps by applying a down-sampling operation (*5,7–9,11,14*). Max pooling and global average pooling are 2 common approaches. As the name suggests, max pooling creates an output equal to the maximum value within a patch of data in the feature map (*5,7–9,14*). A 2×2 filter with a stride of 2 means that each set of 4
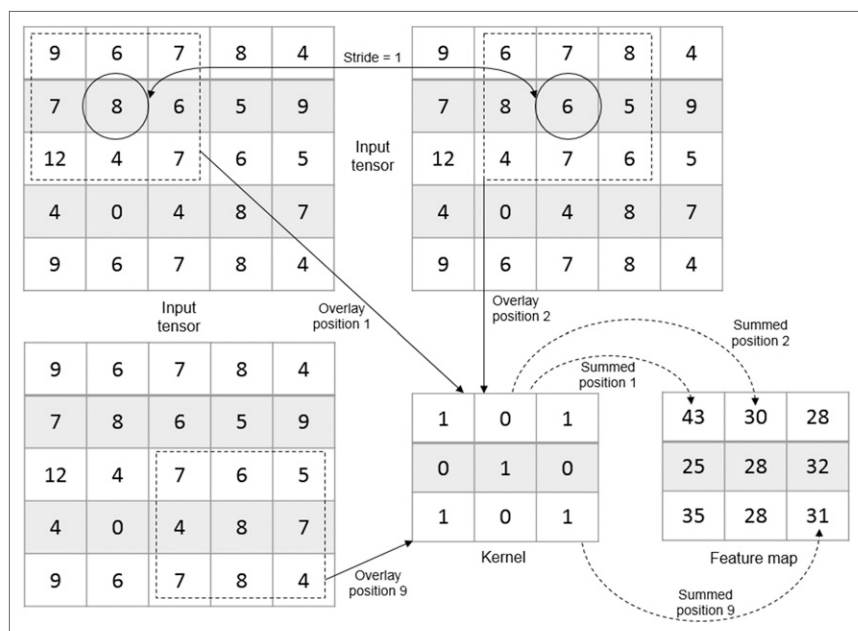
FIGURE 5. Convolution uses 3×3 kernel to run sequential (in this case, successive, to provide stride of 1) 3×3 array of elements. Weighted sum of kernel for 3×3 input tensor creates single representative value in feature map. Multiple feature maps are produced by different kernels.
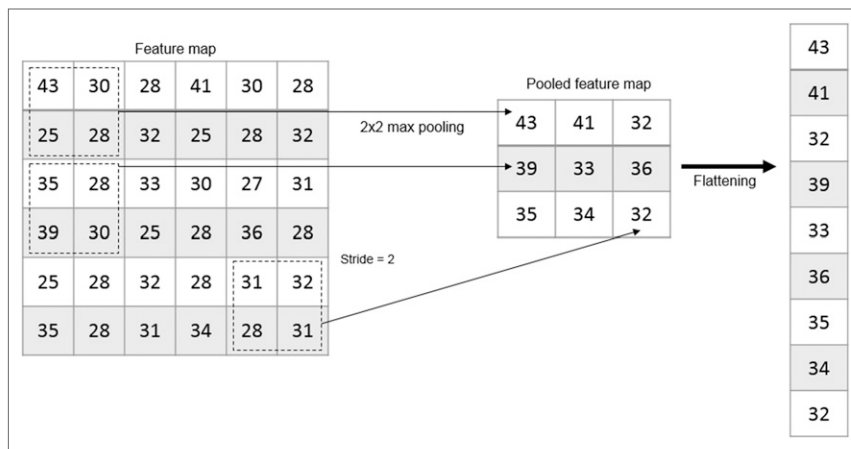
FIGURE 6. Pooling using max pooling method and 2 × 2 array produces pooling of maximum count among 4 connected elements (patch) to represent those data in pooled feature map. Use of consecutive blocks of 2 × 2 elements means stride of 2. Final pooled feature map immediately before input into neural network can then be flattened from 2-dimensional data into single dimension; this approach avoids need for global pooling.

elements is represented as a single value equal to the maximum value and all other data are discarded (Fig. 6). Global pooling, on the other hand, represents a feature map as a single value equal to the mean of the element values, essentially down-sampling a feature map to a 1 × 1 array (5,7–9,14). Global pooling is typically applied once, immediately before the fully connected layers; however, the max pooling method is more common (5,7–9,11,14). Multiple sequential convolution, kernel, and pooling processes produce layers of data that are transformed into a 1-directional array of vectors (numbers) through a process called flattening (5,7,11).

A parameter is a variable automatically learned by the CNN, whereas a hyperparameter is a variable that needs to be set (7). These vary in the different layers of the CNN (Fig. 7). In the convolution layer, kernels are the parameter, and kernel size, kernel number, stride, and activation function are the hyperparameters. The pooling layer has no parameters, but the pooling method, filter size, and stride are all hyperparameters. The fully connected layer of nodes uses weights as the parameters, whereas the activation function and the number of weights are the hyperparameters.

There are a wide variety of applications of CNN and DL in nuclear medicine, but application of a CNN has been effectively demonstrated in recent dementia studies. SPECT images with known outcomes were used to train a CNN to evaluate the images themselves and identify key features—specifically, the cingulate island sign indicative of dementia with Lewy bodies (15). Perhaps a more important approach would be the use of a CNN trained to identify specific features on the images themselves indicating findings of an urgent nature—pulmonary embolism on a lung scan, for example. Rather than the CNN's providing a definitive diagnosis, a list-based report
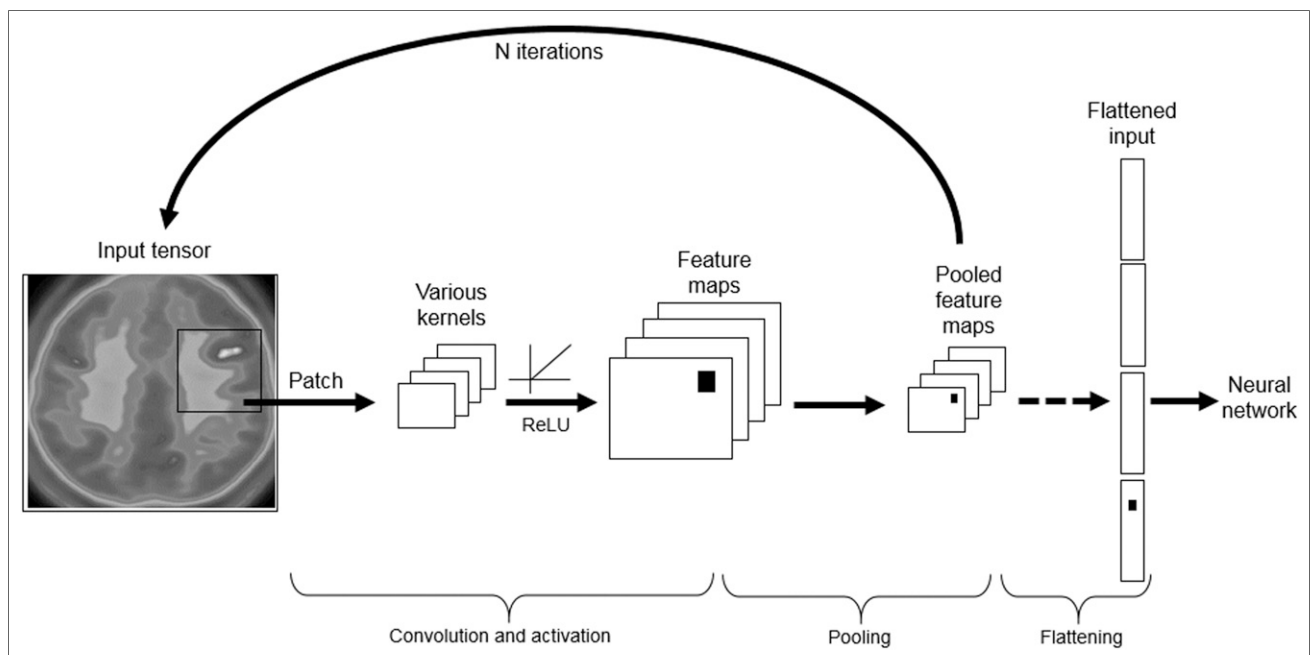
FIGURE 7. CNN will have several convolution and pooling layers before flattening and input to neural network. Several kernels can be used on same input tensor to produce layers of feature maps via rectified linear units (ReLU) for pooling and eventually flattening.

could be initiated and the findings used to triage a positive outcome to the front of the reporting list. Clearly, a CNN could be readily trained to identify features to drive automated segmentation or region identification, and this ability may have significant applications in radiation dosimetry (*16*).

However, a degree of caution is required with application of DL and CNNs. Although a CNN can identify features, or relationships between features, in a large volume of data not possible for a human observer, unsupervised learning may see unusual features identified. For example, consider a CNN trained to identify pulmonary embolism on a lung scan. If that CNN were shown to be more accurate than a human observer in detecting pulmonary embolism, it makes sense that the CNN has identified features not typically considered by the human observer. This should prompt enquiry to educate the human observer about previously unconsidered features. In theory, the entire process improves. It may, however, reveal that instead of identifying features in the lung fields themselves, the CNN has learned other features that strongly correlate with pulmonary embolism: electrocardiography leads in situ, annotation indicating referral from emergency, patient age or sex. Anecdotal discussion at conferences revealed that a CNN to detect pneumonia on chest radiographs was making decisions based on whether the study was performed in the department or by a mobile radiography unit.

DISCUSSION

Despite an emergence of medical literature outlining the applications of DL and CNNs in nuclear medicine and radiology, AI, ML, ANNs, and CNNs afford numerous opportunities besides those mentioned in the literature. There are several key areas in which AI, ML, ANNs, or DL has been successful or potentially impactful in radiology (*2,3,17*), and these are equally apt for nuclear medicine:

- Prediction of positivity rate among similar patients to inform diagnostic decision trees and optimize procedure choice.
- New image reconstruction methods that produce images from lower-radiation-dose studies (e.g., PET and SPECT); generation of pseudo-CT for attenuation correction or with reduced imaging time (e.g., MRI).
- Quality assessment algorithms built into systems to improve image quality and decrease repeat studies.
- Image-triage algorithms that identify cases likely to be positive or that may have an urgent finding, allowing prioritization of reporting and earlier intervention.
- Computer-aided detection, automated image annotation, and information extraction.
- AI methods that explain analysis and interpretation and provide preliminary reporting.
- Lesion or disease detection (enhanced computer-aided detection) and classification.
- Automated segmentation, identification and extraction of features from images (radiomics), and quantitation.

Detection of incidental findings is an important potential application of AI and ANNs not generally discussed in the literature but readily expressed in a mathematical algorithm (variation from normal). The emergence of the important role of radiation dosimetry modeling in radionuclide therapy will elevate precision nuclear medicine and theranostics, no doubt unveiling an important application of AI and ANNs.

The future of AI is promising and looks beyond DL. Patrick Ehlen from Loop AI Labs explained in 2018 at a conference in Cologne, Germany (https://www.loop.ai/ai-the-end-of-deep-learning?contentid=1302036), that the next generation of AI will go beyond DL. He used the liar paradox from Star Trek to highlight that AI is trained to solve problems logically. The human brain not only is capable of logical thought but also operates in the sphere beyond logic (an agility known as super logic, or sometimes referred to as illogical). The simple liar paradox of AI's interpreting 2 pieces of information, the first being "everything Harry tells you is a lie" and the second coming from Harry saying, "I am lying," defies first-order logic (https://www.youtube.com/watch?v=QqCiw0wD44U). Higher-order logic that would prevent AI from being outwitted by human super logic requires a framework of quantum-based logic. Although a tutorial on quantum computation is beyond the scope of this article (*18*), the basic premise is that AI does not understand pragmatics. Humans process the contrasting context associated with pragmatism. These different foundation contexts could be seen as different basis vectors in quantum probability theory and allow AI to develop higher-order reasoning and problem-solving skills. The differentiation of imitation of human thought (AI) and manufactured but authentic intelligence gives rise to the concept of synthetic intelligence (SI). This capability has the potential to make dramatic steps in interpretation of complex images and pathologic states associated with PET, SPECT, MRI, and CT. The field of nuclear medicine and radiology has its strength in making clinical judgments and decisions based on data and feature extraction, not in the feature extraction or analysis itself (*19*). Thus, AI techniques such as ML and DL provide an opportunity to enhance the accuracy and efficiency of the physician or radiologist without threatening redundancy. AI may represent a shift in practice; AI has a high degree of capability in rudimentary tasks—tasks that will thus be lost to the radiologist or physician, but this loss simply provides more time for the radiologist or physician to focus on the higher-order semantic tasks that are beyond, but enhanced by, the capabilities of AI. On the surface, this is a strong argument against the idea that AI may make the physician or radiologist redundant. Quantum logic in SI may renew that debate.

CONCLUSION

AI has penetrated the daily practice of nuclear medicine over recent decades with little disruption. The emergence of ANNs and CNN applications has seen the landscape

undergo a significant shift whose opportunity outweighs the threat. Nonetheless, understanding of the potential applications and the principles of AI, ANNs, and DL will equip nuclear medicine professionals for ready assimilation, averting the doomsday fears permeating radiology. Counter to the concerns among radiologists, in nuclear medicine the disruptive potential of the technology is perhaps of greatest impact on technologists and physicists rather than physicians; those of us most likely to develop and apply AI, ML, and DL in the research and clinical environment.

DISCLOSURE

No potential conflict of interest relevant to this article was reported.

REFERENCES

1. Currie GM. Intelligent imaging: artificial intelligence–augmented nuclear medicine. *J Nucl Med Technol.* 2019;47:217–222.
2. Tang A, Tam R, Cadrin-Chenevert A, et al. Canadian Association of Radiologists white paper on artificial intelligence in radiology. *Can Assoc Radiol J.* 2018;69:120–135.
3. McBee MP, Awan OA, Colucci AT, et al. Deep learning in radiology. *Acad Radiol.* 2018;25:1472–1480.
4. Ben-Nun T, Hoefler T. Demystifying parallel and distributed deep learning: an in-depth concurrency analysis. arXiv.org website. https://arxiv.org/abs/1802.09941. Published February 26, 2018. Last revised September 15, 2018. Accessed August 14, 2019.
5. Lundervold AS, Lundervold A. An overview of deep learning in medical imaging focusing on MRI. *Z Med Phys.* 2019;29:102–127.
6. Uribe CF, Mathotaarachchi S, Gaudet VC, et al. Machine learning in nuclear medicine: part 1—introduction. *J Nucl Med.* 2019;60:451–458.
7. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights Imaging.* 2018;9:611–629.
8. Maier A, Syben C, Lasser T, Riess C. A gentle introduction to deep learning in medical imaging processing. *Z Med Phys.* 2019;29:86–101.
9. Goodfellow I, Bengio Y, Courville A. *Deep Learning.* Cambridge, MA: MIT Press; 2016.
10. Nielsen MA. *Neural Networks and Deep Learning.* San Francisco, CA: Determination Press; 2015.
11. Shen D, Wu G, Suk H. Deep learning in medical image analysis. *Annu Rev Biomed Eng.* 2017;19:221–248.
12. Iqbal B, Currie G, Kiat H. Regional cardiac sympathetic nervous system evaluation using ^{123}I-mIBG SPECT in patients with heart failure. *J Med Imaging Radiat Sci.* 2018;49:397–405.
13. Currie GM, Iqbal B, Wheat JM, et al. Risk stratification in heart failure using ^{123}I-MIBG. *J Nucl Med Technol.* 2011;39:295–301.
14. Lin M, Chen Q, Yan S. Network in network. arXiv.org website. https://arxiv.org/pdf/1312.4400.pdf. Published March 4, 2014. Accessed August 14, 2019.
15. Iizuka T, Fukasawa M, Kameyama M. Deep-learning-based imaging-classification identified cingulate island sign in dementia with Lewy bodies. *Sci Rep.* 2019;9:8944.
16. Jackson P, Hardcastle N, Dawe N, Kron T, Hofman MS, Hicks RJ. Deep learning renal segmentation for fully automated radiation dose estimation in unsealed source therapy. *Front Oncol.* 2018;8:215.
17. Langlotz CP, Allen B, Erickson B, et al. A roadmap for foundational research on artificial intelligence in medical imaging: from the 2018 NIH/RSNA/ACR/The Academy Workshop. *Radiology.* 2019;291:190613.
18. Ying M. Quantum computation, quantum theory and AI. *Artif Intell.* 2010;174:162–176.
19. Liew C. The future of radiology augmented with artificial intelligence: a strategy for success. *Eur J Radiol.* 2018;102:152–156.