**Professor: Dr. Leif Azzopardi**

**Dept. of Computer and Info Science**

**University of Strathclyde**

**Glasgow**



Advanced Machine Learning Assignment

CS 987

**Team Artemis**

MSc. Data Analytics

**Bennie Kristor Samuel (201875501)**[*bennie.samuel.2018@uni.strath.ac.uk*](mailto:bennie.samuel.2018@uni.strath.ac.uk)

**Parth Varangaonkar (201851829)***parth.varangaonkar.2018@uni.strath.ac.uk*

**Sourabh Mahajan (201855626)***sourabh.mahajan.2018@uni.strath.ac.uk* **Ullas Kakanadan (201852589)***ullas.kakanadan.2018@uni.strath.ac.uk*

**IMDB Dataset**

The imdb dataset has 50,000 movie reviews classified into equal sets of positive and negative reviews. This dataset is a sequence of binary classifications. The task at hand was to perform sentimental analysis which was implemented using word embedding and convolutional neural network using ‘*keras’* API and ‘*tensorflow’*. Convolutional layers were chosen for the model as they are quite robust when working on 2D problems. Another reason for choosing CNN was that CNNs train 50% to 60% faster than LSTMs.

**Architecture**

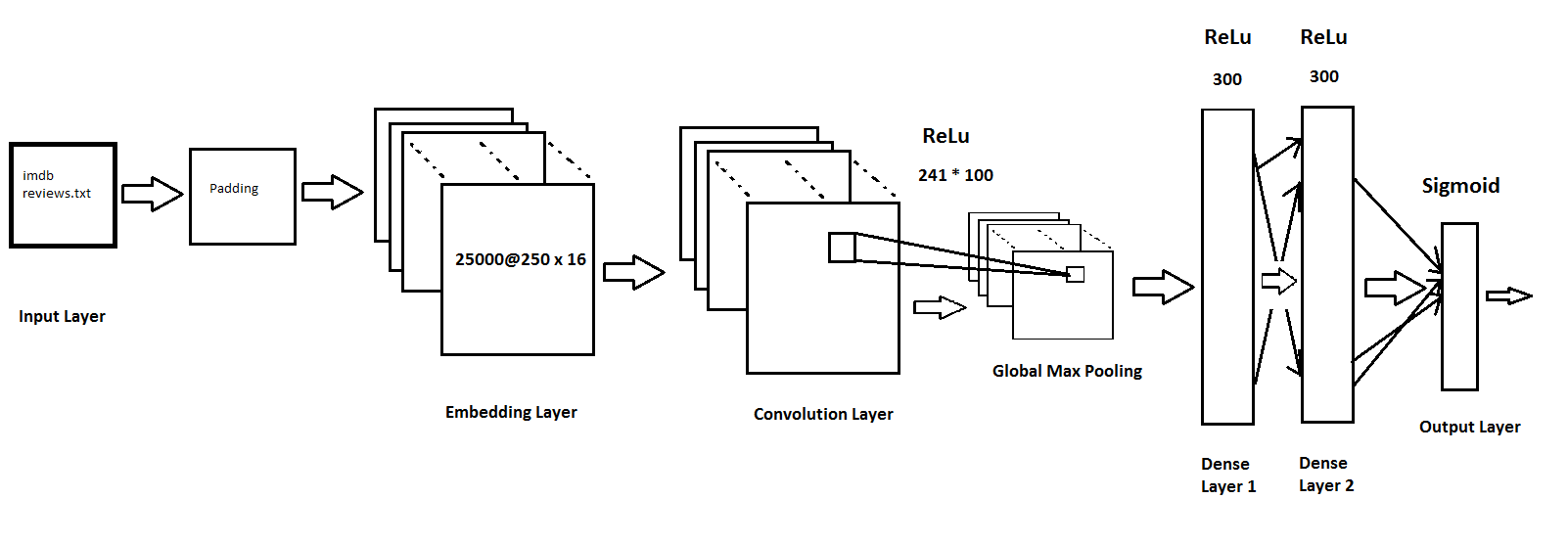


Figure 1: CNN Architecture for IMDB dataset

As shown in the above figure, at the first stage the network takes in the input where it is encoded into integers. Then the next phase is padding. This is done in order to design the CNN layers when the dimensions of the input volume need to be preserved in the output volume. After this, the output of the padding is given into the keras embedding layer which is used as the first hidden layer of the network. The embedding layer has a vocabulary of 5000 words, a vector space of 16 dimensions in which words will be embedded and each input document is restricted with 250 words. The output of this layer is a 2D vector with one vector assigned to each word in the input sequence (review text file). The next layer is the convolutional layer with a stride length of one. This layer uses the ReLu activation function with 100 filters. Global Max Pooling layer follows this layer. It simplifies/reduces the output information from the convolutional layer. It is then connected to two Dense layers of 300 neurons. The first Dense layer has a dropout percentage of 30%. Dropout was necessary to avoid overfitting the training data. These two layers also use the ReLu Activation function. The second Dense layer is connected to an output layer which uses the Sigmoidal Activation function. The loss function used in this output is *‘binary\_crossentropy’*. The cost function used to measure the mean squared error at every iteration was also considered during the process. The training set was split into validation set and training set in the ratio of 25:75. TensorFlow was also used to create logs and checkpoint files (.ckpt files) via keras.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Combination** | **Example** | **Parameters and Configuration** | **Training**  **accuracy** | **Validation accuracy** | **Testing accuracy** |
| **1** | **1** | **LR: 0.002, Epochs: 70, batch size:100, 2 hidden layers of 300 neurons each, 1 convolutional layer, dropout: 30%** | **0.8961** | **0.87792** | **0.87388** |
| 2 | 1 | LR: 0.002, Epochs: 10, batch size 25, hidden layer: 500 neurons, 2 convolutional layers, dropout: 30% | 0.9339 | 0.87024 | 0.859 |
| 1 | 2 | LR: 0.004, Epochs: 40, batch size:200, 2 hidden layers of 300 neurons each, 1 convolutional layer, dropout: 30% | 0.9384 | 0.87584 | 0.87048 |
| 2 | 2 | LR: 0.002, Epochs: 25, batch size 250, hidden layer: 500 neurons, 2 convolutional layers, dropout: 30% | 0.9029 | 0.86080 | 0.85732 |
| 1 | 3 | LR: 0.01, Epochs: 25, batch size:250, 2 hidden layers of 300 neurons each, 1 convolutional layer, dropout: 30% | 0.8888 | 0.87168 | 0.86492 |
| 2 | 3 | LR: 0.004, Epochs: 15, batch size 125, hidden layer: 500 neurons, 2 convolutional layers, dropout: 30% | 0.8729 | 0.86608 | 0.8598 |

*Table 1: combination table for networks*

There were two networks that were designed to deal with this particular problem. The First architecture has 217,001 parameters which has been mentioned above. The second has 247,101 parameters and 2 convolutional layers which use the ReLu Activation function. The output of the layer is connected to a hidden layer. This hidden layer with 500 neurons further produces an output for the final output layer which uses Sigmoidal Activation function.

3 variations for each of the 2 combinations(networks) were built. The batch size was varied in a manner that it was divisible by 25000 (Total input). The batch size has been varied between 25-250 to measure the performance of the networks. Similarly, the Epochs and learning rate were changed. It was observed that lower the learning rate, the accuracy of the model seemed to improve. The best combination was obtained in the first network with learning rate of 0.002 with 70 iterations and batch size being 100. This network gave the testing accuracy of 87.388% which was the highest.

**Fashion Dataset**

For the second data set fashion MNIST dataset was chosen which has images and needs to be classified into 10 categories. CNN work best when images are to be classified. The network works faster than LSTM networks and here, two networks are built. The dataset consists of 70,000 grey scale images of clothing articles. The images are of low resolution (28x28 pixels). The data-set is pre-cleaned dataset and doesn’t require much pre-processing. The dataset is imported from the keras library and split into training(85%) and testing(15%). Further, the training set set was split into validation and training in the ratio 25:75. The data is scaled between 0 and 1 to reduce the size before feeding it into the neural network. Furthermore, the input has to be made to a 3D array (28x28x1) as in the model we are using a 2D convolution layer. The architecture and working of the best network has been explained further

**Architecture**

The input layer takes in the data as a 3D array (28x28x1) and passes it to the next layer. From this layer the data is moved to a 2D-Conviolution layer where the data is flattened to 2D array. This layer has a filter size of 5x5 and 100 layers. The next layer is pooling layer which is used to reduce the dimensionality size. A 2D Max Pooling layer is used with a down scale factor of 2 (vertical and horizontal window) with a stride length of 2. Here, in a 2x2 window, the maximum value is chosen in the region represented by the filter, max of that region and creates a new output matrix where each element is the max of a region in the original input. The next layer is ‘flatten’ layer which converts the array into a series of vectors (1D vector). The layer flattens the input it receives and feeds it to the fully connected (FC) layers. The FC layers is nothing but the hidden layer and the output layer. The hidden layer has 256 neurons (chosen at random). The output of the hidden layer is passed to the output layer and the output layer has ten outputs as it has to classify the images into ten categories. All the layers use the rectified linear function unit (ReLu) for activation except the output layer which uses the SoftMax activation function.

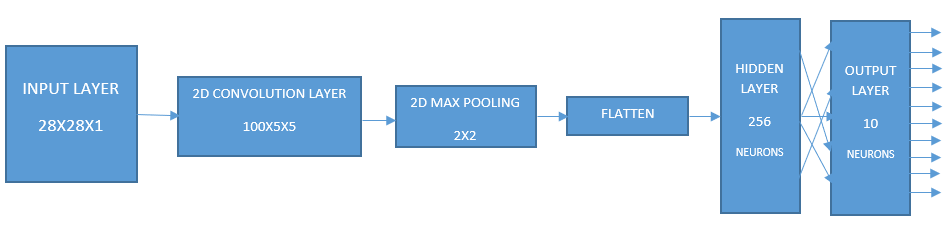


Figure : CNN architecture for Fashion MNIST

The model is designed using the *‘Keras’* API. The model is constructed as per the above architecture. The model has 3,691,826 parameters. Checkpoint is created to save the iteration with the highest validation accuracy along with logs of training via TensorBoard. For the measurement of loss *‘sparse\_categorical\_entropy’* is used*.* The network was tested with varying the learning rate, epoch and batch size (*Table 2*). The best iteration for each combination was recorded, the model was recalled and was tested on the testing set.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Combination** | **Example** | **Parameters and Configuration** | **Training accuracy** | **Validation accuracy** | **Testing accuracy** |
| 1 | 1 | LR:0.005 Epochs: 100, batch size:100, hidden layer of 256 neurons, seed:12345 | 0.9474 | 0.90767 | 0.9046 |
| 2 | 1 | LR:0.005 Epochs: 100, batch size:100, hidden layer of 600 neurons, seed:12346, dropout: 20% | 0.9838 | 0.9064 | 0.9007 |
| 1 | 2 | LR:0.009 Epochs: 50, batch size:200, hidden layer of 256 neurons, seed:123456 | 0.9519 | 0.9024 | 0.9046 |
| 2 | 2 | LR:0.009 Epochs: 50, batch size:250, hidden layer of 600 neurons, seed:123467, dropout: 20% | 0.9679 | 0.90227 | 0.8973 |
| **1** | **3** | **LR:0.007 Epochs: 25, batch size:250, hidden layer of 256 neurons, seed:1234567** | **0.9420** | **0.90567** | **0.9046** |
| 2 | 3 | LR:0.007 Epochs: 25, batch size:500, hidden layer of 600 neurons, seed:1234598, dropout: 20% | 0.9727 | 0.90620 | 0.9033 |

*Table 2: Combination table for Fashion MNIST*

Similar to the earlier mentioned dataset, this one was also tested on 3 combination of 2 networks each. The performance of the second network also mentioned *Table 2.*