International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 1004-1009 www.ijaem.net ISSN: 2395-5252

CNN Vs RNN

Prakhar verma, Shivam tiwari, Er.Shilpi khanna[3]

1Student of Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management Lucknow, Uttar Pradesh, India
2Student of Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management Lucknow, Uttar Pradesh, India
3Professor, Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management Lucknow, Uttar Pradesh, India

Date of Submission: 15-05-2023 Date of Acceptance: 30-05-2023

ABSTRACT: Artificial neural networks are a critical component of many advanced AI applications. Despite their complexity, understanding the different types of neural networks doesn't have to be difficult. There are significant variations between types of artificial neural networks, which are computing systems that imitate the workings of the human brain. Familiarizing oneself with these unique forms, and their subtle distinctions and diverse applications, can mean the difference between success and failure in implementing AI and machine learning.
Each type of artificial neural network in machine learning is tailored to perform specific tasks. To explore these tasks and the best strategies for completing them, this article will introduce two types of artificial neural networks: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs employ filters within convolutional layers to change data, while RNNs are predictive, using activation functions from previous data points in a sequence to create the next output in a series. Both types haveassociated key terms and are widely used in real-life applications, particularly in the field of computer vision.

KEYWORDS: automate, computerized, Smart, Billing, online orders.

I. INTRODUCTION

Deep neural networks (DNNs) have two main types of DNN architectures, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are widely used to handle various NLPtasks, image processing etc. [1] CNNs are known for extracting position-invariant features, while RNNs are suitable for modeling sequential units. The performance of CNNs and RNNs on various tasks often differs, and this study aims to provide The ability of CNNs and RNNs to process temporal information, or input that comes in sequences, is the main distinction betweenthem. RNNs are specifically designed for this purpose, whereas CNNs are not as effective in interpreting temporal information. Hence, CNNsand RNNs are utilized for different purposes, and the structures of the neural networks also differ to fit those different use cases [2]

While RNNs are predictive and reuse activation functions from earlier data points in the sequence to generate the next output in a series, CNNs use filters within convolutional layers to alter data [3].

Recurrent neural networks use previous data points in a sequence to make better predictions and are suitable for interpreting temporal or sequential information [4]. They take input and reuse activations from the previous data pointsin a sequence to generate output.

II. DISCUSSION

Convolutional Neural Networks

Image processing uses Convolutional Neural Networks (CNNs), a subset of Deep Learning algorithms.. They take an image as input, assign weights and biases to different parts of the image, and use activation functions to perform various tasks such as Image Recognition, Image Classification, Object and Face Detection, etc.

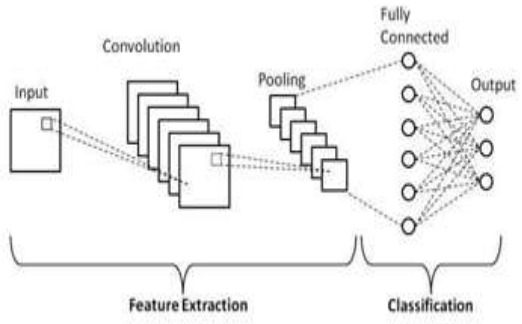
The basic principle of a CNN is to receive an input image, which can be either labelled or unlabeled. Depending on this, the algorithms can be classified into Supervised Algorithms (where images are labelled) or Unsupervised Algorithms (where images are not labelled). Theinput image is seen as an array of pixels, usuallyin the form of a matrix, with dimensions h x w x d (where h = Height, w = Width, d = Dimension). For instance, a 16 x 16 x 3 matrix array represents an RGB Image (3 denotes RGB values), while a 14 x 14 x 1 matrix array represents a grayscale image.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 1004-1009 www.ijaem.net ISSN: 2395-5252

Layers of Convolutional Neural Network

As shown in the above basic Architecture of a Convolutional Neural Network, a CNN Model consists of several layers through which the input images undergo pre-processing to get the output. These layers are essentially divided into two categories:

● The Input Layer, Convolution Layer, and Pooling Layer are the initial three layers that serve as a feature extraction tool to extract the fundamental features from the images supplied into the model.



● Using the output from the feature extraction layers, the final fully connected layer and output layer predicts a class for the image based on the features extracted.

In the first layer, called the Input Layer, the image is input into the Convolutional Neural Network Model as an array of matrices measuring 32 by 32by 3, where 3 indicates that the image is an RGBimage with an identical height and width of 32 pixels. The mathematical technique of convolution is then applied to these input images in the convolutional layer.

The kernel or filter, a second square matrix, is convolved with the input picture. We generate the output image known as the feature map, which contains details about the fundamental elements ofthe image, such as edges and lines, by sliding thekernel over the pixels of the input image one by one.

The Pooling layer, which aims to reduce the size of the feature map to save computational cost, comes after the Convolutional layer. Several pooling techniques, including Max Pooling, Average Pooling, and Sum Pooling, are used to accomplish this.

The Convolutional Neural Network Model's FullyConnected (FC) Layer is the penultimate layer inwhich the layers are flattened and fed.

The label prediction occurs here and is provided in the final Output Layer employing activation functions such the Sigmoid functions.

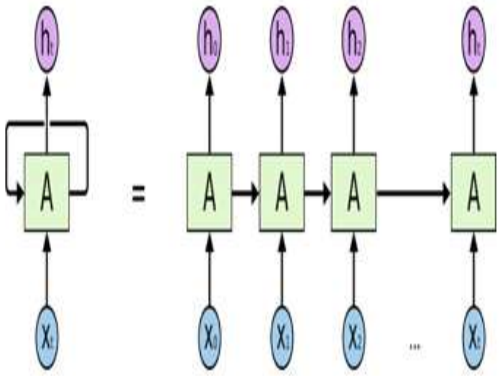
Where the CNNs Fall Short

Convolutional Neural Networks (CNNs) have a tiny drawback in that they do not perform well with a sequence of images (videos), and they struggle to understand the temporal information & blocks of text, while having so many useful applications in visual image data.

We need algorithms that can learn from both thepast and the future data in the series in order to deal with temporal or sequential data, such as sentences. Thankfully, Recurrent Neural Networks accomplish this.

Recurrent Neural Networks

Networks created specifically to comprehend temporal or sequential data are known as recurrent neural networks. RNNs improve their predictions by using additional data points in a sequence. In order to affect the output, they take in input and reuse the activations of earlier or later nodes in thesequence.



Source

Recurrent neural networks are able to accurately forecast what will happen next because of their internal memory, which allows them to recall important information like the input they received. As a result, they are the algorithm of choice for sequential data, including time series, voice, text, audio, video, and a variety of other types. Compared to other algorithms, recurrent neural networks can develop a far deeper grasp of a sequence and its environment.

Work of Recurrent Neural Networks

The foundation for understanding how recurrent neural networks work is the same as that

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 1004-1009 www.ijaem.net ISSN: 2395-5252

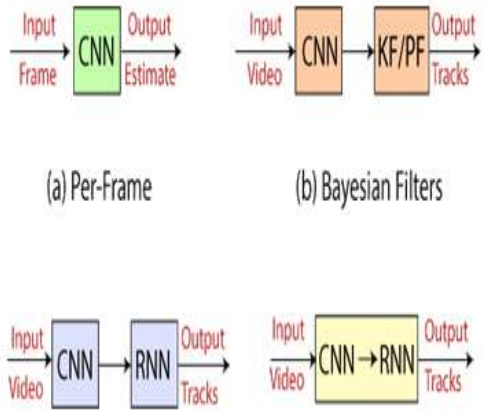
for convolutional neural networks, sometimes referred to as perceptron or simple feed-forward neural networks. Additionally, the output from the prior step is given as an input to the current step in recurrent neural networks. The fundamental distinction between the RNN and other Neural Networks is that, in most Neural Networks, the output is typically independent of the inputs and vice versa.

Source

Thus, the present and recent past are the two inputs of an RNN. This is significant because an RNN may perform tasks that other algorithms are unable to, because the data sequence conveys essential information about what will happen next. The Hidden state, which retains some informationabout a sequence, is the primary and mostsignificant characteristic of Recurrent Neural Networks. The calculations that have been made are all stored in the memory of the recurrent neural networks. The complexity of the parameters is decreased by applying the same parameters to all inputs and carrying out the same task on all inputs or hidden layers.

III. RESULTS

Following are the diagram shows the schematicrepresentation of CNN and RNN



The vector embedding of 2512 reports from the Stanford corpus served as the training data for the CNN model. We utilized a mini-batch size of 50, a dropout rate (p) of 0.5, and a l2 constraint (s) of 3. The Adadelta update mechanism was used in conjunction with stochastic gradient descent (SGD) over randomized mini-batches to update the network weights. The Xavier initialization approach was used to initialize weights randomly in order to roughly maintain the gradient scale throughout all layers.

We employed pre-trained word embedding to represent words in RNN-based models in a manner similar to CNN 300 hidden units (LSTMs, as mentioned in Section 3.2) made up each RNN. Our RNN- based models were trained using the SGD technique, with update direction computed using a batch size of 32 through the Adadelta update rule, similarly to the CNN models. These models underwent 400 iterations of validation training over 200 epochs. Each granularity, such as a word, sentence, or text, had a 300- dimension vector.

By choosing the optimum trade-off between the validation accuracy and the needed computational memory, we optimized the hyper parameters such as dropout rate, number of epochs, and batch size using the Stanford validation set (1000 reports).

IV. CLASSIFICATION PERFORMANCE

Precision, Recall, F1 value, and Area Under the Curve (AUC) are the measures we use to assess how well our models perform across all the data sets (Stanford, UPMC, Colorado Children, and Duke). We find the best cutoff threshold for the probability of the positive class by maximizing Precision(ti)+Recall(ti) for all thresholds (ti) between 0 and 1 in order to transform the predicted probability values of neural models to binary class labels. Table shows the classification outcomes from all approaches. performance comparison metrics. Boldface numbers indicate greater column-wise performance on a certain dataset.

	PE Positive/Negative				**PE Acute/Chronic**			
	P	**R**	**F1**	**AUC**	**P**	**R**	**F1**	**AUC**
Colorado Children test set								
PEFinder	0.69	0.79	0.73	N/A	0.66	0.87	0.72	N/A
SVM	0.98	**0.99**	**0.99**	0.91	**0.99**	**0.99**	**0.99**	0.94
Adaboost	**0.99**	0.98	**0.99**	0.97	**0.99**	0.99	**0.99**	0.98
CNN	**0.99**	0.95	0.96	0.97	**0.99**	0.98	0.98	**0.99**
HNN	**0.99**	**0.99**	**0.99**	**0.99**	0.56	0.74	0.60	0.81
A-HNN	0.87	0.80	0.83	0.98	0.60	0.87	0.66	0.77
DPA-HNN	0.80	0.80	0.80	0.93	0.71	0.87	0.77	0.98

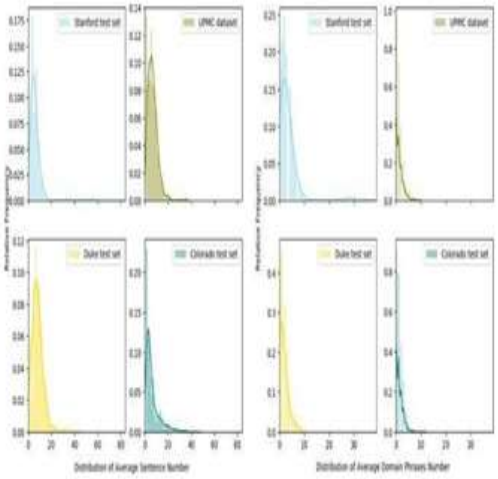
The Stanford test set, the DPA-HNN model showed the highest scores on all evaluation metrics for both PE Positive/Negative and PE

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 1004-1009 www.ijaem.net ISSN: 2395-5252

Acute/Chronic classifications, outperforming both the HNN and A-HNN models. The DPA- HNN model utilizes domain phrase attention which has been shown to enhance performance. Statistical analysis demonstrated that the improvements of the DPA-HNN model over HNN and A-HNN models were significant ($p < 0.05$). Overall, the neural network-based methods performed better than classic PEFinder, SVM, and Adaboost methods in terms of F1 and AUC scores on the Stanford test set.

On the UPMC dataset, DPA-HNN demonstrated the best precision scores for both tasks, while the CNN model had the highest AUC scores. On the Duke test set, DPA-HNN had the highest AUC scores for both tasks, while the CNN model showed the best precision and F1 scores. For the Colorado Children test set, the HNN model had the highest scores on all evaluation metrics for PE Positive/Negative classification, but did not perform as well on the PE classification.

Overall, the DPA-HNN model performs better on the Stanford test set than it did on the UPMC dataset and the Duke test set. The performance on the Colorado Children test set, however, is not the best, which is understandable given that DPA-HNN and other neural network-based approaches were trained on the Stanford dataset,which primarily consists of adult patients as opposed to the unique paediatric population of Colorado Children. Additional analyses showed that the distributions of the average number of sentences and domain phrases in a text varied among the external datasets (UPMC dataset, Duke test set, Colorado Children test set). Figure 4 and Table 1 show the distributionsand statistics. In the DPA-HNN paradigm we propose, DPs are crucial. The DPA-HNN model trained on the Stanford dataset may not perform equally well on Colorado data because, for instance, the average number of DPs in a document in the Colorado data is much lower than the average number of 3.5 in the Stanford test data, and the percentage of documents without DPs is much lower for the Colorado data than the Stanford test data. However, this dataset's average number of sentences per document is 6.2, which is rather close to Stanford's figure of 6.4. Since DPs are not used in the HNN model, It did well on the Colorado test set, however the DPA-HNN model struggled here since there weren't enough DPs.



	PE Positive/Negative				PE Acute/Chronic			
	P	**R**	**F1**	**AUC**	**P**	**R**	**F1**	**AUC**
Stanford test set								
PEFinder	0.87	0.90	0.89	N/A	0.91	0.91	0.91	N/A
SVM	0.96	0.96	0.95	0.85	0.93	**0.97**	0.95	0.93
Adaboost	0.98	0.98	0.98	0.95	0.97	**0.97**	0.97	0.92
CNN	0.92	0.97	0.95	**0.99**	0.91	0.91	0.91	0.99
HNN	0.94	0.96	0.95	0.98	0.92	**0.97**	0.94	0.95
A-HNN	**0.99**	0.96	0.97	0.98	0.97	0.96	0.97	0.99
DPA-HNN	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.97**	**0.98**	**0.99**
UPMC dataset								
PEFinder	**0.87**	**0.96**	**0.91**	N/A	**0.91**	**0.95**	**0.93**	N/A
SVM	0.71	0.70	0.58	0.83	0.49	0.70	0.57	0.86
Adaboost	0.83	0.84	0.83	0.90	0.84	0.83	0.82	0.91
CNN	0.76	0.95	0.85	**0.97**	0.82	0.92	0.87	**0.97**
HNN	0.82	0.74	0.77	0.90	0.88	0.73	0.75	0.88
A-HNN	0.82	0.74	0.77	0.90	0.88	0.72	0.75	0.88
DPA-HNN	**0.87**	0.87	0.87	0.95	**0.91**	0.90	0.90	0.94

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 1004-1009 www.ijaem.net ISSN: 2395-5252

Duke test set

PEFinder	0.84	0.99	0.90	N/A	0.86	**0.99**	0.92	N/A
SVM	0.95	**0.98**	0.96	**0.94**	0.96	0.98	0.97	0.95
Adaboost	0.97	0.96	**0.97**	0.85	0.97	0.97	0.97	0.77
CNN	**0.98**	0.97	**0.97**	0.90	**0.98**	0.98	**0.98**	0.91
HNN	0.93	0.79	0.85	0.92	0.95	0.74	0.81	0.91
A-HNN	0.90	0.83	0.86	0.91	0.89	0.79	0.83	0.88
DPA-HNN	0.94	0.81	0.86	**0.94**	0.90	0.95	0.92	**0.99**

The distribution of the typical number of sentences in a document is shown on the left, while the distribution of the typical number of domain words is shown on the right.
Statistics showing the typical amount of sentences or document pages across four datasets.

We can also see from the data that, on average, the PE Acute/Chronic classification task has lower evaluation scores than the PE Positive/Negative classification job, indicating that the former work is more difficult than the latter.

V. QUALITATIVE ANALYSIS

The impact of input words on the decision output has been visualized in order to better comprehend the data that a deep learning model in a natural language challenge is utilizing to generate its conclusions. . For the CNN model, we used one such method called sensitivity analysis, which takes the partial derivative of theloss function with respect to each input variable.We take the L1 norm of the vector of partial derivatives to obtain an importance score for a certain word because our input variables in this case represent word vectors. The heat maps show the results of input sensitivity analysis for two examples—one positive and one negative—where the CNN model properly predicted from the Stanford test set. The text of a report that is favourable for both of the two prediction classes is displayed in the result on the left. We can see in this case that the model places the most emphasis (dark color)on the first sentence which clearly states that there is no evidence of PE.

Similarly on the right is an example of a report that is negative for PE and all other prediction classes. We can observe that the model in this instance gives the first line, which unambiguously asserts that there is no evidence of PE, the most emphasis (dark color). From these qualitative results we note that the network is able to parse through large sequences of text to focus on phrases that are relevant to classification simply from the document level annotation.

CONCLUSION

Thus, we have learned the fundamentals of both CNN and RNN, as well as the differences between the two most common types of neural networks, convolutional neural networks and recurrent neural networks, in this article about the differences between them. We have also summarized a brief comparison between the twoof them with their applications in the real world.Since CNNs are built to handle images and RNNs are built to handle text, they perform faster than RNNs. RNNs are capable of handlingimages, but they still struggle to distinguish between contrasting characteristics that are close together. RNNs struggle to determine which feature to display first when given an image of a face with eyes, nose, and mouth. CNNs use a grid of points, and an algorithm can be used totrain them to recognize patterns and forms.

RNNs are slower than CNNs because they require more computation power, yet CNNs are superior to RNNs at sorting through images.

Our findings support the application of these techniques at scale in classifying free text imaging reports for a variety of use cases, including radiology patient prioritization, cohort generation for clinical research, eligibility screening for clinical trials, and evaluating imaging utilization. They also suggest the viability of CNNs and RNNs in automated classification of imaging text reports. These methods might also have an effect on other significant fields of research, like the large-scale labelling of data for creating computer vision models in medicine for autonomous image interpretation.

REFERENCES

[1]. Pado Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8:Multi-way classification of semantic relations between pairs of nominals. In Proceedings of SemEval. pages 94–99.
[2]. Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural computation 9(8):1735–1780.
[3]. Rafal Jozefowicz, Wojciech Zaremba, and

Ilya Sutskever. 2015. An empirical exploration of recur- rent network architectures. In Proceedings of ICML.pages 2342–2350.

[4]. Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In Proceedings of ACL.

[5]. Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of ICML. pages 1188– 1196.

[6]. Vu et al. 2016. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324.

[7]. Wen et al. 2016. Distributed representations of words and phrases and their compositionality. In Proceedings of NIPS. pages 3111–3119.

[8]. Adel and Schutze. 2017. Overview of the 2012 shared task on parsing the web. In Notes of the First Workshop on Syntactic Analysis of Non- Canonical Language (SANCL). volume 59.

[9]. Yin et al 2016. Recursive deep models for semantic compositionality over a sentiment tree- bank. In Proceedings EMNLP. volume 1631, page 1642.

[10]. Dauphin et al. 2016. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of EMNLP.pages 1422–1432.

[11]. Mikolov et al.2013. Combining recurrent and convolutional neural networks for relation classification. In Proceedings of NAACL HLT. pages 534–539.

[12]. Chung et al. 2014. Learning text representation using recurrent convolutional neural network with highway layers. SIGIR Workshop on Neural Information Retrieval .

[13]. Jozefowicz et al. 2015. Enhancing freebase question answering using textual evidence. CoRR abs/1603.00957.