**Prabowo or Jokowi stance based on netizens’ tweets classifier**

1. Damar Zaky 1301164460
2. Ki Ageng Satria Pamungkas 1301164422
3. Eldio Ruben M. Langkun 1301164576

IF-40-INT / ICM-39-INT Natural Language Processing

# Problem

What we are trying to solve here is to classify stance based on twitter tweets from Indonesian twitter, whether their tweets are likely to be pro Prabowo or pro Jokowi.

# Method

We use neural network to train the model, and based on 2000 in total of people tweets that was collected by scrapping using twitter api to stream tweets with filter containing #jokowi2periode and #2019gantipresiden. The corpus are labeled with 2 class which are “jokowi” and “prabowo”. The features we use is by using bag of words method as the input, so

1. Remove unnecessary words, remove string like URLs, then stem the words (example: “dibuka” -> “buka”, “dihilangkan”->”hilang”).
2. Iterate through the corpus, put words uniquely into an array/list of unique words (which is the bag of words)
3. Now use the bag of words as the input for neural network

To determine the input, for example, in the first data, we split the sentence into words, then we iterate through the bag of words and check whether any of words in the bag of words are in this sentence, if it exists in the sentence, it becomes 1, and if it doesn’t, it become 0.

Suppose, a bag of words: [“saya”, “adalah”, “suka”, “ada”, “menang”, “rumah”, “daging”, “pemerintah”, “makan”], and a sentence: “adi adalah orang yang suka makan daging”

The input for the neural network becomes: [0,1,0,0,1,1,1].

Now since the data is normalized, we can now use the normalized data as input to the neural network for training the model.

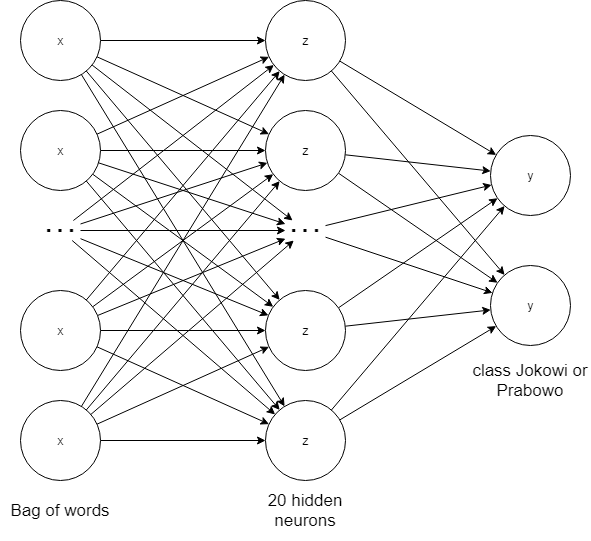


Figure 1: Our neural network model

The amount of input neurons is based on the bag of words length which in our case is 4070 neurons, while the hidden layer has 20 neurons, and we have 2 output neurons one is for jokowi and one is for prabowo.

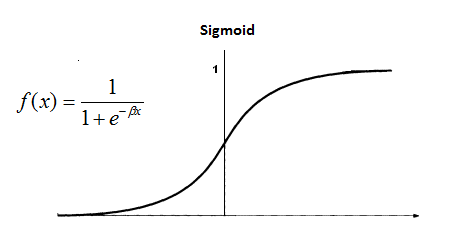


Figure 2: Sigmoid function

The activation function is sigmoid while the cost function we use is mean squared error. The neural network will then reduce the error calculated by cost function by using gradient descent. Our learning rate is 0.1. We iterate the training until 100000 epochs.

When the training phase is finished, we save the model as “model.json” so that we can use it later for testing.

# Test case

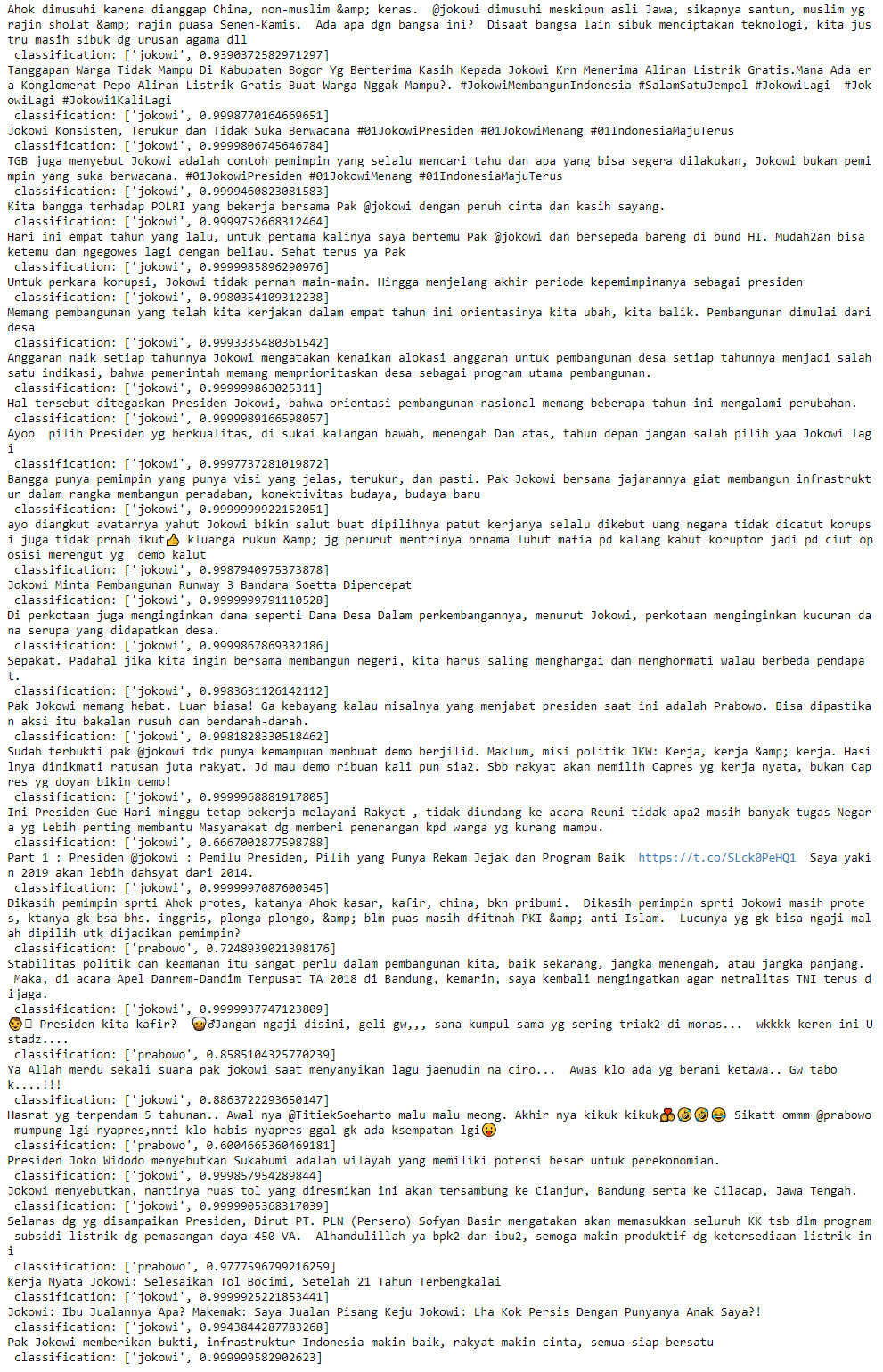


Figure 3: Jokowi test case



Figure 4: Prabowo test case

# Analysis

The jokowi test case gives accuracy of 83.8709677%

The Prabowo test case gives accuracy of 80%

Even though the corpus has 2000 data, the accuracy is not that great, especially when there is no strong keyword like “jokowi” or “prabowo” inside the text. We realize this is happening because the fact that on twitter, many Indonesian wrote Indonesian that are hard to stem, for example words like: “tdk”, “sy”, “jgn”,”gamau”,”jangann”,”jadii”, not to mention many typos’. And also, many tweets data are hashtags, or video/image based as well, so we realize that there aren’t many arguments on those data so that it’s hard to compare. Many stream data might also seem duplicated because of retweets.