Shallow and Deep Neural Networks

Corey Waldner

School of Technology and Engineering, National University

TIM-8131 v1: Data Mining

Dr. Reza Hashemi

September 14, 2024

Shallow and Deep Neural Networks

Deep learning falls under the umbrella of artificial intelligence and machine learning. While deep learning and machine learning are frequently used interchangeably, they serve distinct purposes and involve different processes. Machine learning primarily uses algorithms to train models and make predictions based on structured data. In contrast, deep learning is a more advanced learning method for computations using multi-layer neural networks (Sarker, 2021). Within deep learning, two fundamental types of network architecture are utilized: Neural Networks (NNs) and Deep Neural Networks (DNNs), each representing different approaches to processing data. This paper will explore the distinctions between an NN and a DNN and then apply these distinct approaches to a dataset for comparative analysis.

Neural Network

An Artificial Neural Network (ANN) imitates the human brain's process of information processing. Put simply, the human brain takes in input through dendrites and outputs the necessary information through axons. Dendrites receive information, while axons relay the information to the correct neurons or 'nodes' for processing, allowing the brain to make decisions rapidly (Wood, 2022). The process is mirrored in an ANN, which transforms information and decisions through multiple hidden layers. In an ANN architecture, information is received in an input layer (analogous to dendrites), processed in a hidden layer(s) (analogous to neurons), and the result of the classification or decision is delivered through the output layer (analogous to axons).

The entire process is conducted using calculus-based optimization techniques. In simple mathematical terms, an ANN can be expressed as $Y = f(X)$, where $Y= f(g(b))$, and $X = g(b)$. Here, b, X, and Y represent vectors or, more generally, tensors. In this context, vectors are onedimensional arrays of numbers, matrices are two-dimensional arrays, and tensors denote n-dimensional arrays. The training process of an ANN essentially entails determining optimal values for the coefficients *b* to complete the mapping (Sarker, 2021)

Training

The NN training process comprises three primary stages: forward propagation, loss function calculation, and backward propagation. Forward propagation involves passing input data through the network, layer by layer, to generate an output in a forward direction. During this process, the data is processed by hidden layers, which apply activation functions and pass the transformed data to the next layer. Forward propagation ensures data moves linearly through the network, preventing circular data flow and generating a valid output (H2O.Ai, 2024).

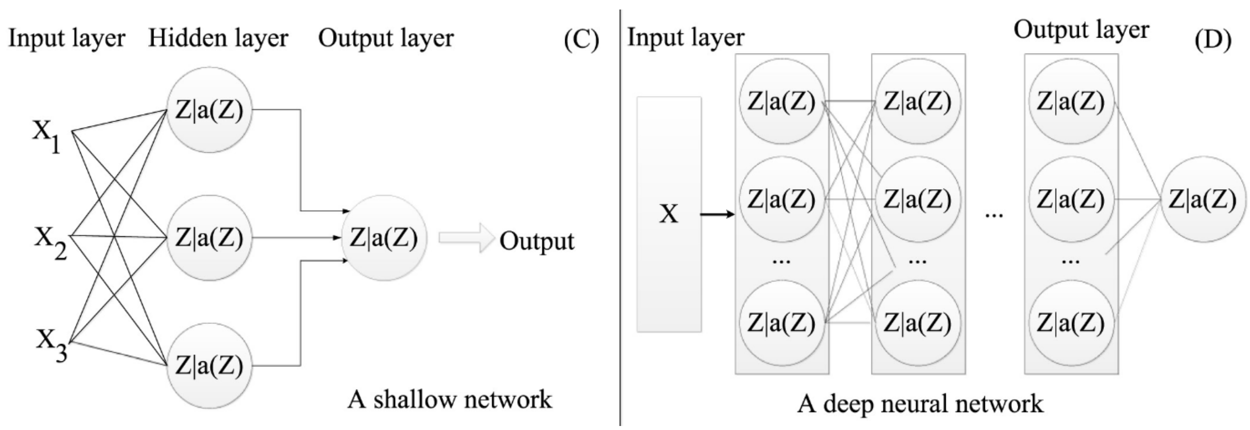
After the forward propagation process, the loss function is computed. This function evaluates the disparity, also known as "error," between the network's predicted output and the actual target output. The loss function quantifies the extent to which the network's predictions deviate, thereby aiding in the adjustment of the network (Medium, 2023).

After the initial calculations, backpropagation adjusts the network's parameters. It involves computing the gradient of the loss function in relation to the neural network's weights and biases. This backward process employs the chain rule from calculus to iteratively modify the weights, effectively reducing errors in subsequent forward passes (D2L, 2024).

Deep Neural Network

A DNN shares the fundamental architecture of a NN but includes many more hidden layers between the input and output layers. As illustrated in Figure 1, a DNN consists of multiple interconnected layers of nodes, which enables the model to capture more intricate and abstract data representations. With additional layers, the network has more opportunities to learn from previous iterations, progressively refining its understanding. Each layer builds on the knowledge from the one before, allowing the network to recognize complex patterns and make more accurate predictions in tasks like image recognition, natural language processing, and speech recognition (AWS, 2024).

Figure 1
Shallow Network vs. Deep Neural Network



Note. This visualizes the differences in how data is processed with different architectures. Source (Dong, Wang, & Abbas, 2021)

While DNNs offer increased accuracy, they have drawbacks, including longer training times, higher computational costs, and more complex architectures. As shown in Figure 2, the trade-offs between using a NN and a DNN largely depend on the problem being addressed. The strength of a NN lies in its simplicity and lower computational demands, making it efficient for less complex data. However, it struggles with more intricate datasets. In contrast, a DNN excels in handling complex data and achieving higher accuracy but at the cost of increased complexity and computational resources.

Figure 2
NN vs. DNN Table Comparison

Feature	Neural Network (NN)	Deep Neural Network (DNN)
Number of Hidden Layers	1 or 2 layers	3 or more
Training Time	Faster training due to fewer layers	Slower training due to more layers
Computation Cost	Low	High
Accuracy	Moderate accuracy for simple tasks	Higher accuracy for complex tasks
Best Use Cases	Simple - binary classification, linear regression, etc.	Complex - image recognition, natural language processing
Strengths	Easy implementation Faster convergence	Can learn complex representations Better performance on large datasets
Weaknesses	Limited capacity to capture complex patterns Poor performance on large-scale tasks	Slower and more computationally expensive Requires large datasets for effective training

Note. *This is a comparison table between NN and DNN highlighting Strengths and Weaknesses.*

Practical Application

A practical application involving both NN and DNN was selected to be executed on the CIFAR-10 dataset. CIFAR-10 is commonly used for training in the field of computer vision. This dataset comprises 60,000 32x32 color images categorized into ten classes, with 6000 images per class. It includes 50,000 training images and 10,000 test images (Krizhevsky, 2014). When executing the code for the CIFAR-10 dataset, the Convolutional Neural Network (CNN) algorithm is employed. CNN is a specialized deep-learning algorithm designed for tasks requiring object recognition, such as image classification, detection, and segmentation. CNNs are extensively used in practical scenarios like autonomous vehicles, security camera systems, and more (Keita, 2023).

The model utilized a shallow NN with a Sequential architecture comprising three layers. The initial layer served as an input layer where the 32x32x3 image data was flattened into a 3072-element vector to facilitate processing by the subsequent fully connected layers. The second layer was a fully connected hidden layer housing 128 neurons, involving 393,344 trainable parameters, encompassing the weights and biases essential for linking the 3072 input elements to the 128 neurons. The parameter calculation in this hidden layer was derived from the formula (3072 * 128) + 128 = 393,344. Lastly, the model included an output layer with ten units correlating to the ten different classes of the CIFAR-10 dataset, accommodating 1,290 parameters. This quantity was determined by establishing connections from the 128 units in the hidden layer to the ten output units, using the formula (128 * 10) + 10 = 1,290, as illustrated in Figure 3.

Figure 3
NN Result

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 3072)	0
dense (Dense)	(None, 128)	393,344
dense_1 (Dense)	(None, 10)	1,290

```
Total params: 394,634 (1.51 MB)
Trainable params: 394,634 (1.51 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.2758 - loss: 2.0275 - val_accuracy: 0.3650 - val_loss: 1.7861
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.3625 - loss: 1.7860 - val_accuracy: 0.3777 - val_loss: 1.7575
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.3882 - loss: 1.7330 - val_accuracy: 0.3941 - val_loss: 1.6991
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4053 - loss: 1.6894 - val_accuracy: 0.4151 - val_loss: 1.6477
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4120 - loss: 1.6589 - val_accuracy: 0.4119 - val_loss: 1.6443
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4155 - loss: 1.6446 - val_accuracy: 0.4246 - val_loss: 1.6284
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4208 - loss: 1.6273 - val_accuracy: 0.4185 - val_loss: 1.6279
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4320 - loss: 1.6063 - val_accuracy: 0.4246 - val_loss: 1.6048
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.4306 - loss: 1.6003 - val_accuracy: 0.4240 - val_loss: 1.6166
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4347 - loss: 1.5921 - val_accuracy: 0.4377 - val_loss: 1.5909
313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 326us/step - accuracy: 0.4397 - loss: 1.5891
Shallow model test accuracy: 0.4377000033855438
```

Note. *This is the output for creating and evaluating an NN against the CIFAR-10 dataset.*

The model underwent ten training iterations, passing the entire dataset through the network multiple times. It started with an accuracy of 27.58% in the first iteration and showed steady improvement throughout training. By the final epoch, the model had achieved an accuracy of 43.47%, marking an overall increase of nearly 16%. This progression also illustrates the model's capacity to learn and enhance its predictive abilities through training.

The DNN was approached similarly to the NN, as depicted in Figure 4, but exhibited some differences. The DNN comprises ten layers, resulting in fewer parameters overall due to weight sharing across spatial dimensions in convolutional layers instead of fully connected layers in the NN. This architectural choice significantly improved accuracy over the ten training iterations, leading to a performance enhancement of over 40% to over 70% accuracy. This outcome underscores the advantages of a DNN, where multiple layers enable the model to learn more intricate and abstract features, ultimately contributing to improved accuracy.

Figure 4
DNN Output

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_2 (Conv2D)	(None, 4, 4, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 128)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	65,664
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

```
Total params: 160,202 (625.79 KB)
Trainable params: 160,202 (625.79 KB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 13s 8ms/step - accuracy: 0.3002 - loss: 1.8634 - val_accuracy: 0.5245 - val_loss: 1.3115
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 15s 10ms/step - accuracy: 0.5184 - loss: 1.3474 - val_accuracy: 0.5581 - val_loss: 1.2137
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 16s 10ms/step - accuracy: 0.5891 - loss: 1.1702 - val_accuracy: 0.6255 - val_loss: 1.0613
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.6371 - loss: 1.0430 - val_accuracy: 0.6558 - val_loss: 0.9849
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 15s 10ms/step - accuracy: 0.6708 - loss: 0.9462 - val_accuracy: 0.6833 - val_loss: 0.9169
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.6934 - loss: 0.8793 - val_accuracy: 0.6907 - val_loss: 0.8957
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.7196 - loss: 0.8082 - val_accuracy: 0.7019 - val_loss: 0.8810
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.7298 - loss: 0.7717 - val_accuracy: 0.7098 - val_loss: 0.8459
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.7450 - loss: 0.7266 - val_accuracy: 0.7113 - val_loss: 0.8635
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 17s 11ms/step - accuracy: 0.7630 - loss: 0.6847 - val_accuracy: 0.7198 - val_loss: 0.8451
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.7204 - loss: 0.8374
Deep model test accuracy: 0.7197999954223633
```

Note. *This is the output for creating and evaluating an DNN against the CIFAR-10 dataset.*

Conclusion

In conclusion, NN and DNN are forms of deep learning, each offering unique benefits depending on the specific problem. NNs, with their simpler architectures, excel in efficiency and decreased computational costs but struggle with handling more intricate data. On the other hand, DNNs, with their multiple hidden layers, can effectively capture more complex patterns and achieve higher accuracy in tasks such as image and speech recognition. A practical comparison of these models using the CIFAR-10 dataset highlighted the advantages of a deeper architecture, as the DNN significantly outperformed the NN in terms of accuracy. Ultimately, the choice between NN and DNN should be determined by the complexity of the task, available computational resources, and the desired level of accuracy.

References

AWS. (2024, September 14). *What's the Difference Between Deep Learning and Neural Networks?* Retrieved from AWS.Amazon: https://aws.amazon.com/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=In%20a%20simple%20neural%20network,layers%20that%20make%20them%20deep.

D2L. (2024, September 14). *5.3. Forward Propagation, Backward Propagation, and Computational Graphs*. Retrieved from Dive into Deep Learning: https://d2l.ai/chapter_multilayer-perceptrons/backprop.html

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications . *Computer Science Review*, 40.

H2O.Ai. (2024, September 14). *What is a Neural Network?* Retrieved from H20.Ai: https://h2o.ai/wiki/forward-propagation/#:~:text=Forward%20propagation%20is%20the%20way,loss%20function%20or%20error%20rate.

Keita, Z. (2023, November). *An Introduction to Convolutional Neural Networks (CNNs)*. Retrieved from Data Camp: https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

Krizhevsky, A. (2014, October 18). *CIFAR-10 - Object Recognition in Images*. Retrieved from Kaggle: https://www.kaggle.com/c/cifar-10/overview

Medium. (2023, May 26). *Deep Learning Course — Lesson 5: Forward and Backward Propagation*. Retrieved from Medium: https://medium.com/@nerdjock/deep-learning-course-lesson-5-forward-and-backward-propagation-ec8e4e6a8b92

Sarker, I. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6).

Wood, T. (2022, July 5). *How similar are Neural Networks to our Brains?* Retrieved from Fast Data Science: https://fastdatascience.com/ai-in-research/how-similar-are-neural-networks-to-our-brains/