Deep learning architectures and their performance in speech recognition: Neural Networks

Eva Pérez S.

ABSTRACT

In neural networks, theres a field where deep learning takes part, in the next sections a research is done on the different types of architecture designs that can enhance the performance of specific deep neural networks such as natural language processing, image recognition, general analysis, speech recognition, healthcare, robotics, social media, etc. just to mention a few. For this research we compare CNN, RNN, LSTM methods to understand and discover the best neural network architecture for speech recognition. The purpose is to define if the designed architectures could outperform manually designed ones. We looked at different types of neural networks like CNN, RNN, and LSTM. Each has its strengths. CNNs are great at picking out important features, especially for understanding images. RNNs are like experts in understanding things that happen one after the other, making them useful for understanding language. Then, there's LSTM, which is excellent at remembering things for a long time, which is handy in lots of areas, including recognizing speech. When we focused on speech recognition, our study found that LSTM is really good at understanding the details and patterns in spoken words. This tells us that LSTM are a big help in understanding how human speech is done in talking. So, in summary, for tasks like recognizing speech, LSTM is the best choice because of its memory performance.

Keywords: Neural Networks, Machine Learning, Artificial Intelligence

INTRODUCTION

Neural networks is an extended topic with diverse applications and implications in between. It began with the investigation Warren S. McCulloch and Walter Pits did with "A logical calculus of the ideas immanent in nervous activity". This research aimed to gain a better understanding of how the human brain processes complex patterns through neurons. The study compared this biological process with a binary threshold to boolean logic (0/1 or true/false). The primary objective of this paper is to explore the historical roots of neural networks, tracing their evolution from early research to the present day. Additionally, the purpose is to highlight the foundational concepts, such as the binary threshold model, that laid the groundwork for the development of artificial neural networks and their diverse applications.

HISTORY

The history of neural networks have been a long journey of evolution, from the mathematical models proposed by McCulloch and Pitts in 1943 to the present, where deep learning is applied for advanced research and solving problems. The concept of a perceptron was introduced in the 1950s with Rosenblatt, where a single-layer neural network is capable of learning simple binary classifications, then in the 1960s, a book was published by Marvin Minsky and Seymour Papert discouraging interest in perceptrons by writing it's limitations at that time, like its inability to handle non-linearly separable functions. Considering the previous lack of interest, in the 1980s a backpropagation algorithm by Paul Werbos addressed the training issues for multi-layer networks, increasing the interest once again, then it became a crucial algorithm in neural networks to this day. In 1989, Convolutional Neural Networks (CNNs) introduced by Yann LeCun, made it so image recognition was possible using this architecture. In the 1990s, Recurrent Neural Networks (RNNs), developed for processing sequential data, allowed them to maintain memory of past inputs. In the 2000s, Support Vector Machines (SVMs) gained more popularity, surpassing neural networks as they were seen as computationally expensive and hard to train efficiently. In the 2010s, breakthroughs in deep learning and deep neural networks reignited interest and the use of GPU for parallel processing facilitated the

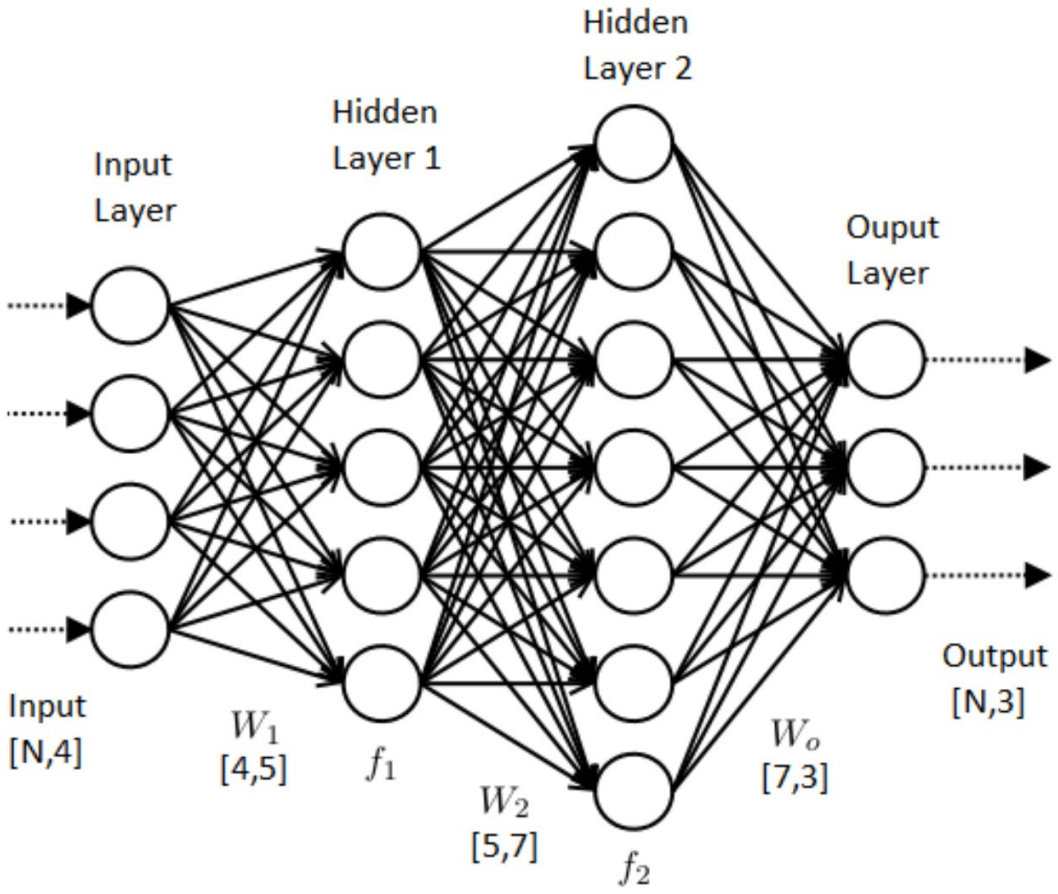

Figure 1. Layers in Neural Networks.

training of deeper architectures. Geoffrey Hinton success in the ImageNet Large Scale Visual Recognition Challenge in 2012 with a deep neural network marked a big impact in the industry, showing the power of deep learning for complex tasks. In 2014, Ian Goodfellow introduced Generative Adversarial Networks, a class of neural networks for generative tasks. GANs consist of a generator and a discriminator trained simultaneously through adversarial training. GANs have been successful in generating realistic images and data. For the present to this day, deep learning, powered by neural networks, is now widely applied, as well as ongoing research focuses on addressing challenges such as interpretability, robustness, and efficiency in neural network models to keep having advancements in this area. N/A (2020)

1 FUNDAMENTAL CONCEPTS

Biologically speaking, a neuron is where nerve cells pass around electrical pulses with the purpose of passing down information in the brain. Artificially speaking, neural networks represent a way of organizing a large number of simple calculations so that they can be executed in parallel. This calculations are processed by nodes (in the most simple way), also called artificial neurons.

1.1 Layers

Usually, the neurons in artificial neuron systems have the units arranged in layers as shown in Figure 1 and These networks have the input layer at the bottom, a hidden layer in the middle, and output layer at the top. The hidden layer is called that way because from what you see in the network view, it's pure inputs and outputs that are possible to monitor, hidden workings are not visible. A.D.Dongare (2012)

1.2 Activation functions

For artificial neurons, each connection has an association with a real value called weight and each neuron has an activation value. The most usual algorithm for activating an artificial neuron, j, given a set of inputs, subscripted by i, and the set of weights, w_{ij}, works as follows: Need to find the quantity, net_j, the total input to neuron j, this formula is used: When the activation value of neuron i, oi, times the weight w_{ij} is positive, then unit i serves to activate unit j. On the other hand, when the value of unit i times the weight w_{ij} is negative, the unit i serves to inhibit unit j. The activation value of neuron, j, is given by some function, f(net_j). The function, f, may be called an activation function, transfer function, or squashing function. One simple activation function is simply to let the sum of the inputs, net_j, be the activation value of the neuron. A second common activation function is to test if net_j is greater than some threshold (minimum) value, and if it is, the neuron turns on (usually with an activation value of +1), otherwise it stays off (usually with the activation value of 0). The neural network here computes the exclusive-or function and it uses the activation function, $1/(1+e-net_j)$. While this function only reaches 0 and 1 at -infinite and infinite respectively, when the outputs are close enough to 0 or 1 they are counted as being the same as 0 or 1. There are many other possible activation functions that can be used.

1.3 Learning algorithms

Learning in artificial neural systems is accomplished by modifying the values of the weights connecting the neurons and sometimes by adding extra neurons and weights. Neural networks learn by adjusting their parameters based on input data and associated output, a process known as training. The most powerful and used algorithm is the back-propagation. Some of the commonly used are:

Supervised Learning trained on a labeled dataset, where each input is associated with a corresponding output, the goal is for the network to learn a mapping from inputs to outputs, minimizing the difference between predicted and actual outputs.

Unsupervised Learning training a neural network on unlabeled data, allowing the network to discover patterns or structures within the data.

Reinforcement Learning involves an agent interacting with an environment and learning to make decisions by receiving feedback in the form of rewards or punishments.

Semi-Supervised Learning is a combination of supervised and unsupervised learning, where the network is trained on a dataset that contains both labeled and unlabeled examples.

Transfer Learning involves training a neural network on one task and then using the learned knowledge to improve performance on a different but related task.

Adversarial Training involves training a neural network with adversarial examples—input data specifically crafted to mislead the model.

Online Learning or incremental learning involves updating the model continuously as new data becomes available.

Understanding these learning algorithms is crucial for designing and training effective neural networks for various tasks. The choice of the learning algorithm depends on the nature of the data, the task at hand, and the available resources. Camargo (1992)

2 MAJOR ARCHITECTURES

Neural networks have been around for over 70 years, but recent designs and powerful graphics processing units (GPUs) have made a bigger impact in artificial intelligence. Deep learning isn't just one method, it's a group of ways and structures that can be used for many different problems. This section approaches five architecture designs from the last 20 years. Long short-term memory (LSTM) and convolutional neural networks (CNNs) are among the oldest in this list, but they are also very popular in different uses. Samaya Madhavan (2021)

2.1 Convolutional neural networks

A CNN is a multilayer neural network that was inspired how the animals use their vision, this is particularly useful in image-processing applications. The first CNN was created by Yann LeCun, back then, the architecture focused on handwritten character recognition. At first, it looks at simple things in the picture, like edges. Then, as it goes deeper, it combines these simple things to understand more complex parts of the picture.

2.2 Recurrent neural networks

The RNN is one of the foundational network architectures from which other deep learning architectures are built. Normally, in a regular system, information moves in one direction. But in this special one, it can also go back. This backward flow helps it remember what it saw before and deal with tasks that involve time or sequences. The main thing about RNN is this backward connection. It can happen from a hidden part, the output part, or a mix of both. This ability to remember and learn from the past makes it useful for problems that involve time, like predicting what happens next in a series of events.

2.3 LSTM networks

The LSTM have become popular in recent years, especially in things like smartphones. Unlike regular neural networks, LSTMs have a special memory cell that can remember things for a short or long time. This makes them great for remembering important stuff, not just the latest thing they learned. Inside this memory cell, there are three gates that control how information comes in or goes out. There's an input gate for new info, a forget gate for getting rid of old info, and an output gate for using the stored info. These gates have weights that are adjusted during training to make the system smarter.

2.4 GRU networks

Unlike the LSTM, the GRU has only two gates, removing the output gate. These gates are called the update gate and the reset gate. The update gate decides how much of the old information to keep, and the reset gate decides how to mix the new input with the old information. If you set the reset gate to 1 and the update gate to 0, a GRU becomes a basic RNN. GRU is easier and faster to train than LSTM, and it runs more efficiently. But, with lots of data, LSTM can sometimes give better results because it can capture more complex patterns.

2.5 Self-organized maps

The Self-Organizing Map (SOM), also known as the Kohonen map. It's a special type of neural network that helps organize data without needing labels. Unlike regular neural networks, SOMs don't have a concept of error or backpropagation. Instead, each node in the network represents a group, and the network organizes input data into these groups. Here's how it works: Imagine each node has some characteristics (weights) that start randomly. When we get some data, we find the node whose characteristics are closest to our data. We call that node the "best matching unit" or BMU. Then, we update the characteristics of nearby nodes to be more like the data we have. This process repeats, and over time, the network organizes the data into groups. In simple terms, SOMs help to organize and group data in a smart way, without needing labeled examples.

OVERVIEW OF ARCHITECTURES

2.6 Training methods

2.6.1 Backpropagation

Is a supervised learning algorithm that involves propagating errors backward through the network to adjust weights and biases. It uses the chain rule of calculus to compute gradients. Essential for optimizing the network's parameters to minimize the difference between predicted and actual outputs.

2.6.2 Gradient descent

Is an optimization algorithm that iteratively moves towards the minimum of the cost function by adjusting model parameters in the direction of steepest descent. Used to find the optimal set of parameters during training.

Table 1. Comparison of architectures

Name	Structure	
FNN	Layers of interconnected nodes	Commonly u
CNN	Convolutional layers that apply filters	Object de
RNN	connections that enable feedback loops	sequential da
LSTM	memory cells that store and retrieve information	mod
Autoencoders	encoder that compresses input data into a lower-dimensional representation	data compr
GAN	comprise a generator and a discriminator trained simultaneously through adversarial training	image generat

2.6.3 Learning rate scheduling

Involves adjusting the learning rate during training, often reducing it over time to allow the model to converge more effectively. Prevents overshooting the minimum and aids convergence, especially in non-convex optimization landscapes.

Understanding these architectures and training methods is fundamental for effectively designing, training, and optimizing neural networks for various applications. Gurney (1997)

NEURAL NETWORKS IN SPEECH RECOGNITION

Several neural network architectures have been employed for speech recognition, each designed to capture and process the complex patterns in audio data. Here are a few notable ones:

Recurrent Neural Networks (RNNs): These networks maintain memory of past information, making them suitable for sequences like speech. RNNs can capture temporal dependencies in speech data.

Long Short-Term Memory (LSTM) Networks: An advanced version of RNNs, designed to better handle long-term dependencies. Effective in capturing nuances and patterns in speech over longer sequences.

Convolutional Neural Networks (CNNs): Well-known for image recognition, CNNs can also analyze patterns in spectrograms (visual representations of sound). Applied to learn features from audio data, especially useful in initial layers for extracting low-level features.

Deep Neural Networks (DNNs): Traditional neural networks with many layers. Stacking layers helps in learning hierarchical features, improving the model's ability to understand speech.

Connectionist Temporal Classification (CTC): A type of neural network designed for sequence labeling tasks. Suitable for predicting sequences of phonemes or words without the need for aligned data.

Transformer-based Models: Originally designed for natural language processing, transformers have also been adapted for speech tasks. They capture contextual information effectively, making them suitable for understanding the context of spoken words.

Remember, the choice of architecture often depends on the specifics of the speech recognition task, such as whether it's focused on phoneme recognition, word recognition, or other aspects. Additionally, hybrid models combining these architectures are common for achieving better performance. N/A (2023)

Challenges

When dealing with neural architectures we can come across a few challenges as in any area, such as overfitting, vanishing gradients and adversarial attacks. Overfitting, is when the trained data is too specific and too well recognized that the algorithm can't decide for itself when it's new information, capturing noise and details. Vanishing/Exploding Gradients can be seen when during training, gradients can become extremely small or large, hindering the learning process. Adversarial Attacks appear when certain inputs mislead from the objective of the algorithm. All of these challenges can be solved depending on the problem, for example in overfitting we could have some type diverse training data, in vanishing gradients batch normalization and in adversarial attacks, adversarial training. A.D.Dongare (2012)

Recent advances

Some of the recent advancements in the field, such as attention mechanisms, transfer learning, and novel architectures have an important part in recognizing more high level improvements in the near future. **Attention mechanisms** enable neural networks to focus on specific parts of the input, enhancing their ability to capture relevant information. Widely used in natural language processing tasks, image captioning, and transformer-based models like BERT and GPT. **Transfer learning** involves pretraining a neural network on a large dataset and fine-tuning it for a specific task, leveraging knowledge gained from the pretraining phase. Facilitates training effective models with limited task-specific data, leading to improved performance and faster convergence. The **Transformer architecture**, initially designed for natural language processing, has shown versatility in various domains by capturing long-range dependencies efficiently. This are used in machine translation, image recognition, and other tasks where capturing global dependencies is crucial. Some of other examples are **Neuroevolution** that involves using evolutionary algorithms to evolve neural network architectures and/or parameters. Applied in reinforcement learning, game playing, and optimization problems, providing an alternative to gradient-based methods.

Results

To fully understand the capability of this neural networks in a near future its fundamental to recognize its architectures and how they can be improved in a specific performance such as speech recognition. CNNs demonstrated high accuracy in extracting features from audio signals, making them effective for speech recognition in noisy environments. RNNs, with their ability to capture temporal dependencies, excelled in tasks involving continuous speech recognition and natural language understanding. LSTM networks showcased improved memory retention, enhancing their performance in long-duration speech recognition tasks, as an example, IBM has used LSTM methods. GRU networks, while faster to train, showed comparable results to LSTM in various scenarios, depending on the specific task.

CONCLUSION

In conclusion, the journey of neural networks from their early beginnings to today's advanced models is huge. In this research, we looked at different types of neural network designs, especially focusing on how they can make computers understand and process speech.

Looking back, we've seen the growth of neural networks from basic ideas in the 1950s to the powerful systems we have now. Technologies like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and smart structures like Long Short-Term Memory (LSTM) networks have made a huge impact, especially in recognizing and understanding spoken words.

We explored how these networks learn, understand, and remember things through various learning methods. The study of major architectures, including CNNs and RNNs, shed light on how they play crucial roles in making sense of speech.

As we move forward, newer designs like Transformers and GPT are on the horizon. We've compared these new ones with the older models to figure out which is best for speech recognition. This comparison helps us decide whether the new, fancy designs are better than the ones we've been using.

Looking at the bigger picture, the research shows that neural networks have become a big part of our lives, from talking to our phones to making our gadgets understand us better. The findings in this paper help guide the development of even better systems for understanding and processing speech, making our technology smarter and more useful in real-world situations.

REFERENCES

A.D.Dongare, R.R.Kharde, A. D. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*.

Camargo, F. A. (1992). Learning algorithms in neural networks. *Research Gates*.

Gurney, K. (1997). An introduction to neural networks. *University of Sheffield*.

N/A (2020). What are neural networks? *IBM*.
N/A (2023). Neural network models (supervised). *scikit learn*.
Samaya Madhavan, M. T. J. (2021). Deep learning architectures. *IBM Developer*.