IRIS CLASSIFICATION USING BPN AND IMDB SENTIMENT ANALYSIS CNN-RNN

1st Rakesh Pv
dept. of computer scince (MCA.)
christ deemed to be university
Bangalore, India
rakesh.pv@mca.christuniverisity.in

***Abstract*—This project paper refers to experiments towards the classification of Iris plants with back propagation neural networks (BPNN). The problem concerns the identification of Iris plant species on the basis of plant attribute measurements. The paper outlines background information concerning the problem, making reference to statistics and value constraints identified in the course of the project. A discussion concerning the experimental setup is included, describing the implementation specifics of the project, preparatory actions, and the experimental results. The results generated by the networks constructed are presented, with the results being discussed and compared towards the identification of the fittest architecture for the problem constrained by the data set. In conclusion, the fittest architecture is identified, and a justification concerning its selection is offered. Sentiment Analysis has been a classic field of research in Natural Language Processing, Text Analysis and Linguistics. It essentially attempts to identify, categorize and possibly quantify, the opinions expressed in a piece of text and determine the author's attitude toward a topic, product or situation. This has widespread application in Recommender systems for predicting the preferences of users and in e-commerce websites to analyse customer feedback reviews. Based on the sentiments extracted from the data, companies can better understand their customers and align their businesses accordingly.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A. IRIS CLASSIFICATION - BPN

This project paper is related to the use of back propagation neural networks (BPNN) towards the identification of iris plants on the basis of the following measurements: sepal length, sepal width, petal length, and petal width. There is a comparison of the fitness of neural networks with input data normalised by column, row, sigmoid, Also contained within the paper is an analysis of the performance results of back propagation neural networks with various numbers of hidden layer neurons, and differing number of cycles (epochs). The analysis of the performance of the neural networks is based on several criteria: incorrectly identified plants by training set (recall) and testing set (accuracy), specific error within incorrectly identified plants, overall data set error as tested, and class identification precision. The fittest network architecture identified used column normalization, 40000 cycles, 1 hidden layer with 9 hidden layer neurons, a step width of 0.15, a maximum nonpropagated error of 0.1, and a value of 1 for the number of update steps.

B. IMDB SENTIMENT CLASSIFICATION - CNN : RNN

Sentiment Analysis has been a classic field of research in Natural Language Processing, Text Analysis and Linguistics. It essentially attempts to identify, categorize and possibly quantify, the opinions expressed in a piece of text and determine the author's attitude toward a topic, product or situation. This has widespread application in Recommender systems for predicting the preferences of users and in e-commerce websites to analyse customer feedback reviews. Based on the sentiments extracted from the data, companies can better understand their customers and align their businesses accordingly. Before the advent of the Deep Learning era, Statistical methods and Machine Learning techniques found ample usage for Sentiment Analysis tasks. With the increase in the size of datasets and text corpora available on the internet, coupled with advancements in GPUs and computational power available for these tasks, Neural Networks have ushered in and vastly improved the state-of-the-art performance in various NLP tasks, and Sentiment Analysis remains no exception to this. Recurrent Neural Networks (RNN), Gated RNNs, Long-Short Term Memory networks (LSTM) and 1D ConvNets are some classic examples of neural architectures which have been successful in NLP tasks.

II. DATSET BACKGROUND

A. IRIS DATASET

This project makes use of the well known Iris dataset, which refers to 3 classes of 50 instances each, where each class refers to a type of Iris plant. The first of the classes is linearly distinguishable from the remaining two, with the second two not being linearly separable from each other. The 150 instances, which are equally separated between the 3 classes, contain the following four numeric attributes: sepal length and width, petal length and width. A sepal is a division in the calyx, which is the protective layer of the flower in bud, and a petal is the divisions of the flower in bloom. The minimum values for the raw data contained in the data set are as follows (measurements in centimetres): sepal length (4.3), sepal width (2.0), petal length (1.0), and petal width (0.1). The maximum values for the raw data contained in the data set are as follows (measurements in centimetres): sepal length

(7.9), sepal width (4.4), petal length (6.9), and petal width (2.5). In addition to these numeric attributes, each instance also includes an identifying class name, each of which is one of the following: Iris Setosa, Iris Versicolour, or Iris Virginica.

B. IMDB DATASET

This project uses the Large Movie Review Dataset which has been in-built with Keras. This dataset contains 25000 highly polar movie reviews for training, and another 25000 reviews for testing. It does not contain more than 30 reviews for any single movie, and also ensures there are equal number of positive and negative reviews in the both the training and test sets. Additionally, neutral reviews (those with rating 5/10 or 6/10) have been excluded. This dataset has been a benchmark for many Sentiment Analysis tasks, since it was first released in 2011.

III. MODELS

A. BPN - IRIS DATASET

This project uses various back propagation neural networks (BPNN). BPNN use a supervised learning mechanism, and are constructed from simple computational units referred to as neurons. Neurons are connected by weighted links that allow for communication of values. When a neuron's signal is transmitted, it is transmitted along all of the links that diverge from it. These signals terminate at the incoming connections with the other neurons in the network. The typical architecture for a BPNN is illustrated in Figure 1.

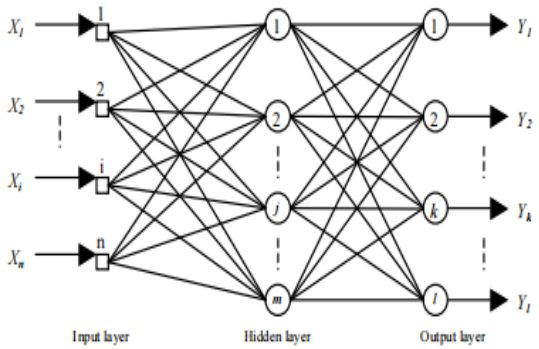
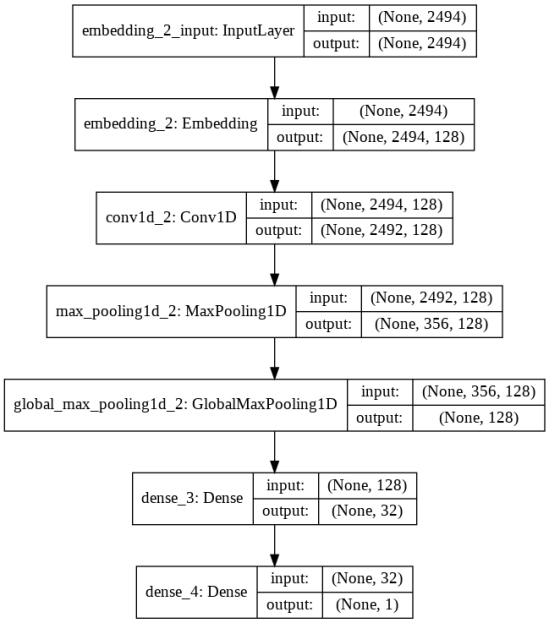


Figure 1. The architecture of a BPNN

In a BPNN, learning is initiated with the presentation of a training set to the network. The network generates an output pattern, and compares this output pattern with the expected result. If an error is observed, the weightings associated with the links between neurons are adjusted to reduce this error. The learning algorithm utilized has two stages. The first of these stages is when the training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output layer results are generated. Then, if the results differ from the expected, an error is calculated, and then transmitted backwards through the network to the input layer. It is during this process that the values for the weights are adjusted to reduce the error encountered. This mechanism is repeated until a terminating condition is achieved.

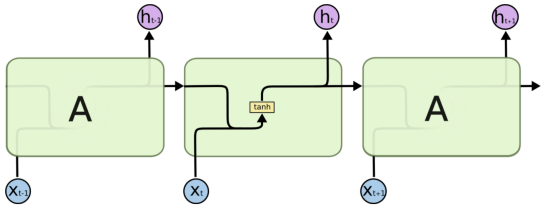
B. Convolution Neural Network - IMDB DATASET

The idea of Convolutional Networks has been quite common in Computer Vision. The use of convolutional filters to extract features and information from pixels of an image allows the model to identify edges, colour gradients, and even specific features of the image like positions of eyes nose (for face images). Apart from this, 1D Convolutional Neural Networks have also proven quite competitive with RNNs for NLP tasks. Given a sequential input, 1D CNNs are well able to recognize and extract local patterns in this sequence. Since the same input transformation is performed at every patch, a pattern learned at a certain position in the sequence can very easily later be recognized at a differnt position. Further, in comparison to RNNs, ConvNets in general are extremely cheap to train computationally - In the current project (built using Google Colaboratory with a GPU kernel), the LSTM model took more than 30 minutes to complete an epoch (during training) while the CNN model took hardly 9 seconds on average.

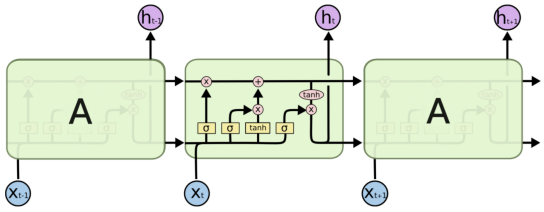


C. Recurrent Neural Network - IMDB DATASET

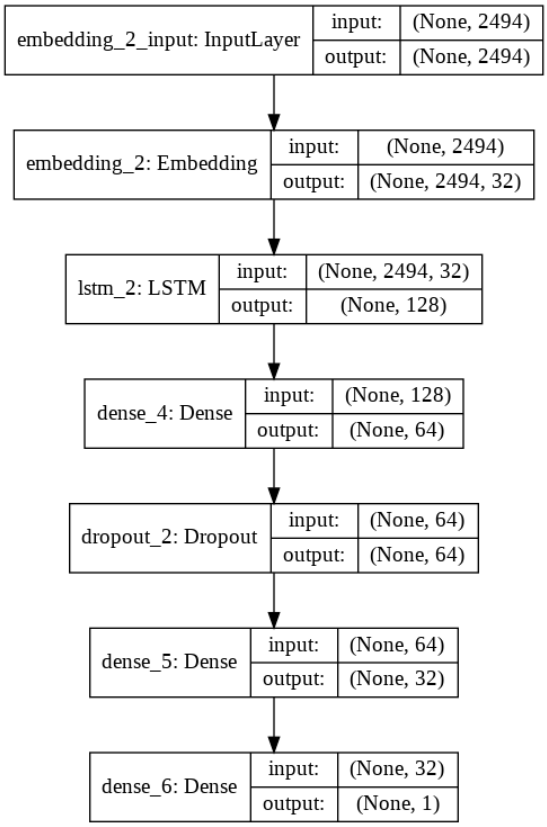
Recurrent Neural Networks are especially suited for sequential data (sequence of words in this case). Unlike the more common feed-forward neural networks, an RNN does not input an entire example in one go. Instead, it processes a sequence element-by-element, at each step incorporating new data with the information processed so far. This is quite similar to the way humans too process sentences - we read a sentence word-by-word in order, at each step processing a new word and incorporating it with the meaning of the words read so far.



LSTMs further improve upon these vanilla RNNs. Although theoretically RNNs are able to retain information over many time-steps ago, practically it becomes extremely difficult for simple RNNs to learn long-term dependencies, especially in extremely long sentences and paragraphs. LSTMs have been designed to have special mechanisms to allow past information to be reutilised at a later time. As a result, in practise, LSTMs are almost always preferable over vanilla RNNs. Here, I built an LSTM model using Keras Sequential API. A summary of the model and its layers is given below. The model was trained with a batch size of 64, using the Adam Optimizer.



A plot of the model and its layers While tuning the hyper-parameters, a Dropout layer was introduced as measure of regularization to minimize the overfitting of the model on the training dataset. A separate validation set (taken from the training data) was used to check the performance of the model during this phase. This model managed to achieve an accuracy of 85.91 when evaluated on the hidden test dataset (and 99.96 on the training dataset).



IV. METHODOLOGY

A. *BPN : Iris Datset Preprocessing:*

Before diving into preprocessing we have to split our data into training (80 percentage) and test data (20 percentage). We will use the training data (contain the classes for our iris species) to learn the model; and test data (contain only the features, without the classes) to measure the accuracy of our prediction model. We will encode categorical variables with LabelEncoder() since a prediction model cannot work with categorical variables.The second preprocessing technique is to scale our data with StandardScaler() as MLP (and gradient descent) is sensitive to un-normalized features. This helps us to speed up our optimization algorithm (gradient descent) and obtain a more accurate classifier

1) TRAIN THE MODEL: Every NN model must be trained with representative data before using. There are basically two types of training, supervised and unsupervised [10].The basic idea behind training is to pick up set of weights (often randomly), apply the inputs to the NN and check the output with the assigned weights. The computed result is compared to the actual value. The difference is used to update the weights of each layer using the generalized delta rule [6, 10]. This training algorithm is known as 'back propagation'. After several training epochs, when the error between the actual output and the computed output is less than a previously specified value, the NN is considered trained. Once trained, the NN can be used to process new data, classifying them according to its required knowledge.

Training = Minimize the loss function (categorical cross entropy here because oof the nature of the output)

Solver = Gradient Descent = Find the values of weights that minimize the loss function

B. IMBD SENTIMENT ANALYSIS : CNN-RNN

For the CNN architecture, we have trained the model for 8 epochs with a batch size of 128 as after that there was no more decrease in loss during the training phase of the architecture. For the same reason, we have trained LSTM network for 5 epochs with a batch size of 128. For the LSTM-CNN network, we have trained the network for 6 epoch with the same batch size as LSTM and CNN. We have used Adam optimizer [12] in order to minimize the loss function which was computed using Binary Cross-Entropy [13]. We have used Dropout [14] technique to avoid overfitting in the networknd does the expectations for the test information. The input of the CNN consists of 500 words per review which are fed into embedded layer to create a 100-dimensional vector for each word. We have used two convolutional layers in order to extract features and two pooling layers to provide translation invariance. We have used ReLU [9] activation function in convolutional layer. The output of convolutional and pooling layer are fed into a fully connected layer which in turns feed the extracted features into a hidden layer for classification. The output layer consists of one node with sigmoid activation function as it is a binary classification problem.

V. SIMULATION RESULTS

A. BPN IRIS DATSET

IRIS Plant	Total	Classified	Not Classified
Setosa	25	23	2
Versicolor	25	22	3
Virginnica	25	25	0

In 5000 iteration, out of 25 instances of Setosa class only 23 are classified, out of 25 instances of Versicolor class only 22 are classified and out of 25 instances of Virginica class 25 instances are classified. So accuracy rate = 96.66 percentage

B. IMBD DATSET

Evaluation Measure	CNN	LSTM
Accuracy	0.90	0.88
Recall	0.95	0.82
Specificity	0.84	0.90
Precision	0.87	0.90
F-Score	**0.91**	**0.86**

From the above table, we can observe that CNN has outperformed LSTM. The reason behind that is LSTM performs well in NLP task where the syntactic and semantic structure both is important. In the case of sentiment analysis, finding the positive and negative catchphrases are more important the syntax or semantic structure of the sentence. In fact, exploring the syntax of sentences sometimes results in a degradation in classification performance for sentimental analysis. So that is the main reason CNN has outperformed the other methods

VI. CONCLUSION

A. CNN - RNN

Sentiment analysis is becoming very important as the amount of online data increasing at a huge rate. For this reason, we need sentiment analysis on social media or online reviews for predicting and forecasting public opinion. We have found that CNN has performed better than LSTM because of the reason that syntax is not as important as positive or negative in sentiment classification. CNN has performed 2 percentage better than LSTM in terms of accuracy. CNN has also outperformed other state-of-the-art method on IMDb dataset. For future work, we have decided to use the convolutional neural network in other fields of natural language processing and evaluate the performance of used methods in those fields.

B. BPN

The Multi Layer Feed Forward Neural network gives us a satisfactory result, because it is able to classify the three different types of IRIS of 150 instances with just few errors for the other one. From the graphs we observe that Back propagation Algorithm gives the best accuracy .The no. of epochs required to train the neural network range from 500 to 50000 and the accuracy ranges from 83.33 percentage to 96.66 percentage. From the above results, graphs and discussion, it is concluded that Multi Layer Feed Forward Neural Network (MLFF) is faster in terms of learning speed and gave a good accuracy, i.e., has the best trade-off between speed and accuracy. So, for faster and accurate classification, Multi Layer Feed Forward Neural Networks can be used in many pattern classification problems.