# SMAI Project Presentation

Aaditya M Nair (201302161)
Atul Agarwal (201330188)
Parth Kolekar (201301143)

# Latent Dirichlet Allocation

- A Bayesian unsupervised learning model.
- Go through each document, and randomly assign each word in the document to one of the K topics.
- This random assignment already gives you both topic representations of all the documents and word distributions of all the topics (not very good ones).
- Go through each word w in each document d.
- For each topic t, compute two things: 1) p(topic t | document d)=words in document d that are currently assigned to topic t. 2)p(word w | topic t)= assignments to topic t over all documents that come from this word w.

# Latent Dirichlet Allocation

- Reassign w a new topic, where you choose topic t with probability p(topic t | document d) * p(word w | topic t).
- In essence, we're assuming that all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how documents are generated.
- After repeating the previous step a large number of times, we'll eventually reach a roughly steady state where our assignments are pretty good. Thus we get the estimates of the topic mixtures of each document (words assigned to each topic within that document) and the words associated to each topic (words assigned to each topic overall).

# Problems with LDA

- Problem with LDA is during online tasks an LDA takes a lot of time for inference speed.
- Hence need to learn DNN which approximates the LDA quite well and increases the computational speed drastically.

# Parameters of DNN

- For a particular document d, DNN takes it as a bag of words format, in the input layer.
- The output layer corresponds to topic mixture $\boldsymbol{\theta}$(d), for the document d. Number of output units = Number of topics.
- In DNN-2L, the number of hidden units is twice of the output units; in DNN-3L, the number of hidden units are three and two times of the output units.
- For the output layer, the softmax activation function is used, and the hyperbolic tangent function is used as the activation function for the other layers. The training employs the stochastic gradient descent (SGD) method with loss function as categorical cross entropy.

# Preparing Data

- We took the dataset Reuters-21578, and used the training and test data split as provided by the nltk toolkit. We were left with 6577 training documents and 2583 test documents.
- We removed stop words, numbers and applied a wordnet lemmatizer from the nltk toolkit.
- The documents with multiple ground truth labels were removed during the **classification** phase but still were used during the **training** phase.

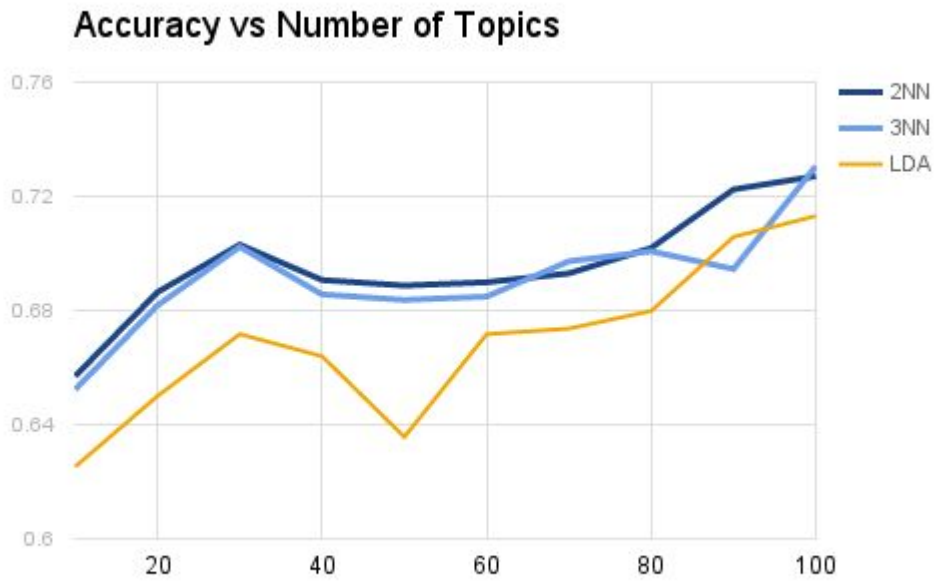
# Performing LDA

- The LDA was trained with the documents over 100 passes and the parent model generated was stored for varying number of topics.
- After we learn the topic mixture model for each document we applied a multi-class SVM to classify documents into different classes. We use the labels provided by the dataset itself as the ground truths.

# Training the Neural Network

- The previously trained LDA is used as the parent model.
- Both the NN models were trained using the bag-of-words representation of the document as input and the topic distribution of the document (as calculated by the LDA) as the output.
- Here again, SVM was used to classify the output of the NN into classes.

# Results


Accuracy vs Number of Topics
0.76
0.72
0.68
0.64
0.6
20
40
60
80
100
2NN
3NN
LDA

# Results

- Here we see that the NN model significantly outperforms the LDA model.
- This may be due to the fact that the NN is more generalised to the given dataset.

# Challenges Faced

1. There were several documents which were provided with multiple ground truth labels. It would have been very difficult to incorporate them into the classification and hence were ignored during that phase.
   Note that these documents were still used during training phase.
2. We had some documents with very small word count. They were supposed to be ignored if not for the already small dataset we were using.

# Conclusion

1. We can see that a simple DNN can approximate the LDA quite well, while the inference speed is quite a lot faster.
2. This indicates that transferring knowledge from a Bayesian model to a Neural Model is possible.

# References


- https://radimrehurek.com/gensim/ For LDA modelling.
- http://www.nltk.org/ For enhancing the datset and removing stop words.
- https://keras.io/ For building the DNN.
- http://scikit-learn.org/stable/ For building multi class SVM.