

ANALYSIS OF THE EMOTIONAL BEHAVIOR OF USERS OF SITES BASED ON NEURAL NETWORKS

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА КОММЕНТАРИЕВ НА САЙТЕ

Author: **NIKITIN Danila**, e-mail: danila.n2015@yandex.ru

Faculty of Physics and Mathematics
Pskov State University

Head of the current research: **LOBAREV Dmitry Sergeevich**
Pskov State University

Purpose of work: Create a neural network capable of predicting the emotional coloring of a comment.

Keywords: machine learning , neural networks, python, recurrent neural network.

The structural basis of a neural network is a formal neuron. The input signal (x_1, x_2, \dots, x_n) is fed to the network input, an expression: $OUT = F(\sum_{i=1}^m x_i w_i + b)$ is formed at the output of the neuron, where

w_i are called the neuron weights

b -threshold

F -activation function

$OUT = F(\sum_{i=1}^m x_i w_i + b)$ - combined input neuron weights

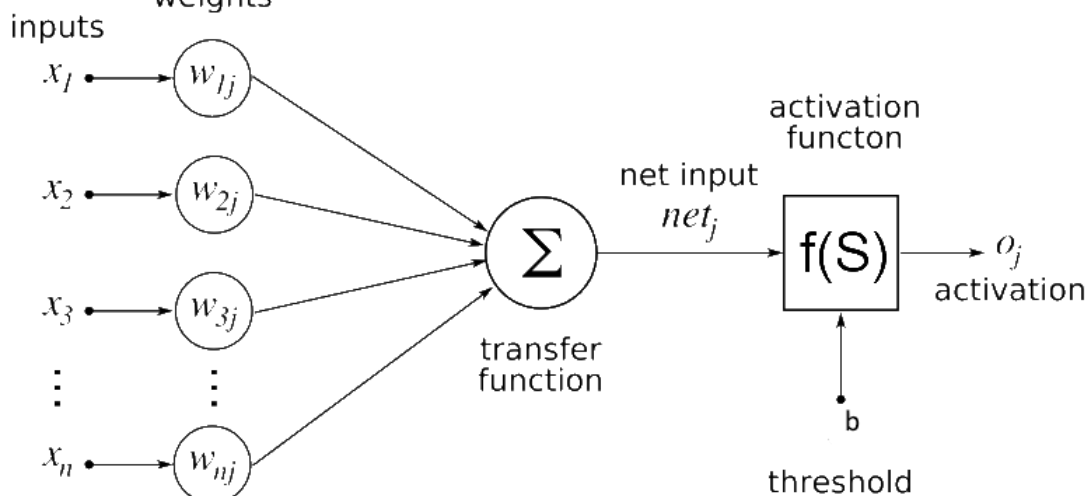


Figure 1. Structure of the formal neuron

Neural networks are connections of several neurons by synaptic connections. The set of weights of all neurons in the neural network are the parameters that determine its ability to process the signal and produce a result.

Educational data is a variety of types:

$\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ - Educational data

X_i - known input vector.

Y_i - corresponding known output vector

Learning a neural network is a process in which, using training data, find such values of network parameters, in which a **neural network using unknown data will produce the desired result.**

Let there be a training sample U of the form
 $U = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$, where
 X_i -known input vector.

Y_i - corresponding known output vector

It is also known a lot of real network outputs $\{Z_i\}$, where

Z_i is the real output of the network when the X_i input is applied to it.

The network loss for the real output $Z_i = \{z_1, z_2, \dots, z_m\}$ and the true output $Y_i = \{y_1, y_2, \dots, y_m\}$ is equal to the binary cross-entropy $BN(Z_i, Y_i)$

$$BN(Z_i, Y_i) = -\frac{1}{m} \sum_{j=1}^m y_j \log z_j + (1 - y_j) \log(1 - z_j)$$

The network loss E on the entire training set U is equal to:

$$E = \frac{1}{n} \sum_{i=0}^n BN(Z_i, Y_i)$$

During training, the network parameters P change in the direction opposite to the direction of the gradient of the loss function on its parameters:

$$\Delta P = -s \nabla E(P), \text{ where}$$

P -set of network parameters

E -network loss function of its parameters

S -learning speed

ΔP -change network parameters

$s \nabla E(P)$ is the gradient of the loss function by its parameters.

A **recurrent neural network** is a group of neural networks in which the output signal depends both on the current input signal and on its previous network states. In other words, this type of neural networks is able to “remember” its previous states and use them for signal processing, i.e. the output of the recurrent neural network depends on the order of the input vectors.

Due to this ability, recurrent networks are used to process natural language (machine translation of texts, generation of language, determination of emotional coloring of text, prediction of the next word).

The simplest recurrent neural network is the **Elman neural network**. Artificial neural network Elman, also known as Simple Recurrent Neural Network, consists of three layers - input (distribution) layer, hidden and output (processing) layers. In this case, the hidden layer has a feedback on itself. Figure 1 presents the scheme of the neural network of Elman (Figure 2).

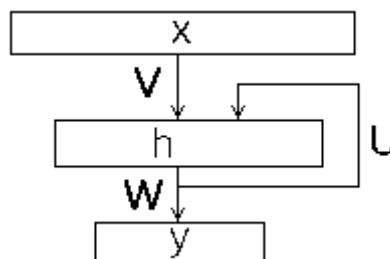


Figure 2. The simple Recurrent Neural Network

In contrast to the normal forward propagation neural network, the input vector of the recurrent network is not one vector, but the sequence of vectors, the vectors of the input image

in a given order are fed to the input, while the new state of the hidden layer depends on its previous states. Elman's neural network can be described by the following relations.

$$h_t = f(V \cdot x_t + U \cdot h_{t-1} + b1)$$

$$y_t = g(W \cdot h_t + b2)$$

where,

x_t - input vector,

h_t - the state of the hidden layer to enter x_t ($h(0)=0, h'(0)=0$),

$y(t)$ - network output for x_t input

U - distribution matrix weight matrix,

W - weight (square) matrix of feedbacks of the hidden layer,

$b1$ - displacement vector of the hidden layer,

V - output weight matrix,

$b2$ - output shift vector

f - hidden layer activation function

g - output layer activation function.

The disadvantage of the Elman network is a short memory. This disadvantage is solved by more advanced recurrent networks, such as LSTM networks and GRUs networks.

The **LSTM**(Long Short-Term Memory) - a special kind of the recurrent neural network architecture capable of learning long-term dependencies.

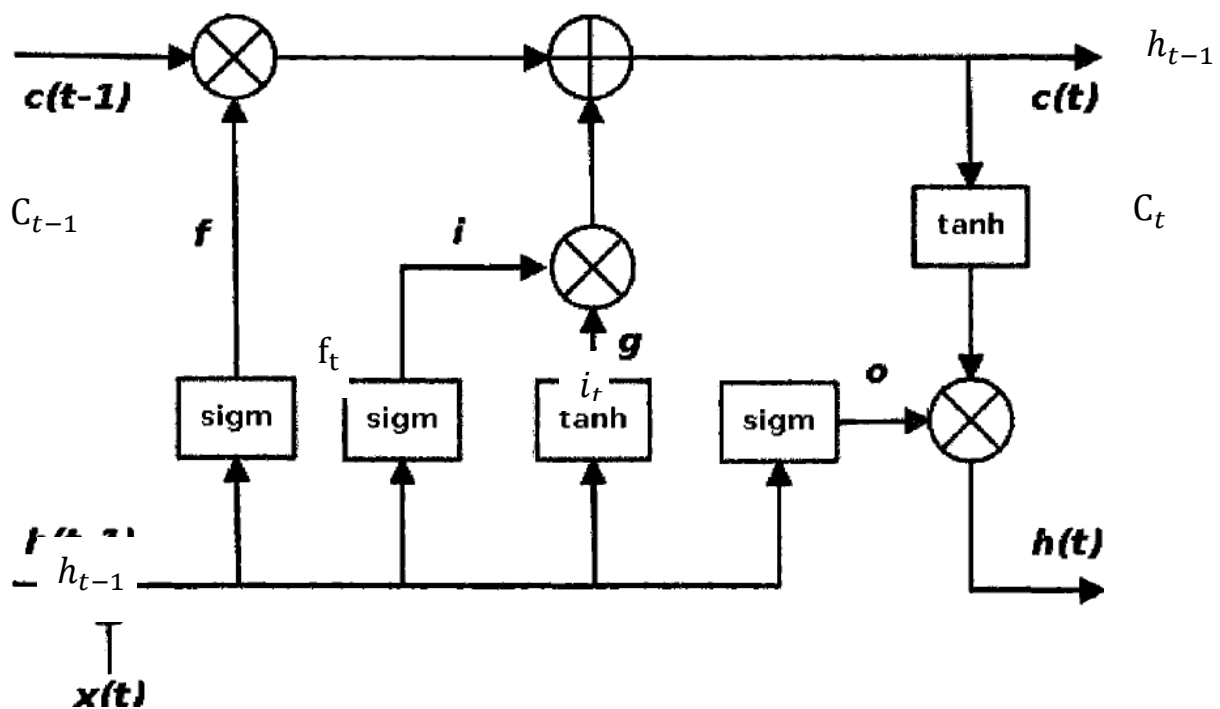


Figure 3. LSTM Layer Structure variant

The operation of the LSTM layer is described by the following formulas:

$$i = \sigma(W_i h_{t-1} + U_i x_t)$$

$$f = \sigma(W_f h_{t-1} + U_f x_t)$$

$$o = \sigma(W_o h_{t-1} + U_o x_t)$$

$$g = \tanh(W_g h_{t-1} + U_g x_t)$$

$$c_t = (c_{t-1} \otimes f) \oplus (g \otimes i)$$

$$h_t = \tanh(c_t) \otimes o$$

The GRU layer is an analogue of the LSTM layer, with a simpler architecture.

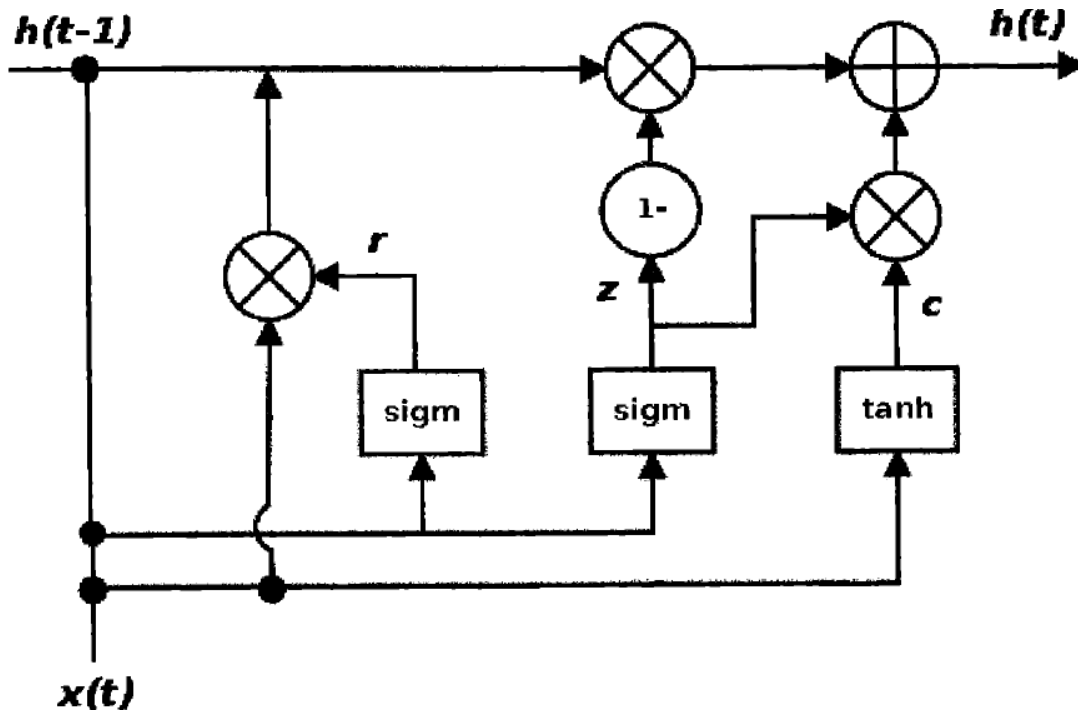


Figure 4. GRU Layer Structure variant

The work of the GRU layer is described by the following formulas:

$$z = \sigma(W_z h_{t-1} + U_z x_t)$$

$$r = \sigma(W_r h_{t-1} + U_r x_t)$$

$$c = \tanh(W_c (h_{t-1} \otimes r) + U_c x_t)$$

$$h_t = (z \otimes c) \oplus ((1 - z) \otimes h_{t-1})$$

For practical work, networks with LSTM and GRU layers were selected.

All kinds of user comments from the feedback site were chosen as training for the neural network data, which can be accessed via the following [link](#). The [data set](#) consists of 47 thousand lines, in which each user comment is assigned a score from 0 to 5. The data set was obtained by programmatically parsing the site in a python programming language. The output of the neural network is a number from 0 to 1. 0 is an extremely negative comment, and 1 is extremely positive comment. To evaluate neural networks was used. **For comparison of neural networks**, the proportion of correct answers (**accuracy**) achieved by the neural network on the test data is used.

The neural network input is a numerical sequence of fixed length equal to the maximum sentence length (maximum number of words in a sentence). To create an input vector, each word is assigned a number (an index depending on how often the word was found in phrases). If the length of the phrase is less than the input length signal then missing indexes are replaced by 0.

As a tool for learning this type of neural networks, the programming language Python and its library for machine learning Keras was used.

Python is an object-oriented programming language with dynamic typing and flexible syntax. It has a large number of high-performance scientific and mathematical libraries. Due to these features, at present this language is one of the main ones for creating learning models.

Keras is an open neural network library written in Python, which is aimed at efficient work with neural networks, and is designed to be compact, modular and expandable. It was

created as part of the ONEIROS research effort, and its main author is Francois Chollet, Google engineer.

This library contains numerous implementations of widely used building blocks of neural networks, such as layers, target and transfer functions, optimizers, and many tools to simplify work with images and text.

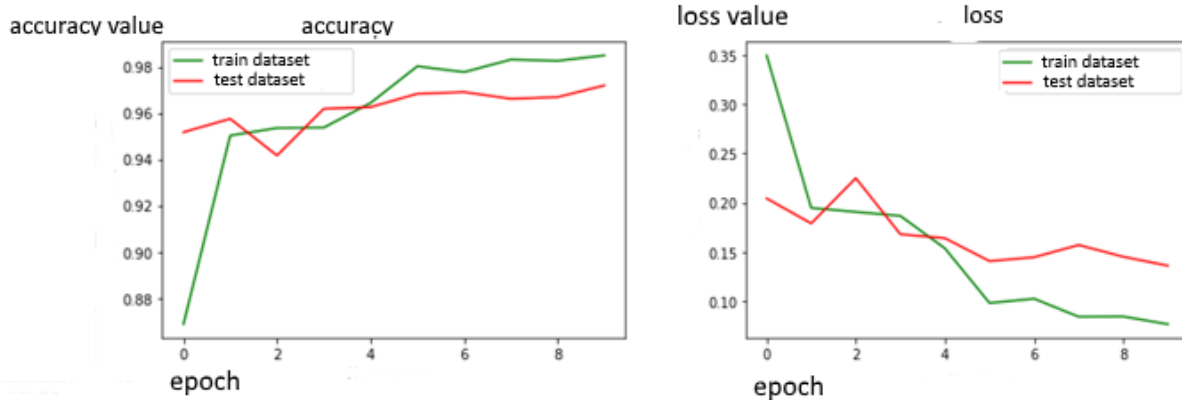


Figure 5. Learning neural network with LSTM-layer

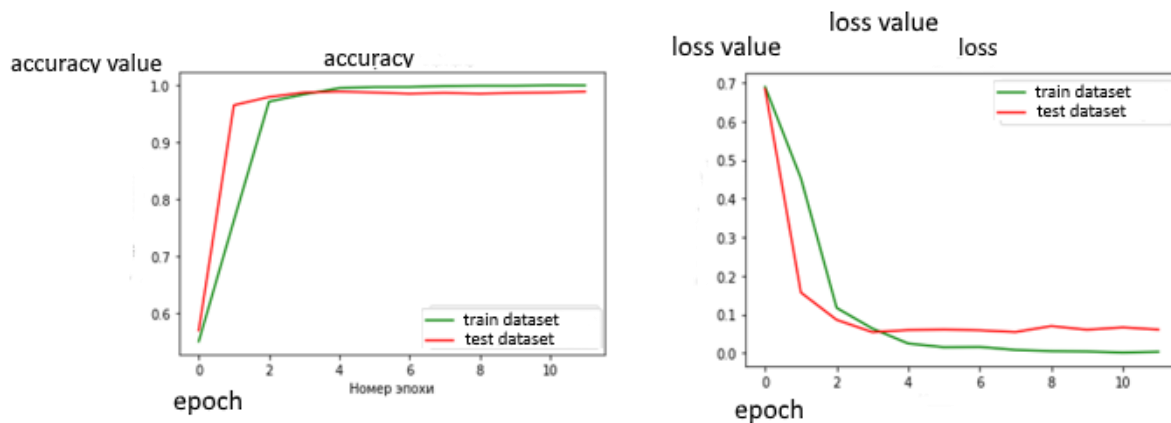


Figure 6. Learning neural network with GRU-layer

The relevance: The trained models can be used to analyze opinions from various sites and databases. You can select users' site statements, including social networks and blogs on emotional coloring, and assess people's attitudes to a particular phenomenon, this can be halfway through statistical analysis. filter negative messages in social networks.

Bibliography

- 1) Antonio D., Sujit P. Keras library is a tool for deep learning. Realization of neural networks with the help of Theano library and TensorFlow / lane. from ang. Slinkin A.A.- M., 2018.
- 2) Neural networks for beginners. Part 1 // habr.com URL: <https://habr.com/post/312450/> (access date: 03/06/2019).
- 3) Python for complex tasks: data science and machine learning. - SPb .: Peter -, 2018. - 576 p.: Ill. - (O'Reilly Bestsellers series).
- 4) WEB SCRAPING WITH THE HELP OF SCRAPY AND PYTHON 3 // <https://www.8host.com> URL: <https://www.8host.com/blog/web-scraping-s-pomoshhyu-scrapy-i-python-3/> (appeal date: 04/02/2019).
- 5) PROCESSING LANGUAGE DATA IN PYTHON 3 BY NLTK // <https://www.8host.com> URL: <https://www.8host.com/blog/obrabotka-yazykovyx-dannyx-v-python-3-s-pomoshhyu-nltk/> (appeal date: 04/05/2019).