Use of neural networks for managing the resources usage

Piotr Oleksy[1], Tomasz Wis[1], Włodzimierz Funika[1,2]

[1]AGH-UST, Faculty of Computer Science, Electronics and Telecommunication, Department of Computer Science, al. Mickiewicza 30, 30-059 Kraków, Poland
[2]AGH-UST, Academic Computer Centre CYFRONET AGH, ul. Nawojki 11, 30-950 Kraków, Poland
emails: funikaw@gmail.com, {poleksy,wis}@student.agh.edu.pl

Research motivation

- Creation of a system capable of provisioning resources in an optimal way
- Aspects addressed: upload/download speed, volume, RAM, location
- Constantly updating user profiles in order to predict their future behavior
- Use of artificial neural network to solve the problem.

Objectives

- Design a method supporting provisioning system resources
- Find a way to keep our system up-to-date with users
- Minimization of server maintenance costs

Our approach

After an amount of research into ANN-based resource management we decided to implement a perceptron-based recommender system. Each user is assigned a specific weight vector used to compute the happiness function between this user and hardware as:

$$H_{user}\left(X_{hardware}\right) = \Theta_{user}^{T} \cdot X_{hardware}$$

where *H_user* stands for a denoted user hypothesis function, *THETA_user* stands for a user weights vector generated by our system, *T* denotes transposition, while *X_hardware* stands for the denoted hardware weight vector generated from its properties.

As a perceptron learning algorithm we decided to use the *gradient descent with regularization*. In order to check the quality of learning we have used the learning curves checking method. This method consists of plotting a cost function of the number of training examples both for the training set and cross validation set of the generated data.

We defined the cost function as:

$$J = \sum\left((X - Y)^2\right)$$

where *X* stands for predicted happiness value, *Y* – real happiness value, which gives us a full batch training system.

Results

The system is performing very well on bigger numbers but rather poorly on smaller ones. This is probably due to the impact of the linearity of perceptron whereas our happiness function is more like a square root.

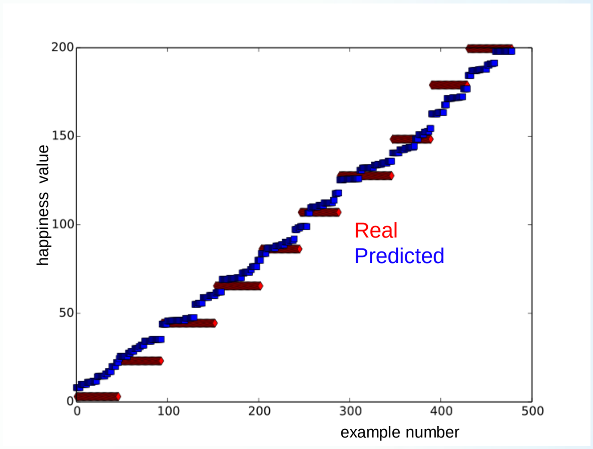
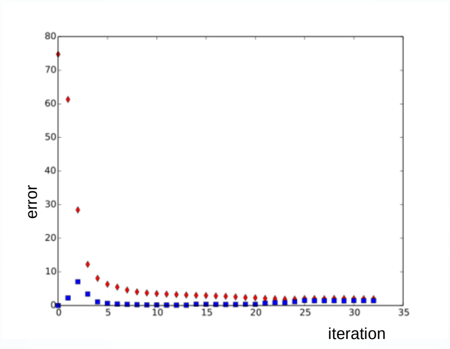


Fig. 1. Real and predicted results for selected happiness values



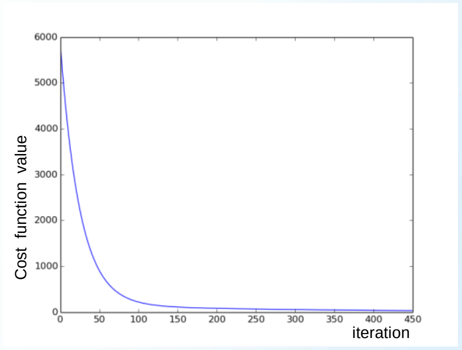


Fig. 2. Learning curves

Fig. 3. User cost function in time.

According to the learning curves (Fig.2), there is no underfitting or overfitting effect in our network as indicated by the two graphs approaching each other with a low error. The plot of the user cost function (Fig.3) points out that it is necessary to conduct at least 150 iterations to obtain satisfactory results.

Future work

Based on the results obtained so far, we are looking forward to optimizing the hypothesis function in order to make the whole system work more efficiently. Using a neural network with a hidden layer and logistic neurons would be probably a better idea than just a one layer network with linear neurons.

References

1. Andrew Ng: Machine Learning Course on Coursera.org https://www.coursera.org/course/ml
2.Geoffrey Hinton: Neural Networks for Machine Learning Course on Coursera.org `https://www.coursera.org/course/neuralnets`

Acknowledgments

This research is partly supported by the European Union within the EU ICT-269978 VPH-Share Project.