Survey on Deep Learning Techniques in Sentiment Analysis

M.Poongothai,
Department of Information Technology,
Institute of Road and Transport Technology,
Erode, India.
poongothaiirtt@gmail.com

M.Sathyakala,
Department of Information Technology,
Institute of Road and Transport Technology,
Erode, India.
msathyakalairtt@gmail.com

Abstract: - Deep learning has emerged as a powerful technique for machine learning that learns multiple layers of data representations or characteristics and produces state-of-the-art results for prediction. Deep learning has also been used in sentiment analysis in recent years, in addition to the popularity of deep learning in many application domains. The paper offers a summary of deep learning techniques and then provides a quantitative survey of its latest applications in the study of sentiment classification.

Keywords: Deep learning, Sentiment classification.

1. Introduction

Sentiment analysis or opinion mining is a computational study of the beliefs, thoughts, attitudes, evaluations, and opinions of people towards entities such as goods, services, organisations, persons, situations, events, objects and their descriptions. Because of the rapid growth of social media, we have a massive amount of opinion data, such as reviews, forum discussions, blogs, microblogs, Twitter comments, and social networks in the digital form. Nowadays, without asking one's friends and family for opinions, one can purchase a consumer product, since there are many user feedback and product discussions on the Web in public forums. Nevertheless, it remains a matter of discovering and tracking opinion pages on the Web and distilling the information found in them. The proliferation of disparate sites is a daunting challenge. Usually, each site contains a huge volume of opinion text that in long blogs and forum postings, it is not always easily deciphered. The average human reader is going to have trouble recognizing the related sitess and the extraction and description of the opinions in them. Automated systems of sentiment analysis are thus needed.

Deep learning techniques are a modern upgrade to artificial neural networks that take advantage of inexpensive, abundant computing. They are concerned with building much larger and more complex neural networks, and very large datasets of labelled analogy data, such as image, text, audio, as well as video. On datasets that

have up to a few hundred elements, or columns, most machine learning algorithms work well. But when dealing with unstructured data, the ability to process large numbers of features makes deep learning very efficient. Via several "layers" of neural network algorithms, deep learning algorithms run data, each of which transfers a simplified representation of the data to the next layer. As it goes through each neural network layer, they increasingly learn more about the data. Early layers learn how to classify low-level characteristics, and subsequent layers integrate earlier layer characteristics into a more holistic representation. In several application areas, they have achieved state-of-the-art performance, ranging from computer vision and speech recognition with NLP. It has also recently become very common to apply deep learning to sentiment analysis. Second, this paper gives an Deep learning techniques summary, and then offers a detailed survey of deep learning-based sentiment analysis studies.

2. Deep learning algorithms

Deep learning network uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. The lower layers close to the data input learn simple features, while higher layers learn more complex features derived from lower layer features. The architecture forms a hierarchical and powerful feature representation. Figure 1 shows the feature hierarchy from the left (a lower layer) to the right (a higher layer) learned by deep learning in face image classification [1]. It shows that the learned image features grow in complexity, starting from blobs/edges, then noses/eyes/cheeks, to faces.

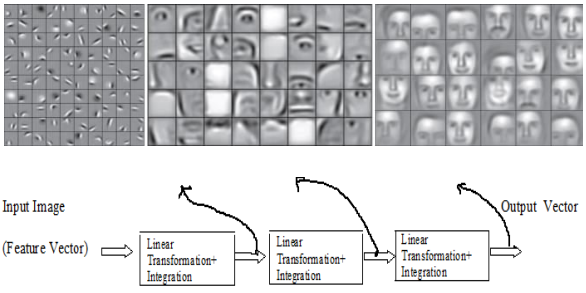

Fig. 1 Feature hierarchy using deep learning

2.1 Convolutional Neural Network (CNN)

The advancements in Computer Vision with Deep Learning have been constructed and perfected with time, primarily over one particular algorithm — a Convolutional Neural Network. A special form of feedforward neural network originally used in the field of computer vision is the Convolutional Neural networks (CNN). The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The human visual cortex is a visual mechanism in the animal brain. This influences the

architecture of CNN. Fig.2 shows the architecture of CNN. In small and overlapping subregions of the visual fields, which are called receptive fields, the visual cortex comprises several cells that are responsible for sensing light. Over the input space, these cells serve as local filters. CNN consists of different layers of convolution, each of which performs the role in the visual cortex that the cells process.

A CNN is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The role of the CNN is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

By implementing a local connectivity, CNN has a specific spatially local link Pattern of adjacent layers between neurons. Such a function is useful for NLP classification, in which we expect to find clear local clues about class membership, but these clues may appear in the input at different locations.

Convolutional and pooling layers allow the CNN to learn to find such local indicators, regardless of their positions [2]. The first part of a Convolutional Layer is called the Kernel/Filter, K. Fig. 3 shows the Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel.

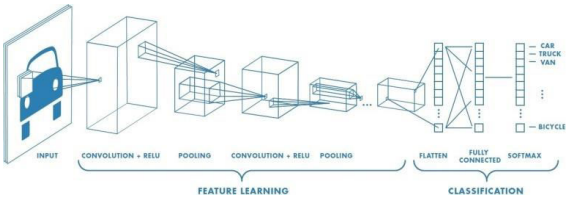

Fig. 2 Architecture of CNN

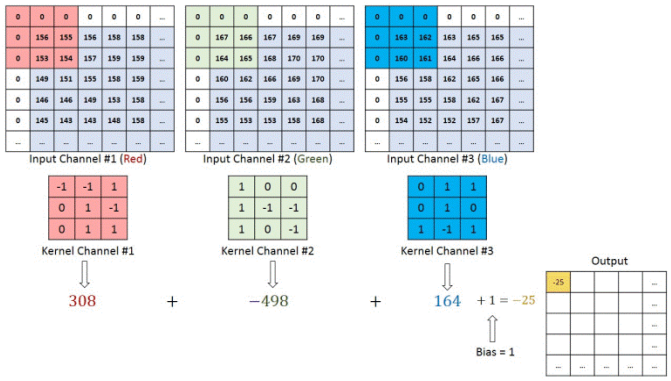

Fig. 3 Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel

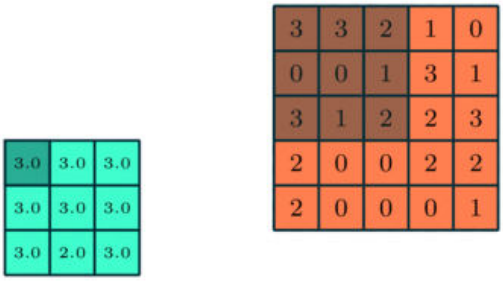

Fig 4. 3x3 pooling over 5x5 convolved feature.

The next layer is the Pooling layer, which is responsible for reducing (either by Max Pooling or Average Pooling) the spatial size of the Convolved Feature. This layer is useful for extracting dominant features. The converted input image is in suitable form for our Multi-Level Perceptron, then

flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

2.2 Recurrent Neural Networks (RNNs)

Recurrent neural network [3] is a class of neural networks whose connections between neurons form a directed cycle. Unlike feedforward neural networks, RNN can use its internal "memory" to process a sequence of inputs,

which makes it popular for processing sequential information. The "memory" means that RNN performs the same task for every element of a sequence with each output being dependent on all previous computations, which is like "remembering". information about what has been processed so far. Fig. 5 shows an example of an RNN. The graph is a folded sequence network with three-time steps. The number of time steps is determined by the length of input. For example, if the word sequence to be processed is a sentence of six words, the RNN would be unfolded into a neural network with six-time steps or layers. One layer corresponds to a word. RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers (ie., w1=w2=w3 and b1=b2=b3), thus reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer.

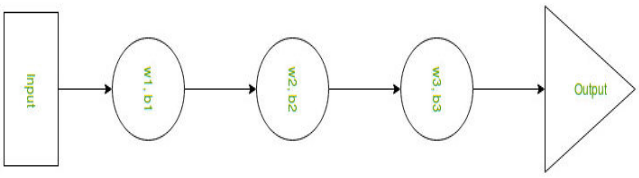

Fig. 5 RNN with 3 hidden layers

Formula for calculating current state h_t

$$h_t = f(h_{t-1}, x_t) \quad (1)$$

h_t -> current state
h_{t-1} -> previous state
x_t -> input state

Formula for applying Activation function(tanh)

$$h_t = \tanh (W_{hh}h_{t-1} + W_{xh}x_t) \quad (2)$$

w_{hh} -> weight at recurrent neuron
w_{xh} -> weight at input neuron

Formula for calculating output.

$$y_t = W_{hy}h_t \quad (3)$$

Y_t -> output
W_{hy} -> weight at output layer

The formulae (1-3) give the current state, activation function and output, respectively.

2.3 Long Short-Term Memory Networks (LSTMs)

LSTM or Long Short-Term Memory Networks [5] is a specific type of Recurrent Neural Network (RNN) that is very effective in dealing with long sequence data and learning long term dependencies. All RNNs have the form of a chain of repeating modules. In standard RNNs, this repeating module normally has a simple structure. However, the repeating module for LSTM is more complicated. Instead of having a single neural network layer, there are four layers interacting in a special way. Besides, it has two states: hidden state and cell state. Figure 7 shows an example of LSTM. At time step t, LSTM first decides what information to dump from the cell state. This decision is made by a sigmoid function/layer σ, called the "forget gate." The function takes ht − 1 (output from the previous hidden layer) and xt (current input), and outputs a number in [0, 1], where 1 means "completely keep" and 0 means "completely dump" in Equation (4).

$$f_t = \sigma\left(W^f x_t + U^f h_{t-1}\right) \quad (4).$$

A sigmoid function/layer, called the "input gate" as Equation (5), decides which values LSTM will update.

$$i_t = \sigma\left(W^i x_t + U^i h_{t-1}\right) \quad (5)$$

A tanh function/layer creates a vector of new candidate values ~ Ct , which will be added to the cell state

$$\sim Ct = \tanh\left(\left(W^n x_t + U^n h_{t-1}\right)\right) \quad (6)$$

The old cell state Ct − 1 into new cell state Ct as Equation (7).

$$C_t = f_t * C_{t-1} + i_t * \sim Ct : \quad (7)$$

LSTM first runs a sigmoid layer, which decides which parts of the cell state to output in Equation (8), called "output gate."

$$o_t = \sigma\left(W^o x_t + U^o h_{t-1}\right) \quad (8)$$

Then, LSTM puts the cell state through the tanh function and multiplies it by the output of the sigmoid gate, so that LSTM only outputs the parts it decides to as shown in (9).

$$h_t = o_t * \tanh(C_t) \quad (9)$$

As the time sequence grows in RNNs, it's possible for weights to grow beyond control or to vanish. To deal with the vanishing gradient problem [4] in training conventional RNNs, Long Short-Term Memory (LSTM) [5] was proposed to learn long-term dependency among longer time periods. In addition to input and output gates, forget gates are added in LSTM. They are often used for time series prediction, and hand-writing recognition. In this paper, we utilize LSTM in learning sentiment classifiers of short texts. In [6] , a neural network was used to learn document representation, with the consideration of sentence relationships. It first learns the sentence representation with CNN or LSTM from word embeddings. Then a GRU is utilized to adaptively encode semantics of sentences

and their inherent relations in document representations for sentiment classification.

In [7] LSTM has been utilized for Twitter sentiment classification by simulating the interactions of words during the compositional process. Multiplicative operations between word embeddings through gate structures are used to provide more flexibility and to produce better compositional results compared with the additive ones in simple RNN. Like bidirectional RNN, the unidirectional LSTM can be extended to a bidirectional LSTM (Graves & Schmidhuber, 2005) by allowing bidirectional connections in the hidden layer.

2.4 Stacked Auto-Encoders

In [2], domain adaptation problem for sentiment classification is studied. They proposed a deep learning system based on stacked DAE with sparse rectifier units, which can perform an unsupervised text feature/representation extraction using both labeled and unlabeled data. The features are highly beneficial for domain adaption of sentiment classifiers.Stacking of autoencoders in order to boost performance of deep networks. A key function of stacked autoencoders is unsupervised pre-training, layer by layer, as input is fed through. Once the first layer is pre-trained (neurons h1(1), h2(1), .., h4(1)), it can be used as an input of the next autoencoder. The final layer can deal with traditional supervised classification and the pretrained neural network can be fine-tuned using backpropagation.

3. Conclusions

Deep learning technique for machine learning that learns multiple layers of data representations or characteristics and produces state-of-the-art results for prediction. Deep learning has also been used in sentiment analysis in recent years, in addition to the popularity of deep learning in many application domains. The paper offers a summary of deep learning techniques and then provides a quantitative survey of its latest applications in the study of sentiment classification. Each technique has its unique functionality, applying the technique in combination gives a better result.

References

[1] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the International Conference on Machine Learning (ICML 2009).

[2] Glorot, X., Bordes, A., & Bengio, Y. (2011a). Deep sparse rectifier neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2011).

[3] Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14, 179–211.

[4] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Journal of Neural Computation, Vol. 9 No. 8, pp. 1735-1780, 1997

[6] Tang, D., Qin, B., & Liu, T. (2015a). Document modelling with gated recurrent neural network for sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015).

[7] Wang, X., Liu, Y., Sun, C., Wang, B., & Wang, X. (2015). Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2015).