Machine Learning Project II

Andres Montero, Elias Poroma, Jonas Jäggi
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—We predict the mood of a tweet (positive or negative). We use GloVe word representation. We use logarithmic regression as a baseline model. We also use neural networks. We pre-process the data to get even better results (?) The best model is a neural network architecture which predicts N-Grams for groups of words.**

I. Introduction

II. Word Representation

Words are represented as embeddings (representation of each word as a vector with n elements)

A. Training our own word embeddings

We use matrix factorization to get an embedidding matrix for our words based on the co-occurece matrix of these words in the training tweets.

B. Using pre-trained word embeddings

We use an embedding matrix which was created by Standford university based on 2,6 Billion tweets in different languages. We filter out onl the words which occur at least 5 times in our training datast.

III. Baseline - Logarithmic regression

We take the ean of the word vectors for each tweet. We then perform

Table I
Logarithmic regression - Accuracy on a validation set of 10'000 tweets

GloVe Dimensions	Our GloVe	Pretrained GloVe
25	0.0	0.0
50	0.0	0.0
100	0.0	0.0
200	0.0	0.0

Higher embedding dimensions seem to deliver better results. Pretrained GloVe outperforms our GloVe by far.

IV. Convolutional Neural Networks

A. Network architecture

We have tried different network architecctures. We train in mini-batches of size 1024 tweets. We can increase the dimensions of the network layers (see table below). Higher dimensions deliver better results, but increase computation time significantly.

Table II
Validation accuracy for different network architectures. Time per minibatch serves as an indicator for network complexity.

Network	Accuracy	Time per minibatch [ms]
Network 1	0.0	0.0
Network 2	0.0	0.0
Network 3	0.0	0.0
Network 4 - 64 channels	0.0	0.0
Network 4 - 128 channels	0.0	0.0
Network 4 - 256 channels	0.0	0.0
Network 4 - 1024 channels	0.0	0.0

B. Measures against overfitting

We use L2 regularization for our network layerrs. In addition, wee can use dropout (explain dropout here).

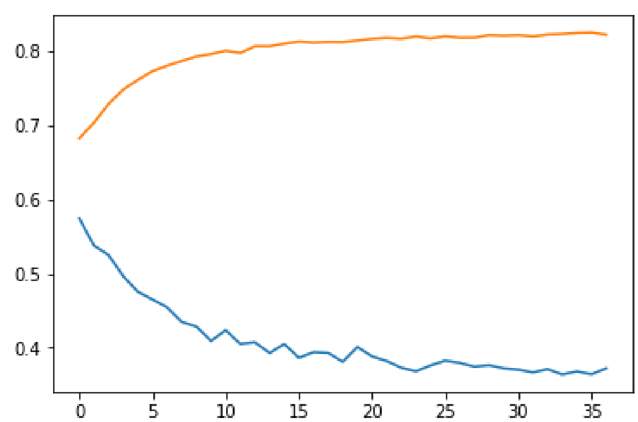

Figure 1. Evolution of Loss and Validation Accuracy

Dropout reduces the risk of overfitting, but in our case the tradeofff of having a less specific model is not worth it.

V. Text Preprocesing

We use different methods to preprocess text.

Table III
Text preprocessing - Accuracy on a vlidation set of 10'000 tweets, using the NGrams network architecture

Preprocessing method	Accuracy
Method 1	0.0
Method 2	0.0
Method 3	0.0
Method 4	0.0

VI. Results and Summary

Neural Networks offer a significative accuracy gain over logarithmic regression. Our implementations deliver consistent accuracy and are not much affected by overfitting. However, due to their computationa complexity special hardware is required for training them. Preprocessing the txt of the tweets allows us to increase accuracy even further (?). Achieving 100% accuracy is not possible - some tweets don't express a clear emotion, and it is therefore impossible, even for a human, to guess the mood of it. A detailed explanation of how to run the program is in the "README" file presented with the results.