Handwritten Characters Recognition Using Modification of Classic Rosenblatt's Perceptron Extended with Multiple S-layers

Pavlo Serdyuk[1], Vasyl Mosiiuk[2]

[1]Software Department, Lviv Polytechnic National University, UKRAINE, Lviv, S. Bandery street 28A, E-mail: pavlo.serdyuk@gmail.com

[2]Software Department, Lviv Polytechnic National University, UKRAINE, Lviv, S. Bandery street 28A, E-mail: vasyl.mosiiuk@gmail.com

***Abstract* – Presented structure and principles of perceptron with multiple sensor layers. Demonstrated how it recognize character image and vote for result.**

Key words – computer vision, CV, perceptron, OCR, machine learning, handwritten characters.

I. Introduction

Perceptron's modification, described in this article is extension of classic Rosenblatt's perceptron to work faster in computer vision processes, like image-to-pattern comparison and voting for recognition result.

II. Structure

The classic perceptron's structure demonstrated on the Fig. 1.

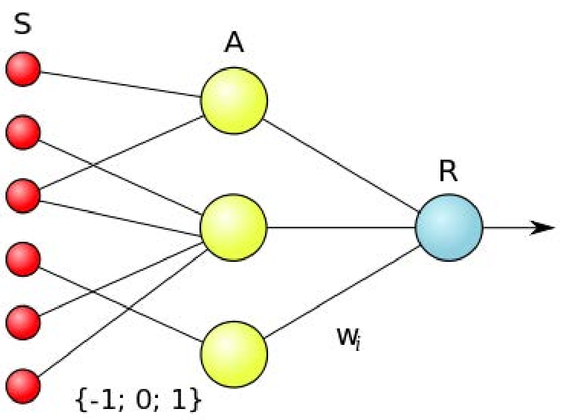

Fig. 1. Classic perceptron structure.

It consists of S (sensors), A (associative) and R (reaction) -layers. Input signals forwards on S elements, which are connected via S-A connections with A elements, which sums values of related S-A connections.

There are two types of S-A connections: positive and negative. In classic version A element is simple S-A connection values adder.

The matter thing is that connections, which builds in training process.

The R-layer calculate sum of products related A element value and weight matrix coefficient, which describes importance of value that A element.

There are kinds of perceptron with multiple count of A layer, with intermediate connections between (A-A connections). Structure of perceptron, described below, uses additionally multiple sensors layers too as showed on Fig. 2.

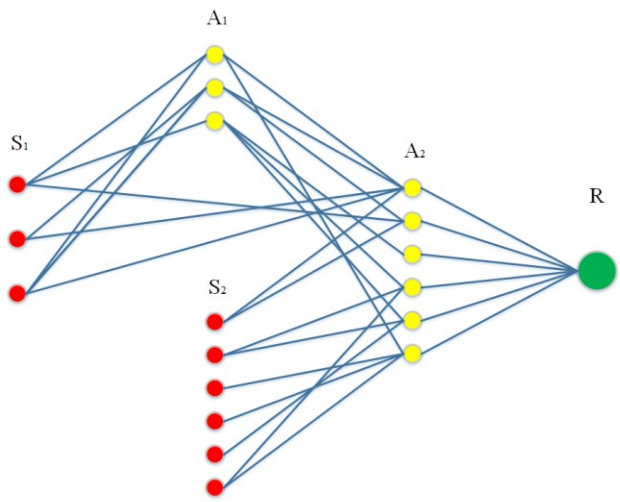

Fig. 2. Perceptron with multiple S-layers

Each of S-layers uses copy of input image, but different size and, so, detalization.

First S-layer operates with smallest image (i.e.7x7).

Each of A-layers have access to personal and previous S-layers, so it can recognize high and low - detailed parts of image.

There is threshold between A-A layers, to trunk impossible variant checks and increase a recognizing speed.

R-layer select proposition from latest A-layer with most maximum value.

Need to notice that S and R -layers are shared for whole system and A-layers are different for each character sample.

In classic edition of perceptron structure, we have count of S-element equal to count of pixels in input, which makes recognition process longer because system have to compute operations with each of them.

Additionally, each S-element may have multiple connections with A-layer, that make a count of calculations much bigger.

Proposed design decrease count of calculations of first stage, because uses tiny S_1-layer size and separates samples of characters which may be on image from others impossible variants.

Also, each next stage doing same work, but with subset of variants, retrieved on previous stage.

III. Recognition process

Depending on input image's size perceptron can to use different count of layers, but for best accuracy this count has to be bigger than 4.

So, in training process we used minimum image size equal 112x112, which descaled by 2 to 7x7, which took a 5 A-layers, related to each image sample.

Input image (input source) is matrix of 0 or 1 values, where 0 describes background and 1 – ink. This reason adds constraints on input: need to add image preprocessing (background removing, thresholding) to transform original image to binary without artefacts (dust, blinks, shadows, etc.).

A design allows to use input sources smaller than sources, used for training, in that case additional A-layers will be trunked (ignored) and R-layer takes value of A-layer of each sample from latest subset.

Internal structure of A-element is formula of N-input values with single result, as showed on Fig. 3.

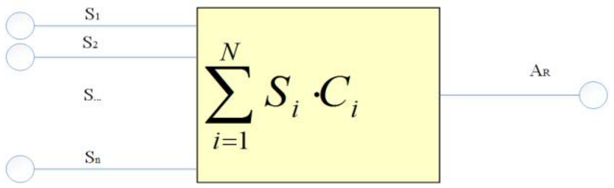

Fig. 3. Internal structure of A-element

C-multipliers are values, filled and fixed in perceptron learning process.

Input and output signals are values in range [0;1] where 1 means full matches with training sample, which was used in training process.

R-layer consists of single R-element which select A result with maximal value and related to that A-layer character value.

IV. Training

We used characters input blanks to get a set of images, related to known value. It's better because allow automatization all following training processes. Input blank part looks like as showed on Fig. 4.

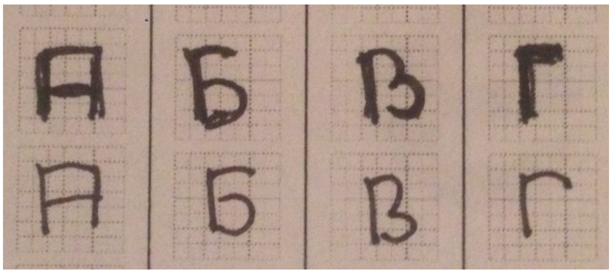

Fig. 4. Part of characters input blank

After training, some characters may have multiple samples and related A-layers, trained to recognize that sample.

It caused because there are different manners of handwriting (i.e. above picture demonstrates that people can write thin and thick characters): thin, thick, italic, etc. Any character, written in different manner would be have a different A-layer.

Optimization of enlargement count of A-layers is joining some of them in single, related to next level A-layer for multiple samples. Another is lookup A-layers duplications and joining them too.

Reason to print characters' sample blanks fields as grid is way to manipulate person, which write these characters.
Grid helps person to write each symbol regularly by columns and rows and, in result, decrease count of unique samples in perceptron.

Conclusion

Proposed modification of classic design of Rosenblatt's perceptron allows to decrease count of calculation and increase count of samples, their A-layers and increase accuracy of recognition.

Additionally, proposed training method automates training process, allow to fill blanks by volunteers without access to system.

References

[1] [F. Rosenblatt, "Principles of Neurodynamic" Spartan Books, Washington, DC, 1962.

[2] [M. L. Minsky, "Perceptrons" MIT Press, Cambridge, MA, 1969.

[3] [S. I. Gallant, "Perceptron-based learning algorithms" IEEE Transactions on Neural Networks, vol. 1, no. 2, pp. 179–191, 1990.

[4] [Y. Hongfeng, "Perceptron-Based Algorithms and Analysis" Spectrum Library, Concordia University, Canada, 1996.

[5] M. D. Estebon, "Perceptrons: An Associative Learning Network" Virginia Tech, 1997.

[6] J. E. Dayhoff, " Neural Network Architecturs: An introduction" Van Nostrand Reinhold, Australia, 19970.